ALIBABA CLOUD

Alibaba Cloud

Apsara Stack Enterprise

Technical Whitepaper

Product Version: 2006, Internal: V3.12.0

Document Version: 20210915

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.IDC requirements

The features and performance of Apsara Stack platforms and services depend on the reliability (24/7 stable operation of servers and network devices) of Apsara stack data centers. This stability relies on the reliability of a series of complex infrastructure such as cooling and power supply. We recommend that you abide to tier 3 or a similar classification when building data centers that host Apsara Stack platforms to reduce stability risks in essence.

1.1. Environment requirements

This topic describes the environment requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Areas prone to flooding, such as the downstream of dams or flood-prone regions	Data centers cannot be set up in such areas.	Required
2	Areas prone to landslides, debris flows, or mountain slopes	Data centers cannot be set up in such areas.	Required
3	Seismic zones or fault zones	Data centers cannot be set up in such areas.	Required
4	Distance from areas where have experienced 100-year floods	No less than 100 meters.	Required
5	Distance from hazardous areas in chemical plants, landfills, gas stations, and polluted sites that have flammables and explosives such as dangerous chemicals and gas.	No less than 400 meters.	Required
6	Distance from military arsenals	No less than 1,600 meters.	Required
7	Distance from airports	The distance from both sides of the runway is no less than 1,000 meters. The distance from runways in the direction of takeoff and landing is no less than 8,000 meters.	Required
8	Distance from public parking lots	No less than 20 meters.	Required
9	Main roads of the physical park	At least two roads are required. One road must be a two-lane, two-way road, which can accommodate trucks 15 meters long and 3 meters wide.	Recommended
10	Distance from commercial and residential areas	No greater than 16,000 meters.	Recommended

No.	Description	Requirement	Matching type
11	Physical park	The physical park is independent or can be isolated to provide secure isolation.	Recommended

1.2. Building requirements

This topic describes the building requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Gross floor area of a single building	No less than 8,000 square meters.	Recommended
2	Acceptance of fire protection systems installed in buildings	Fire protection systems installed in buildings are tested and approved by the local fire department.	Required
3	Floor load capacity	More than 1,000 kg per square meters.	Required
4	Layer height	The clear span of buildings is greater than 3.6 meters.	Required
5	Transportation	Freight elevators are required for buildings no less than two floors and have a weight capacity of no less than two tons. The transportation aisles are no less than 2.4 meters wide and no less than 2.5 meters high.	Required
6	Classification of seismic protection of building constructions	The classification of seismic protection of building constructions is not lower than building type C.	Required
7	Fire-resistance rating	No less than Level 2.	Required
8	Waterproof rating	Level 1.	Required

1.3. Power system

This topic describes the requirements for power systems in Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Power introduction	At least a written certificate with power supply assurance is required.	Required

No.	Description	Requirement	Matching type
2	Route requirements for mains supply introduction	Dual routes are required and their distance must be greater than 10 meters. Cables are routed to the park on different roads.	Required
3	Requirements for mains supply introduction to substations	Class-A mains supply and two different circuits or two 10 kV, 35 kV, or 110 kV substations are used.	Required
4	Diesel generators	Diesel generators are configured for N + 1 redundancy. Diesel generators can start under load within two minutes.	Required
5	Period of time for which oil in tanks can be used	Greater than eight hours.	Required
6	Uninterruptible power supply (UPS) and redundancy	A UPS system based on 2N redundancy configuration is used for AC distribution. Or a high-voltage direct current (HVDC) system is used for a single mains supply.	Required
7	Period of time for which storage batteries can be discharged	No less than 15 minutes.	Required
8	Cabinet power distribution	A dual-circuit power supply system is used, which includes transformers, distribution lines, uninterruptible power supply, rack-mountable power distribution cabinets, and rack power distribution units (PDUs).	Required
9	Cabinet power consumption	No less than eight kW.	Recommended

1.4. Cooling system

This topic describes the requirements for the cooling system in Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Air conditioners, water pumps, water chiller units, and cooling towers in data centers	Air conditioners, water pumps, water chiller units, and cooling towers in data centers are configured for N + 1 redundancy.	Required
2	Power distribution for precision air conditioners in a chilled water system	Uninterrupted power supply (UPS)	Required
3	Power distribution for water supply pumps in a water-cooling system	UPS	Required
4	Period of time for which cool storage equipment can provide cooling	Time period during which cool storage equipment can provide cooling is no less than 10 minutes. When the cooling system is interrupted, the temperature of cold aisles in data centers cannot exceed 30 degrees Celsius.	Required
5	Building automation system	UPS. Redundancy must be provided for the direct digital controller (DDC) system and servers.	Recommended

1.5. Monitoring requirements

This topic describes the monitoring requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Monitoring access standards	Network communication is enabled based on TCP or IP sockets.	Recommended
2	Monitoring scope	The following items in data centers are monitored: temperature and humidity inside the data centers, terminal devices of the air conditioning system, chillers, pumps of the air conditioning system, power distribution cabinets, high-voltage direct current (HVDC) systems, uninterrupted power supply (UPS) systems, transformers, diesel generators, and mains supply.	Required

1.6. O&M requirements

This topic describes the O&M requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Technical team	A technical team must consist of the following personnel: one person for building decoration, one to two persons for air conditioning and refrigeration, one to two persons for high voltage power system, and at least one person for low voltage system monitoring.	Recommended
2	Construction delivery capability	The business deployment requirements are met (1,000 cabinets delivered within six months).	Recommended
3	Service-level agreement (SLA)	The availability of power, cooling, and network is above 99.99%.	Recommended
4	O&M personnel in the O&M system	The level and number of O&M personnel are confirmed.	Required
5	Professional qualifications of O&M personnel in the O&M system	The number of professional and technical personnel is no less than two in each of the following fields: electrical system, heating, ventilation, and air conditioning (HVAC), fire protection, and low voltage system.	Recommended
6	Duty system of the O&M system	The personnel on duty and emergency response mechanism are available 24/7/365 for infrastructure and network maintenance in data centers.	Required
7	Hardware and software maintenance of devices in the O&M system	A 24/7/365 professional maintenance service is purchased.	Required
8	Building management system (BMS) and video surveillance in the O&M system	The power and environment supervision system or BMS is used to monitor the running status of key infrastructure. The 24/7 video surveillance is provided, and the records are retained for 90 days.	Required

No.	Description	Requirement	Matching type
9	Entry and exit management of personnel and articles in the O&M system	A clear management process is provided, and records are complete and traceable.	Required
10	Service qualification	IDC business qualification: An Internet Data Centre Value Added Telecom Service license (IDC VATS) issued by the Chinese government is recommended.	Recommended
11	Third-party certification	The SSAE 16, ISO 17799, and ISO 9001 audits are passed. SSAE 16 ensures that service providers have sufficient security controls and safeguards in place to protect the security of user data. ISO 17799 ensures that the information security of service providers is less likely to be damaged. ISO 9001 sets out the criteria for a quality management system (QMS) of service providers.	Recommended
12	Operator personnel handover interface	A clear personnel handover interface is provided, including the assignment of roles and responsibilities, and the problem escalation path to the personnel who is in charge of the project and who holds the highest rank on the operator side. All responsibilities must be assigned and confirmed at the beginning of the project and be carried out for the entire project.	Recommended

1.7. Communication requirements

This topic describes the communication requirements for Apsara Stack data centers.

No.	Description	Requirement	Matching type
1	Number of direct routes between two data centers	Two direct routes with distances greater than 500 meters. Cables cannot be routed on the same conduit, trench, or route.	Required

No.	Description	Requirement	Matching type
2	Number of outgoing routes in a single data center	Provide two outgoing routes. The number of outgoing routes can be expanded to three as required before the project is delivered.	Recommended
3	Number of outgoing optical fibers	No less than 20 pairs.	Recommended
4	Routing method of optical cables	Optical cables must be placed into buried conduits. Overhead cabling is not allowed.	Required
5	Connection method of optical cables	Optical cables must be connected inside data centers. Outdoor connections are not allowed.	Required
6	Outgoing conduits	The park has more than two outgoing conduits in different directions and their distance is greater than 50 meters. Each outgoing conduit corresponds to a different entrance room.	Recommended
7	Communication rooms	There are two separate communication rooms in each data center.	Recommended
8	Leased lines and bandwidth	The data centers have the capabilities to support leased lines, Border Gateway Protocol (BGP) lines, and static bandwidth.	Recommended
9	Optical cable routes inside data centers	Cables inside data centers must be routed separately to ensure dual routes. The distance between the two routes must be greater than 10 meters.	Required
10	Access to optical cables of other operators	The data centers can access to optical cables of other operators.	Recommended

No.	Description	Requirement	Matching type
11	Number of direct routes between buildings	Two routes are required and four routes are preferred within physical fences. Three routes are required and four routes are preferred outside fences built on property lines. These routes can be completed when data centers are delivered. Direction of the routes must be approved by Alibaba Cloud.	Recommended
12	Number of optical fibers between buildings	At least 384-core × 4 optical fibers are required and can be scaled out.	Recommended

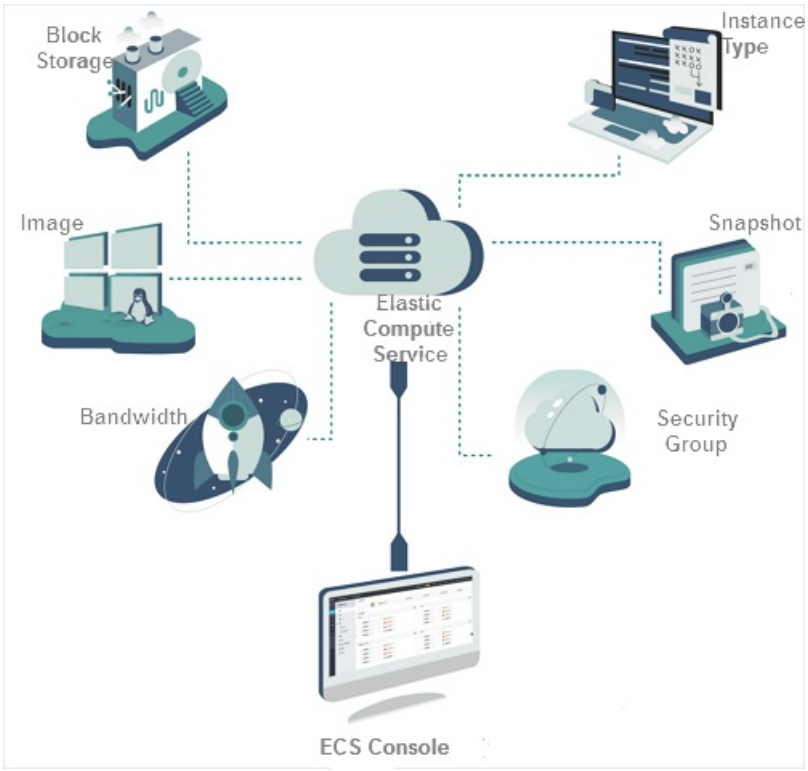
2.Elastic Compute Service (ECS)

2.1. What is ECS?

Elastic Compute Service (ECS) is a computing service that features elastic processing capabilities. Compared with physical servers, ECS instances are more user-friendly and can be managed more efficiently. You can create instances, resize disks, and add or release any number of ECS instances at any time based on your business needs.

An ECS instance is a virtual computing environment that contains the most basic components of computers such as the CPU, memory, and storage. Users perform operations on ECS instances. Instances are core components of ECS, and operations can be performed on instances through the ECS console. Other resources, such as block storage, images, and snapshots, can only be used after they are integrated with ECS instances. For more information, see ECS components.

ECS components



2.2. Architecture

2.2.1. Overview

The ECS system is composed of a virtualization platform with distributed storage, a control system, and an O&M and monitoring system.

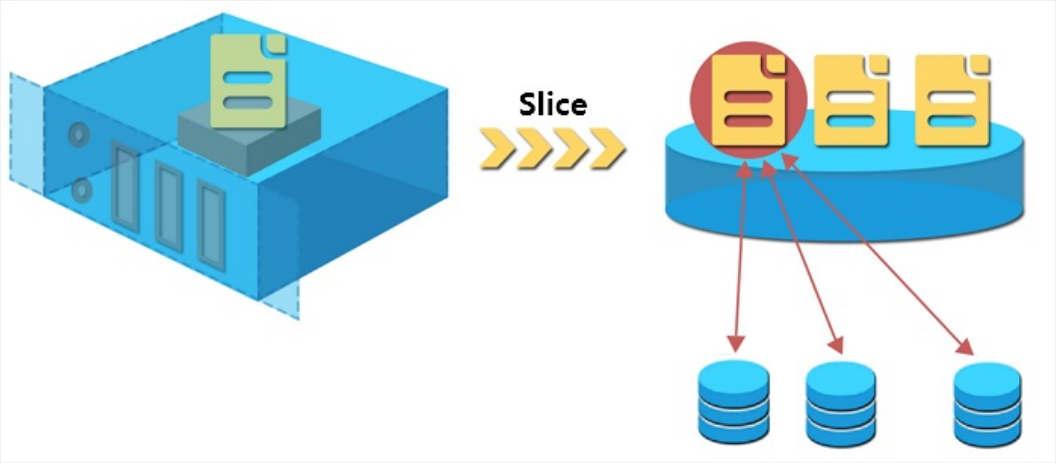
2.2.2. Virtualization platform and distributed storage

Virtualization is the foundation of ECS instances. Apsara Stack uses the Kernel-based Virtual Machine (KVM) virtualization solution to virtualize physical resources and provide them as ECS resources.

An ECS instance contains two important modules: the computing resource module and the storage resource module.

- Computing resources refer to CPU, memory, and bandwidth resources. These resources are created by virtualizing the resources of a physical server and then allocating them to ECS instances for use. The computing resources of a single ECS instance are based on those of a single physical server. When the resources of that physical server are exhausted, you must create new ECS instances on another physical server to obtain more resources. Resource Quality of Service (QoS) ensures that different ECS instances on a single physical server do not conflict with each other.
- ECS storage is provided by a large-scale distributed storage system. The storage resources of an entire cluster are virtualized and integrated into an external service. The data of a single ECS instance is distributed throughout the entire cluster. In the distributed storage system, all data is saved in triplicate. This allows damaged data in one copy to be automatically replicated from the other copies.

  Triplicate backup



  Automatic replication

2.2.3. Control system

The control system is the core of ECS. It determines the physical server on which to start ECS instances as well as processes and maintains all of the features and information of the ECS instances in a centralized manner.

The control system consists of the following modules:

- **Data collection module**

This module is responsible for collecting data throughout the virtualization platform, including data about the usage of computing, storage, and network resources. The data collection module serves as the basis for resource scheduling and allows you to perform centralized monitoring and management of cluster resource usage.

- **Resource scheduling system**

  This module determines on which physical server to start ECS instances. When an ECS instance is created, this module schedules the ECS instance based on the resource loads of the physical server. This module also determines where to restart an ECS instance when the instance fails.

- **ECS management module**

  This module manages and controls ECS instances such as starting, stopping, or restarting instances.

- **Security control module**

  This module monitors and manages the network security of the entire cluster.

2.2.4. ECS Bare Metal Instance

ECS Bare Metal Instance is a compute service that combines the elasticity of virtual machines and the performance and features of physical machines. ECS Bare Metal Instance is designed based on the state-of-the-art virtualization 2.0 technology developed by Alibaba Cloud. The virtualization technology used by ECS Bare Metal Instance is optimized to support common ECS instances and nested virtualization. It maintains the elastic performance of ECS instances and the performance and features of physical machines.

ECS Bare Metal Instance combines the strengths of both physical machines and ECS instances to deliver powerful and robust computing capabilities. ECS Bare Metal Instance uses virtualization 2.0 to provide your business applications with direct access to the processor and memory resources of the underlying servers without virtualization overheads. ECS Bare Metal Instance retains the hardware feature sets (such as Intel® VT-x) and resource isolation capabilities of physical machines, which is ideal for applications that need to run in non-virtualization environments.

By virtue of the independently developed chips, hypervisor system software, and the redefined server hardware architecture, ECS Bare Metal Instance integrates features from both physical and virtual machines. ECS Bare Metal Instance can seamlessly connect with other Apsara Stack services for storage, networking, and database tasks. ECS Bare Metal Instance is fully compatible with ECS instance images. These properties allow you to build resources to suit your business requirements.

When you use ECS Bare Metal Instance, take note of the following items:

- ECS Bare Metal Instance does not support instance type changes.
- When the physical machine that hosts an ECS bare metal instance fails, the system fails the instance over to another physical machine. Data is retained within the data disks of the instance.

2.3. Features

This topic describes the features of ECS instances.

ECS instances are the core component that provides computing services to users in ECS. It takes only a few minutes to create and start an ECS instance. When an ECS instance is created, it has specific system configurations. Compared with traditional servers, ECS instances allow you to compute business data more efficiently.

ECS instances are used and managed in the same way as physical servers. You can perform a series of basic operations on ECS instances remotely or by calling API operations.

The processing power of ECS instances can be expressed in terms of virtual CPUs and virtual memory, while the storage capabilities of ECS disks are measured by the available capacity of cloud disks. ECS instances support more flexible machine configurations than traditional servers. If you find that the configurations of an ECS instance do not meet your business needs, you can flexibly configure them.

The lifecycle of an ECS instance begins when it is created and ends when it is released. After an ECS instance is released, all of its data is permanently deleted and cannot be recovered.

The ECS console in Apsara Stack Cloud Management (ASCM) consists of the following pages:

- Overview

  You can view the number of created and running instances, as well as the distribution of ECS resources in each zone.

- Instances

  On the Instances page, you can view and manage the instances that you have created. You can start, stop, restart, and release instances, as well as log on to the VNC management terminal, replace system disks, modify passwords, and change instance configurations. You can also view the basic information and configurations of instances.

- Disks

  On the Disks page, you can view and manage the disks that you have created. You can re-initialize disks online, create snapshots, configure automatic snapshot policies, release disks, and attach or detach disks. You can also view the basic information and attaching information of disks.

- Images

  On the Images page, you can view and manage the images that you have created or shared. You can copy, share, and delete images.

- Snapshots

  On the Snapshots page, you can view and manage the snapshots that you have created. You can restore disks online, create custom images, and delete snapshots.

- Automatic snapshot policies

  On the Snapshots page, you can view and manage the automatic snapshot policies that you have created. You can configure automatic snapshot policies in batches, modify automatic snapshot policy information, and delete automatic snapshot policies.

- Security groups

  On the Security Groups page, you can view and manage the security groups that you have created. You can create, modify, delete, and batch delete security groups, as well as view the instances and rules associated with a security group.

- ENIs

  On the ENIs page, you can view and manage the elastic network interfaces (ENIs) that you have created. You can create, modify, and delete ENIs, as well as bind ENIs to or unbind ENIs from ECS instances.

- Deployment sets

On the Deployment Sets page, you can view and manage the deployment sets that you have created. You can create, modify, and delete deployment sets, as well as view the basic information of deployment sets.

3.Container Service for Kubernetes

3.1. What is Container Service?

Container Service provides high-performance, enterprise-class management for scalable Kubernetes-based containerized applications throughout the application lifecycle.

Container Service simplifies the creation and scaling of container management clusters. It integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud. Alibaba Cloud is a Kubernetes certified service provider, with Container Service being among the first services to pass the Certified Kubernetes Conformance Program. Container Service provides professional container support and services.

3.2. Container technology

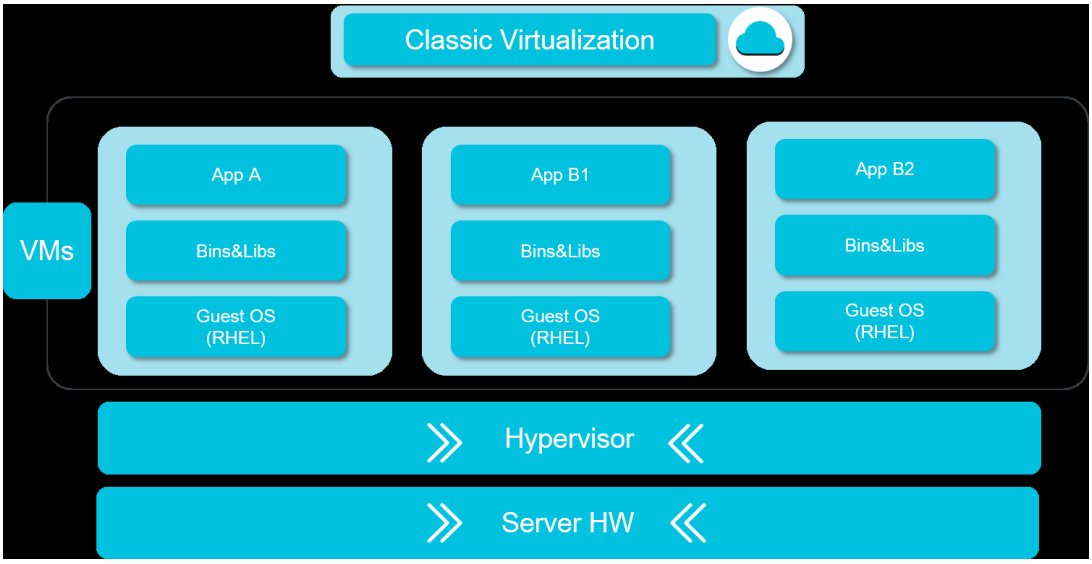
Containers are a lightweight operating system-level virtualization technology. You can use container images to deliver applications. Container images include applications and their necessary runtime dependencies. Container images have excellent portability and ensure deployment consistency in different environments. Containers are isolated from each other during runtime, ensuring excellent security.

Containers avoid potential version conflicts resulting from different applications running in the same environment, and eliminate runtime environment inconsistencies resulting from the same software being run in different environments. Because all containers on a host share the host's OS kernel, containers are more lightweight than virtual machines. This allows you to start containers quickly and gain fine-grained control over container resources.

Container technology and virtualization

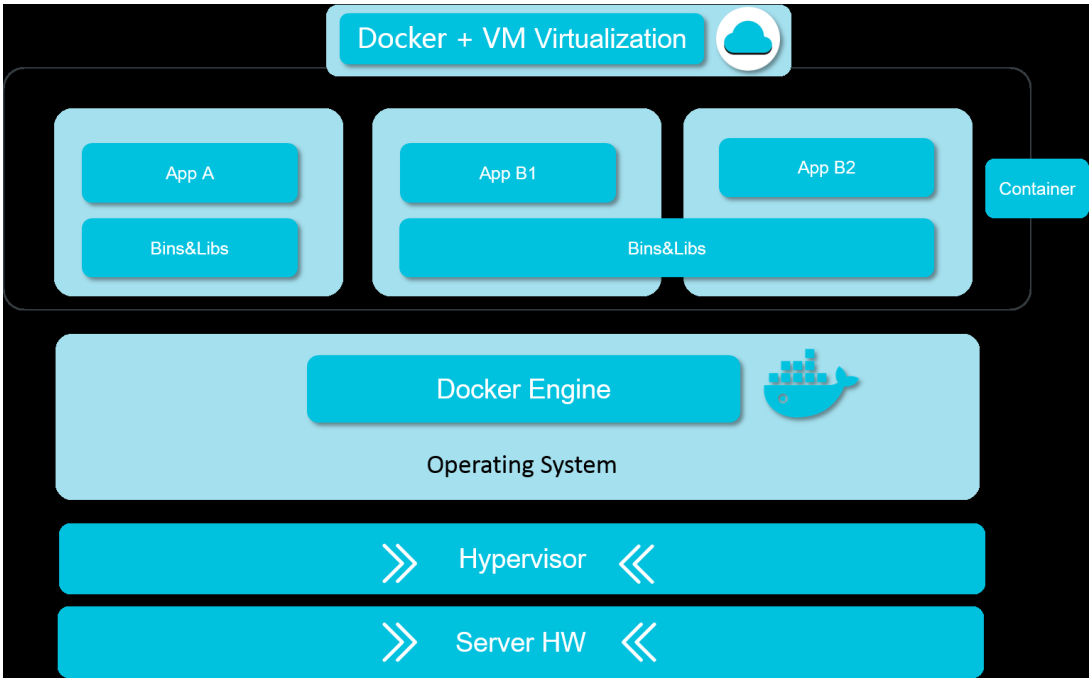
Containers do not conflict with conventional virtualization technologies. Conventional virtualization technologies encompass all elements ranging from operating systems to applications, as shown in the following figure.

Classic virtualization



Containers only package the application code and its runtime environments. Images can be reused within the same environment in different containers, making containers simple to use and operate.

Combination of Docker and virtualization



By combining containers and virtualization technologies, you can use virtual machines to provide an elastic infrastructure that offers improved security isolation and live migration capabilities. You can also use the container technology to streamline the deployment and O&M of applications and implement an elastic application architecture.

Technical features

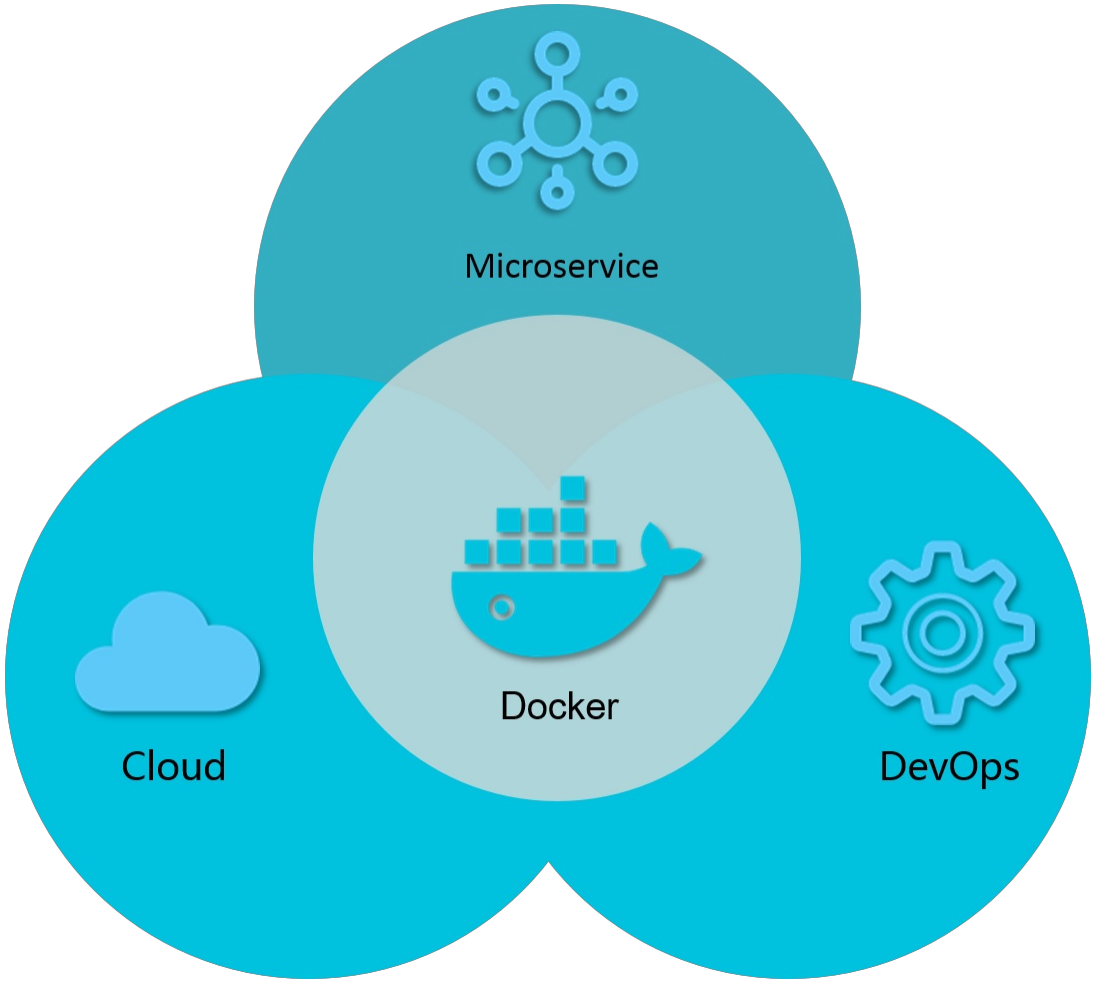
Containers are agile, portable, and highly-controllable.

- **Agility**: Containers attract developers with their simplicity and velocity, and allow enterprises to consistently develop and deliver software with greater efficiency.
- **Portability**: Developers can migrate containerized applications from the development environment, to the testing environment, and ultimately to the production environment. During this process, the operating structures for identical images are consistent. Computing capabilities can be deployed across data centers, making computing capability migration a reality in hybrid clouds.
- **Controllability**: Applications in the production environment must meet SLA goals. This requires that you have comprehensive management, security, and monitoring capabilities. Containers provide standardized application environments, allowing developers to use automated tools to manage the infrastructures and applications and ensure that all operations are automated, controllable, and traceable.

Scenarios

Containers can be applied in a wide range of scenarios. Containers are most often discussed and researched in relation to scenarios that have high container technology requirements, especially DevOps, cloud application management, and microservices.

Common scenarios



3.3. Architecture

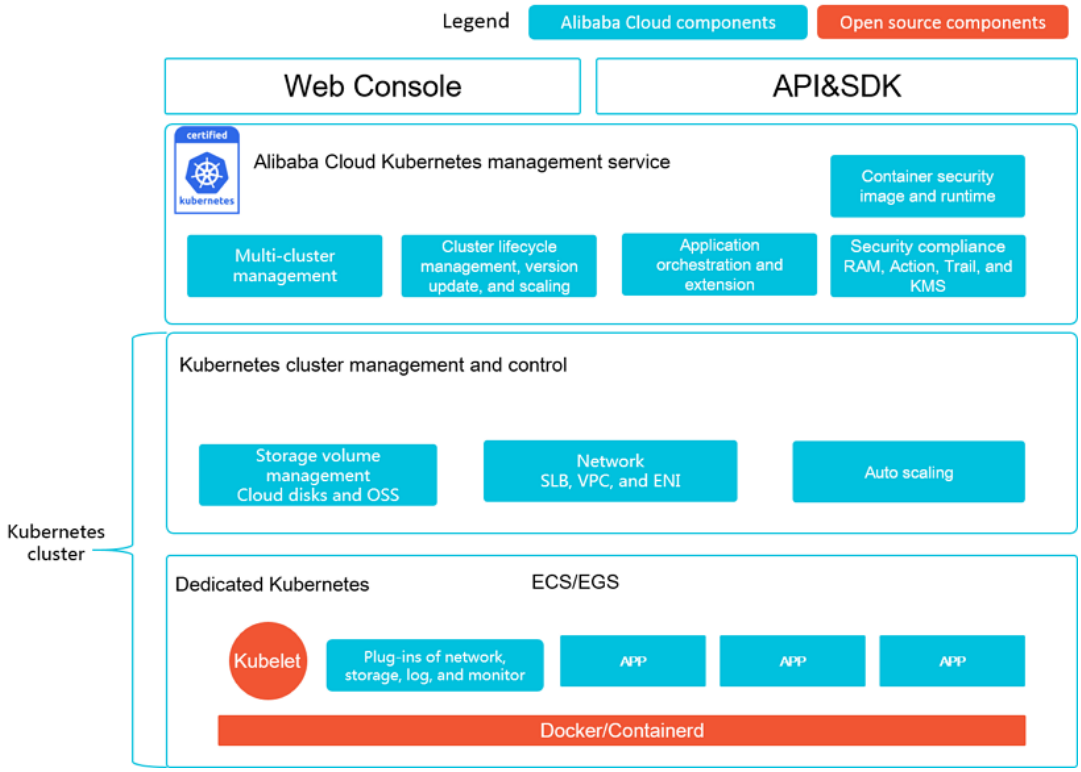
Apsara Stack Container Service supports YAML orchestration and cluster management for Kubernetes to extend and optimize third-party capabilities on Apsara Stack. Container Service allows you to manage clusters and containerized applications through GUIs and APIs.

The underlying architecture allows you to use exclusive cloud servers or physical servers to create a secure and controllable underlying environment where you can customize security group and VPC security rules.

To help migrate your applications to the cloud at a lower cost, Container Service implements APIs that are compatible with standard Docker APIs and all Docker images. Container Service provides Kubernetes YAML orchestration templates which allow you to migrate your applications seamlessly to the cloud. It also provides flexible and customizable mechanisms for third-party capability extensions.

The following figure shows the Container Service architecture.

Architecture

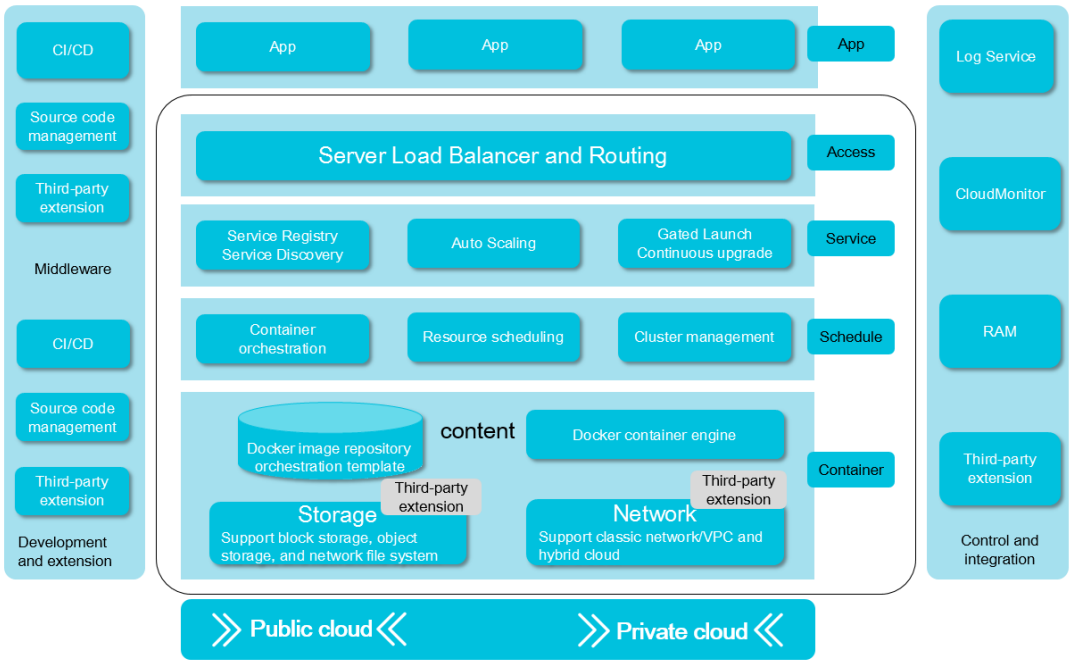


Container Service is adapted and enhanced on the basis of native Kubernetes. This service simplifies cluster creation and scaling and integrates Apsara Stack virtualization, storage, network, and security capabilities, providing the optimal environment to run Kubernetes-based containerized applications in the cloud.

Feature	Description
Dedicated Kubernetes mode	Integrated with Apsara Stack virtualization technologies, the service allows you to create dedicated Kubernetes clusters. Elastic Compute Service (ECS), Elastic GPU Service (EGS), and ECS Bare Metal instances can be used as cluster nodes. Instances can be flexibly configured to different specifications and support a wide range of plug-ins.
Apsara Stack Kubernetes cluster management and control service	The service provides powerful network, storage, cluster management, scaling, and application extension features.
Apsara Stack Kubernetes management service	The service supports secure images and is highly integrated with Apsara Stack Resource Access Management (RAM), Key Management Service (KMS), and logging and monitoring services to provide a secure and compliant Kubernetes solution.
Convenient and efficient use	Container Service for Kubernetes provides services through the Web console, APIs and SDKs.

The following figure shows the Container Service capability stack. Container Service is built on a cloud infrastructure. It is deeply integrated with Apsara Stack capabilities, and supports third-party extensions and applications.

Functional architecture



3.4. Features

Features

Cluster management

- With the Container Service console, you can easily create a classic dedicated Kubernetes cluster supporting GPU servers within 10 minutes.
- Provides container-optimized OS images as well as Kubernetes and Docker versions that have undergone **stability testing and security enhancement**.
- Supports multi-cluster management, cluster upgrades, and cluster scaling.

Provides end-to-end container lifecycle management

- **Network**

  Provides high performance VPC and elastic network interface (ENI) plug-ins optimized for Apsara Stack, boasting 20% increased performance compared with regular network solutions.

  Supports container access and throttling policies.

- **Storage**

  Container Service is integrated with Apsara Stack disks and OSS, and provides the standard FlexVolume drive.

  Supports real-time creation and migration of volumes.

- **Logs**

  Provides high-performance log collection integrated with Apsara Stack Log Service.

Supports the integration with third-party open-source logging solutions.

- **Monitoring**

  Supports both container-level and VM-level monitoring. Integration with third-party open-source monitoring solutions is supported.

- **Permissions**

  Supports cluster-level Resource Access Management (RAM).

  Supports application-level permission configuration management.

- **Application management**

  Supports phased release and blue-green release.

  Supports application monitoring and scaling.

High-availability scheduling policies that allow you to easily handle upstream and downstream delivery processes

- Supports service-level affinity policies and scale-out.
- Provides high availability and disaster recovery across zones.
- Provides cluster and application management APIs to easily implement continuous integration and private system deployment.

4.Auto Scaling (ESS)

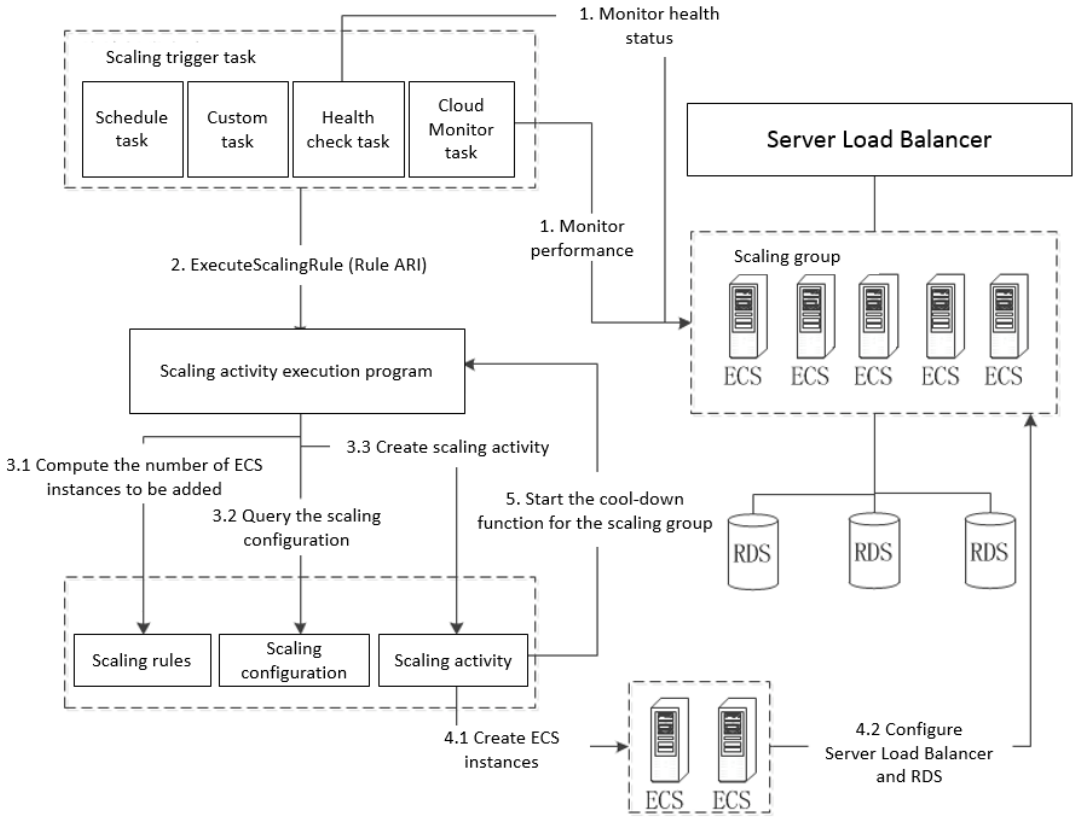
4.1. What is ESS?

Auto Scaling (ESS) is a management service that automatically adjusts the number of elastic computing resources based on your business demands and policies. It is suitable for applications with fluctuating business loads, as well as applications with stable business loads.

ESS automatically schedules computing resources based on customer policies and business changes. It provides support for changing business loads and helps control infrastructure costs within an acceptable range. ESS automatically creates ECS instances based on user-defined scaling policies and modes. When business loads increase, ESS automatically adds ECS instances to ensure sufficient computing capabilities. When business loads decrease, ESS automatically removes ECS instances to save costs. ESS also replaces unhealthy ECS instances to ensure service performance and business availability.

Additionally, ESS is seamlessly integrated with Server Load Balancer (SLB) and ApsaraDB for RDS (RDS). This allows ESS to add or remove ECS instances to or from the backend server groups of the associated SLB instances, as well as to add or remove IP addresses of ECS instances to or from the whitelists of the associated RDS instances. ESS adapts to various complex scenarios without the need for manual operation and automatically processes business loads based on actual requirements. For more information, see Diagram of ESS.

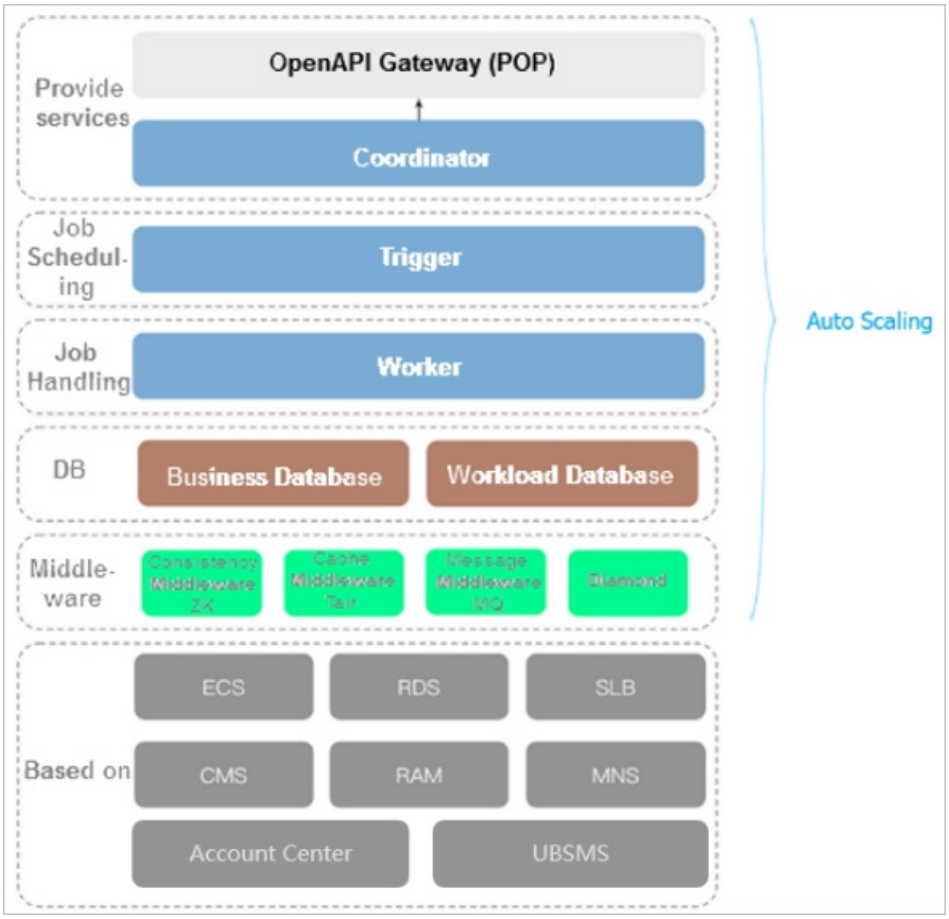
Diagram of ESS



4.2. Architecture

ESS is a system that orchestrates ECS instances and provides services based on basic components such ECS. The ESS system consists of trigger, worker, database, and middleware services.

Diagram of the ESS architecture



Architecture description

Layer	Description
Middleware layer	ZooKeeper: ensures consistency by implementing distributed locks for Server Controller.
	Tair: provides caching services for Server Controller.
	Message Queue (MQ): provides message queuing services of VM statuses.
	Diamond: manages persistent configurations.
Database layer, which contains the business database and workload database	Worker: the core of ESS. After ESS receives a task, it processes the entire lifecycle of the task, including splitting the task, executing the task, and returning the execution results.
	Trigger: obtains information from health checks of instances and scaling groups, scheduled tasks, and Cloud Monitor to perform tasks scheduling.

Layer	Description
Public-facing services	Coordinator: serves as the ingress of the ESS architecture. It provides external management and control for services, process API calls, and triggers tasks.
	Open API Gateway: provides basic services such as authentication and parameter passthrough.

4.3. Features

4.3.1. Scenarios

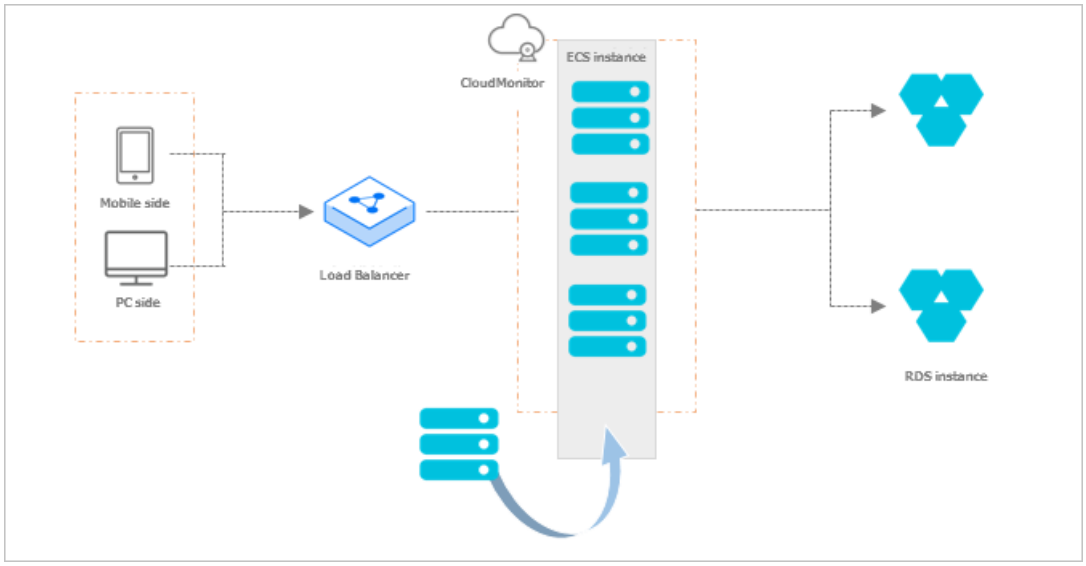
4.3.1.1. Overview

ESS automatically adjusts the number of elastic computing resources to meet fluctuating business demands. When business loads increase, ESS automatically adds ECS instances based on user-defined scaling rules to ensure sufficient computing capabilities. When business loads decrease, ESS automatically removes ECS instances to save costs.

4.3.1.2. Scale-out

When business loads surge above normal loads, ESS automatically increases underlying resources. This helps maintain access speed and ensure that resources are not overloaded.

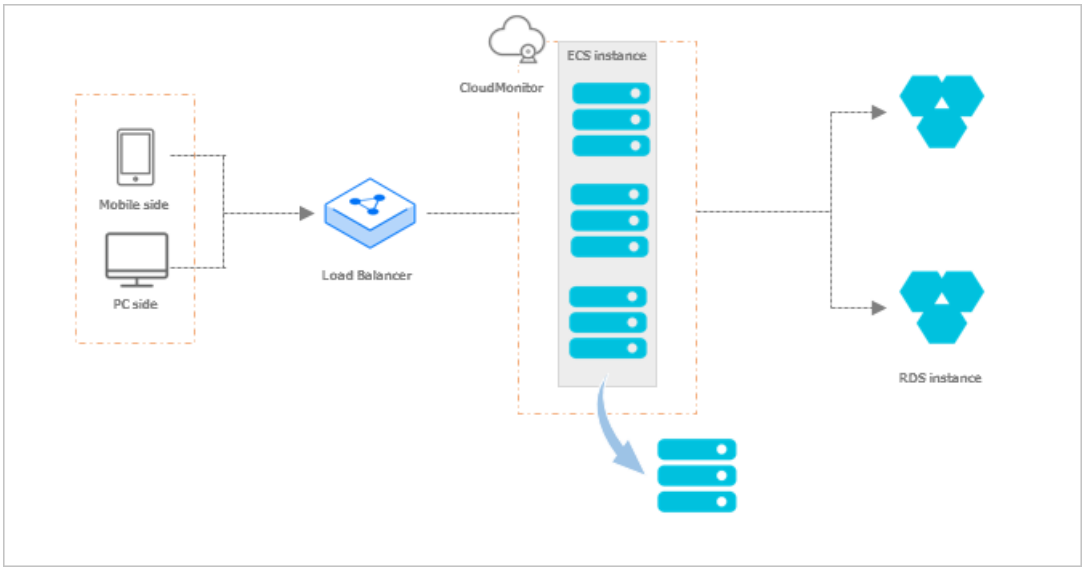
You can create scheduled tasks to perform automatic scale-out at specified points in time or configure Cloud Monitor to monitor ECS instance usage in real time and perform scale-out based on actual requirements. For example, when Cloud Monitor detects that the vCPU utilization of ECS instances in a scaling group exceeds 80%, ESS automatically scales out ECS resources based on user-defined scaling rules. During the scale-out event, ESS automatically creates ECS instances and adds these ECS instances to the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB for RDS instances. The following figure shows the implementation of a scale-out event.



4.3.1.3. Scale-in

When business loads decrease, ESS automatically releases underlying resources to prevent resource wastes and reduce costs.

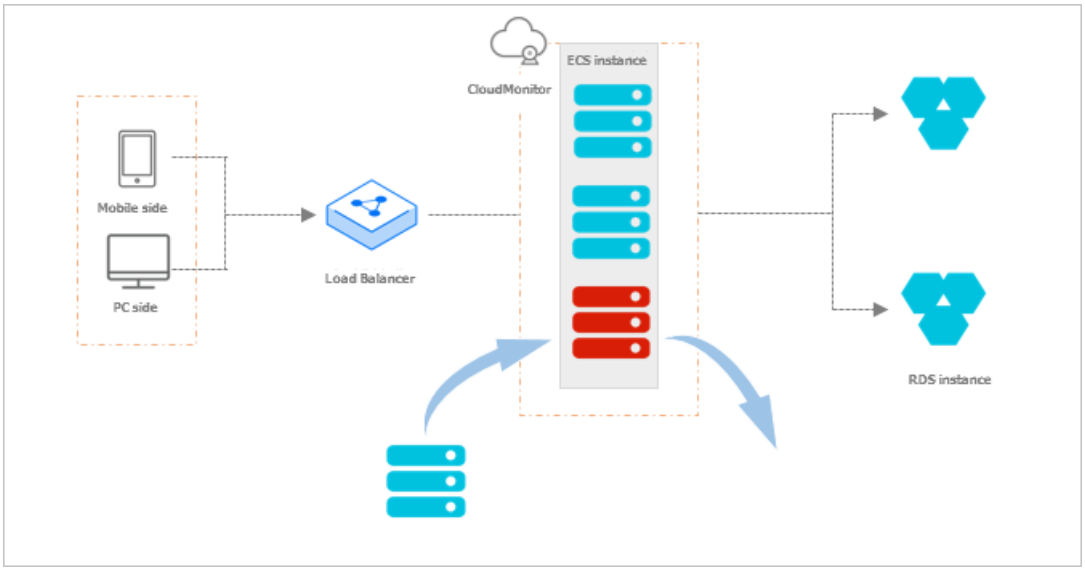
You can create scheduled tasks to automatically scale in ECS resources at specified points in time. You can also configure Cloud Monitor to monitor ECS instance usage in real time and scale in resources based on actual requirements. For example, when Cloud Monitor detects that the vCPU utilization of ECS instances in a scaling group is less than 30%, ESS automatically scales in ECS resources based on the scaling rule that you specified. During the scale-in event, ESS releases ECS instances and removes these ECS instances from the backend server groups of the associated SLB instances and the whitelists of the associated ApsaraDB for RDS instances. The following figure shows the implementation of a scale-in event.



4.3.1.4. Elastic recovery

ESS provides the health check feature and automatically monitors the health status of ECS instances in a scaling group, so that the number of healthy ECS instances in the scaling group does not fall below the user-defined minimum value.

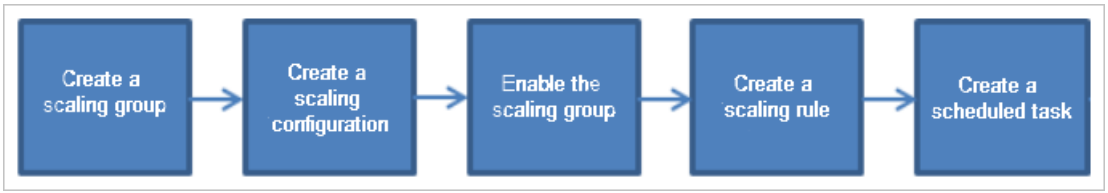
When ESS detects that an ECS instance is unhealthy, it automatically releases the unhealthy ECS instance, creates a new ECS instance, and adds the new instance to the backend server group of the associated SLB instance and the whitelist of the associated ApsaraDB for RDS instance. The following figure shows the implementation of elastic recovery.



4.3.2. Components

To create a complete automatic scaling solution, you must create scaling groups, configurations, rules, as well as scheduled tasks or event-triggered tasks.

The following figure shows the procedure to create a complete automatic scaling solution.



Scaling groups

A scaling group is a group of ECS instances that are dynamically scaled based on the configured scenario. You can specify the minimum and maximum numbers of ECS instances in a scaling group, as well as the SLB and ApsaraDB for RDS instances associated with the group.

Scaling configurations

A scaling configuration is a template in ESS for creating ECS instances. When you create a scaling configuration, you must configure the parameters for creating an ECS instance, such as the instance type, image type, storage size, and Secure Shell (SSH) key pair that is used to log on to the ECS instance. You can also modify an existing scaling configuration.

Scaling rules

A scaling rule specifies a specific scaling activity, such as adding or removing ECS instances. The following scaling rules are supported:

- Change to N instances: After a scaling rule is executed, the number of ECS instances in a scaling group is adjusted to a specific value.

- Add N instances: After a scaling rule is executed, a specific number of ECS instances are added to the scaling group.
- Remove N instances: After a scaling rule is executed, a specific number of ECS instances are removed from the scaling group.

Scheduled tasks

A scheduled task specifies execution actions within a scaling group. It can trigger a specific scaling rule at a specific point in time to execute a scaling activity, such as adjusting the number of ECS instances in a scaling group.

Event-triggered tasks

Event-triggered tasks are scaling tasks associated with Cloud Monitor metrics and can be executed for automatic scaling in response to emergent or unpredictable business changes. After an event-triggered task is created, ESS collects monitoring data for the specified metric in real time and triggers an alert when the metric value meets the alert condition. Then, ESS executes the corresponding scaling rule to dynamically adjust the number of ECS instances in the scaling group.

5.Resource Orchestration Service (ROS)

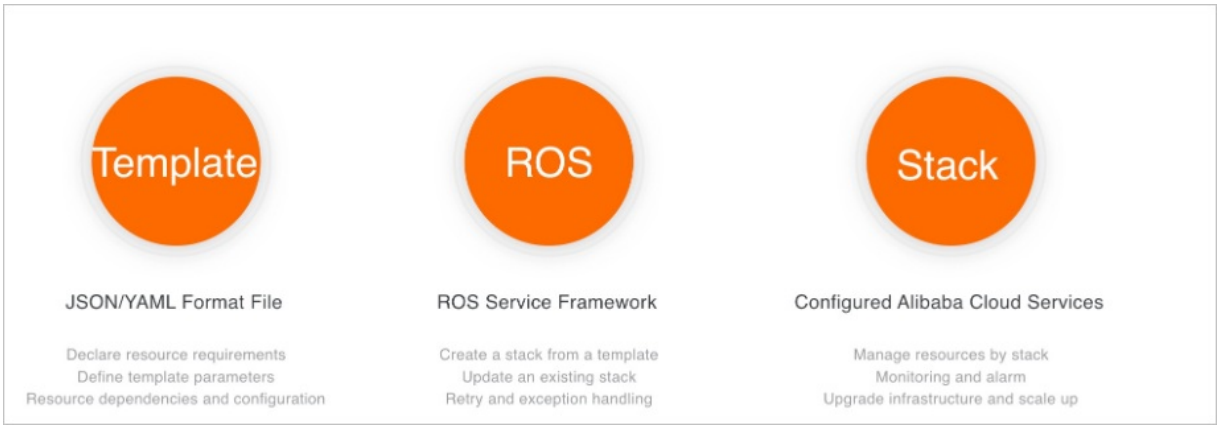
5.1. What is ROS?

Resource Orchestration Service (ROS) is a service provided by Alibaba Cloud to simplify the management of cloud computing resources. You can author stack templates based on the template specifications defined in ROS. Within a template, you can define required cloud computing resources such as ECS and ApsaraDB for RDS instances, and the dependencies between resources. The ROS engine automatically creates and configures all resources in a stack based on a template, making automatic deployment and O&M possible.

An ROS template is a readable, easy-to-author text file. You can directly edit a JSON-formatted template or use the Visual Editor available in the ROS console to edit the template. You can modify templates at any time. You can use version control tools such as SVN and Git to control the template and infrastructure versions. You can use APIs and SDKs to integrate the orchestration capabilities of ROS with your own applications to implement infrastructure as code.

ROS templates are also a standardized way to deliver resources and applications. If you are an independent software vendor (ISV), you can use ROS templates to deliver a holistic system and solution encompassing cloud resources and applications. ISVs can use this method to integrate Alibaba Cloud resources with their own software systems for centralized delivery.

ROS manages a group of cloud resources as a single unit called a stack. A stack is a group of Alibaba Cloud resources. You can create, delete, and clone cloud resources by stack. In DevOps practices, you can use ROS to clone the development, testing, and production environments, as well as migrate and scale out applications.



5.2. Benefits

This topic describes the benefits of ROS. ROS provides a simple and convenient way to automate resources management.

You can use ROS to model and configure your cloud resources. After you create a template that defines your required resources such as ECS and ApsaraDB for RDS instances, ROS creates and configures these resources based on the template.

ROS provides flexible and convenient services at low costs. This allows you to focus on your core business and implement Infrastructure as Code (IaC). In DevOps practices, you can clone the development, test, and production environments to simplify the overall migration and scaling of applications.

ROS has the following benefits.

Automated resource orchestration

ROS creates and manages the lifecycle of cloud computing resources based on the defined templates of the cloud resources and their dependencies. It automates resource configuration and deployment, streamlines versioning, tracks resource changes, and simplifies cloud application delivery. ROS can be integrated with APIs and SDKs to provide automated O&M capabilities.

Simplified resource management

If you want to create a scalable web application that contains backend databases or to create a cluster that consists of dozens of ECS instances, you need to deploy multiple resources such as ECS, ApsaraDB for RDS, Virtual Private Cloud (VPC), Auto Scaling (ESS), and Server Load Balancer (SLB). Typically, you must deploy each resource and manually orchestrate the resources to satisfy your needs. These tasks are time-intensive and add complexity to the operations.

ROS allows you to create and manage your stacks and resources in the following methods:

- Create or modify a template and define the resources and their dependencies in the template. ROS parses this template, creates the resources based on their dependencies and parameters, and automatically orchestrates the resources to satisfy your needs. This ensures that all resources created by using the template run properly.
- Adjust the stack template to fit your business needs.
- Delete all resources that are created by the same template in one click.
- Perform health checks on stacks in one click.

Quick replication of a collection of resources

If you have created a web application or cluster by using ROS, you can reuse the template to replicate the entire collection of resources. The template records the attributes and dependencies of each resource. You do not need to configure the resources again during resource replication.

Flexible integration with cloud products and services

You can use ROS to deploy and configure interactions between multiple cloud services. You can tailor your template based on your business and automated O&M requirements. ROS supports the following cloud products and services: ECS, ApsaraDB for RDS, KVStore for Memcache, KVStore for Redis, ApsaraDB for MongoDB, SLB, Object Storage Service (OSS), Log Service, Resource Access Management (RAM), and VPC.

Simple and visual creation of templates and stacks

ROS provides sample templates in the console to facilitate your operations. You can create a stack based on a sample template. During the stack creation, you need to configure only several parameters.

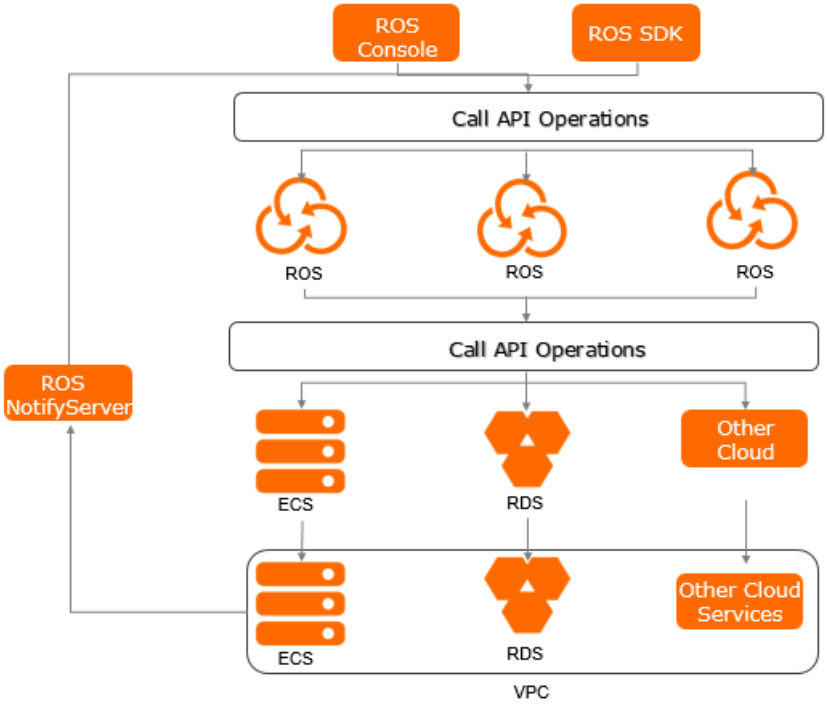
You can use Visual Editor to edit templates and view the stack structure.

5.3. Architecture

This topic introduces the architecture of ROS. You can use ROS by using the console, APIs, and SDKs.

ROS supports the following cloud products and services: ECS, ApsaraDB for RDS, ApsaraDB for MongoDB, KVStore for Redis, KVStore for Memcache, SLB, OSS, VPC, EIP, Auto Scaling, Log Service, and RAM.

The following figure shows the ROS architecture.



5.4. Features

This topic describes the features of ROS. ROS uses an orchestration engine to create and configure resources to perform automatic deployment and O&M.

Resource orchestration is an important service in cloud computing. You can define your cloud infrastructure as ROS templates and deploy them to the cloud from anywhere at any time to implement Infrastructure as Code (IaC). Compared with calling individual APIs of various services, ROS improves business development efficiency. Resources are created in stacks in ROS. You can also access these resources in their own console. You can manage stacks and their resources in the ROS console.

Create a template

A template is a JSON file that you can read and edit. ROS provides Visual Editor in the ROS console. You can use Visual Editor to edit ROS templates. You can also edit the JSON file directly. You can implement versioning by using tools such as SVN or Git. This allows you to control the versions of the infrastructure. You can also use methods such as APIs and SDKs to integrate the orchestration capabilities of ROS into your own applications and implement Infrastructure as Code (IaC).

The template also provides a standard delivery method for resources and applications. You can use the template to deliver integrated systems and solutions that contain cloud resources and applications. Independent software vendors (ISVs) can use this method to integrate cloud resources with their own software systems for consistent delivery.

To create and manage a stack, you must first edit a template, and then ROS can create and configure the resources in the stack based on the template.

Create a stack

A stack is a group of cloud resources that ROS manages as a single unit. Cloud resources can be created, deleted, modified, and cloned in stacks. In DevOps scenarios, you can clone the development, test, and production environments, simplifying the overall migration and scaling of applications.

If you have not created a template, select a sample template and complete the configurations in the ROS console to create a stack.

If you have created a template, select your template on the **My Templates** page in the ROS console and define the resources and their dependencies in the template.

Manage a stack

ROS provides an overview page for all your stacks. You can view the details of stacks, resources, events, and original templates. You can also use ROS to recreate, update, and perform health checks on your stacks. To recreate a stack, you can specify different parameter values in the original template of the current stack, and use the template to create a new stack. You can perform health checks on stacks to check whether the stacks are available. To update a stack, you can modify the original template of the stack.

In addition to creating templates and stacks, you can also view the details of resources, update, recreate, and delete stacks as needed. When you delete a stack, you can choose whether to retain or release resources in the stack.

- Update or recreate a stack
  - You can update or recreate the existing stacks based on your business needs.
  - If you need to modify the region where a stack resides or the current template of the stack, you can recreate the stack.
  - If you need to modify only the current stack name, creation timeout period, and parameters of the resources while leaving the template unchanged, you can update the stack.
- Delete a stack

  You can delete a stack that is no longer needed. You can choose whether to retain or release resources when you delete a stack.

Resource types

All the resource types that are supported by ROS are listed.

You can view the resource types supported by ROS and the details of each resource type. You can compile templates based on the **properties** of resource types and specify specific requirements for resources.

ECS instances

You can view the instance types and images supported by ECS in each zone. You can click **Create** in the instance type field in the ROS console to create resources.

6.Object Storage Service (OSS)

6.1. What is OSS?

6.1.1. Terms

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud.

Compared with user-created server storage, OSS has outstanding advantages in reliability, security, cost-effectiveness, and data processing capabilities. OSS enables you to store and retrieve a variety of unstructured data objects, such as texts, images, audios, and videos over the network at any time.

OSS is an object storage service based on key-value pairs. Files uploaded to OSS are stored as objects in buckets. You can obtain the content of an object based on the object key.

In OSS, you can:

- Create a bucket and upload objects to the bucket.
- Obtain an object URL from OSS to share or download the object.
- Modify the attributes or metadata of a bucket or an object, and configure ACL for the bucket or the object.
- Perform basic and advanced operations in the OSS console.
- Perform basic and advanced operations by using SDKs or calling RESTful API operations in your application.

6.1.2. Benefits

OSS provides secure, cost-effective, and highly reliable services for storing large amounts of data in the cloud. This topic compares OSS with the traditional user-created server storage to show the benefits of OSS.

Advantages of OSS over user-created server storage

Item	OSS	User-created server storage
Reliability	Automatically stores multiple copies of data for backup.	• Prone to errors due to low hardware reliability. If a disk has a bad sector, data may be lost. • Manual data restoration is complex and requires a lot of time and technical resources.

Item	OSS	User-created server storage
Security	• Provides hierarchical security protection for enterprises. • Provides resource isolation mechanisms for multiple tenants and supports zone-disaster recovery. • Provides various authentication and authorization mechanisms, as well as features such as whitelists, hotlink protection, RAM, and Security Token Service (STS) for temporary access.	• Additional scrubbing devices and black hole policy-related services are required. • A separate security mechanism is required.
Data processing	Provides Image Processing (IMG).	Equipment for data processing must be purchased and deployed separately.

More benefits of OSS

- Ease of use

  Provides standard RESTful API operations (some compatible with Amazon S3 API operations), a wide range of SDKs, client tools, and console. You can upload, download, retrieve, and manage large amounts of data for websites or mobile applications the way you use regular file systems.

  - The number and size of objects are not limited.
  - Streaming writes and reads are supported, which is suitable for business scenarios where you must simultaneously read and write videos and other large objects.
  - Lifecycle management is supported. You can delete expired data in batches.

- Powerful and flexible security mechanisms

  Flexible authentication and authorization mechanisms are available. OSS provides STS and URL-based authentication and authorization mechanisms, whitelists, hotlink protection, and RAM.

- Rich image processing functions

  Supports format conversion, thumbnails, cropping, watermarking, resizing for objects in formats such as JPG, PNG, BMP, GIF, WebP, and TIFF.

6.1.3. Scenarios

This topic describes the application scenarios of OSS.

Massive storage for image, audio, and video applications

OSS can be used to store large amounts of data, such as images, audio and video data, and logs. OSS supports various devices. Websites and mobile applications can directly read or write OSS data. OSS supports file writing and streaming writing.

Dynamic and static content separation for websites and mobile applications

By using the BGP bandwidth, you can download data from OSS with an ultra-low latency.

Offline data storage

OSS provides storage with low cost and high availability. Therefore, you can use OSS to store enterprise data that needs to be archived offline for a long period.

6.2. Benefits

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud. This topic describes the benefits of OSS.

Multiple functions

OSS supports the following functions:

- Upload objects by using simple upload, form upload, or append upload, download objects, delete objects, list objects, replicate objects, obtain object metadata, and create multipart upload tasks.
- Create buckets, delete buckets, list buckets, list objects in a bucket, and obtain bucket metadata.
- Create a globally unique bucket and perform cross-region replication (CRR) between buckets.
- Configure lifecycle rules for a bucket to define and manage the lifecycle of all or a subset of objects in a bucket. Change the capacity and ownership of a bucket.
- Configure zone-disaster recovery. In zone-disaster recovery mode, buckets with the same name are replicated. Cluster-based disaster recovery is automatically enabled based on configurations made when the cluster is created. In other words, after a primary bucket is created, a secondary bucket with the same name is automatically created. Information stored in the primary bucket is automatically synchronized to the secondary bucket.
- Configure static website hosting for a bucket and use the bucket domain name to access the static website.
- Configure hotlink protection based on the Referer field in HTTP requests.
- Configure cross-origin resource sharing (CORS). Configure logging and analyze logs in multiple dimensions. You can trace the source of access to your OSS resources.
- Prevent single point of failures (SPOFs) with the architecture that features redundancy.
- Perform concurrent multipart upload and download for large objects. You can also perform resumable upload or download.

High performance

OSS supports the throughput of a cluster that contains tens of thousands of nodes.

Security

OSS supports various features to ensure the security of your data.

- Provides access control lists (ACLs) to manage user access permissions. The following ACLs are supported: private, public read, and public read/write.
- Supports access control based on Apsara Stack tenant account and RAM users. You can create unique AccessKey pairs for each employee, application, and system based on the organization structure to control access to your resources. By default, RAM users do not have any permissions on OSS resources. You can use RAM to grant permissions to RAM users or use Security Token Service (STS) to grant permissions to temporary access.
- Supports server-side encryption, client-side encryption, and encryption transmission through HTTPS.

- Supports hotlink protection to prevent unauthorized access.
- Combines with the intrusion prevention system to prevent DDoS attacks and HTTP flood attacks and ensure that business works properly.
- Supports cross-region replication (CRR) to synchronize data to a specified region in real time for geo-disaster recovery. This way, OSS can protect important data from the impact of extreme disasters and ensures service stability.
- Supports zone-disaster recovery. In zone-disaster recovery mode, buckets with the same name are replicated. Cluster-based disaster recovery is automatically enabled based on configurations made when the cluster is created. In other words, after a primary bucket is created, a secondary bucket with the same name is automatically created. Information stored in the primary bucket is automatically synchronized to the secondary bucket.

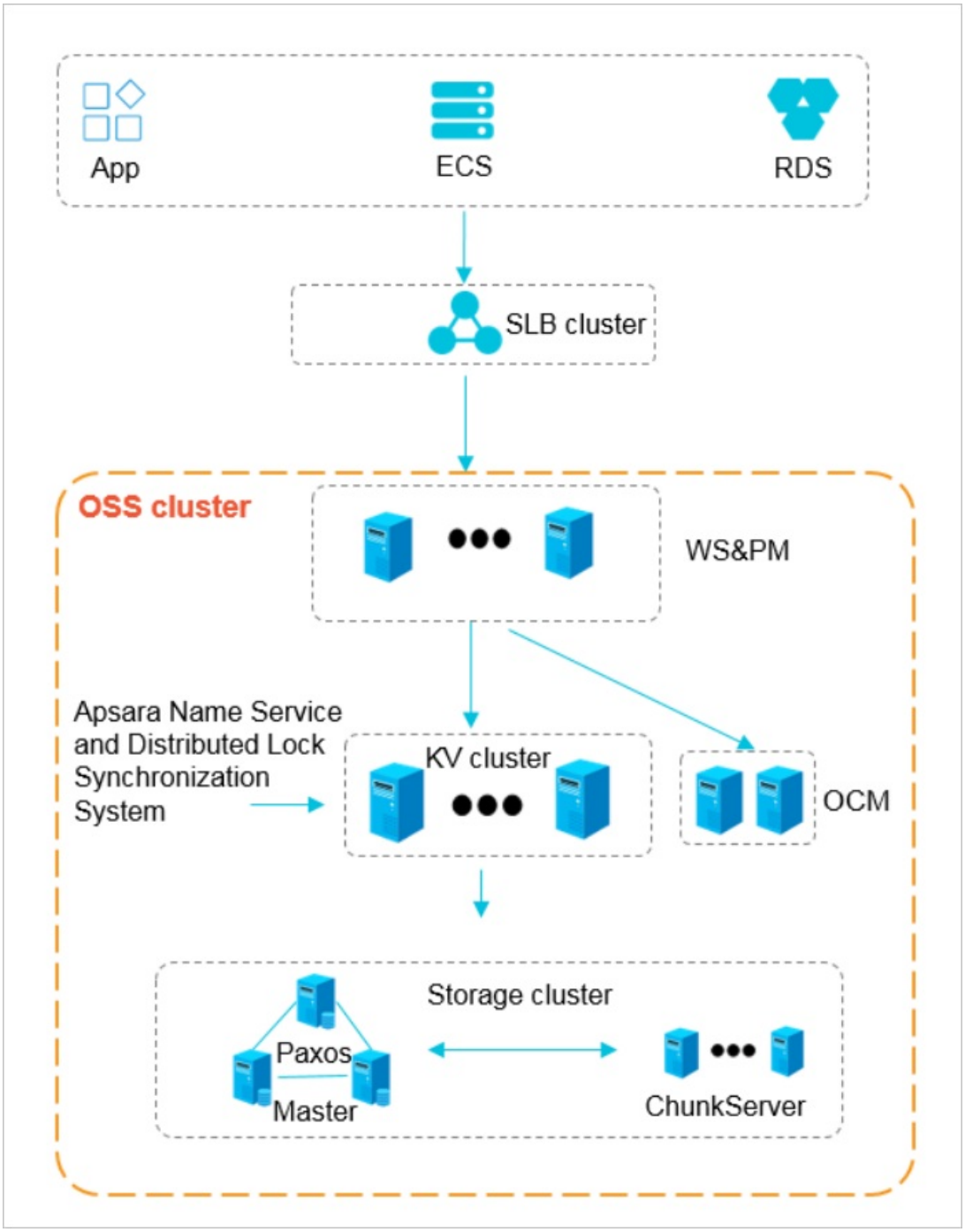
6.3. Architecture

6.3.1. System architecture

OSS is a storage solution that is built on the Apsara system. It is based on the infrastructure such as Apsara Distributed File System and SchedulerX. This infrastructure provides OSS and other Alibaba Cloud services with important features such as distributed scheduling, high-speed networks, and distributed storage.

The following figure shows the architecture of OSS.

OSS architecture



The OSS architecture is composed of three layers: protocol access layer, partition layer, and persistent storage layer.

- Protocol access layer
  - WS: uses the open-source Tengine component, and provides HTTP and HTTPS for external services.
  - PM: parses the HTTP request as the read/write operation on the back-end KV or another module. PM also receives and authenticates the user request sent through a RESTful protocol. If the authentication succeeds, the request is forwarded to KV Engine for further processing. If the request fails the authentication, an error message is returned.
- Partition layer

The partition layer uses keys to query and store structured data. This layer also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point. The partition layer manages indexes of objects, and converts objects to the persistent data objects at the persistent storage layer.

  - SchedulerX is responsible for naming services and is based on Apsara Name Service and Distributed Lock Synchronization System.
  - KV consists of KVMaster and KVServer. KVMaster manages and schedules partitions. KVServer stores indexes and actual data of partitions.

- Persistent layer

  The large-scale distributed file system is deployed at the persistent storage layer. Metadata is stored in masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way, efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any hardware or software faults.

6.3.2. Data transmission process

This topic describes how data is transmitted when a user accesses OSS to obtain data.

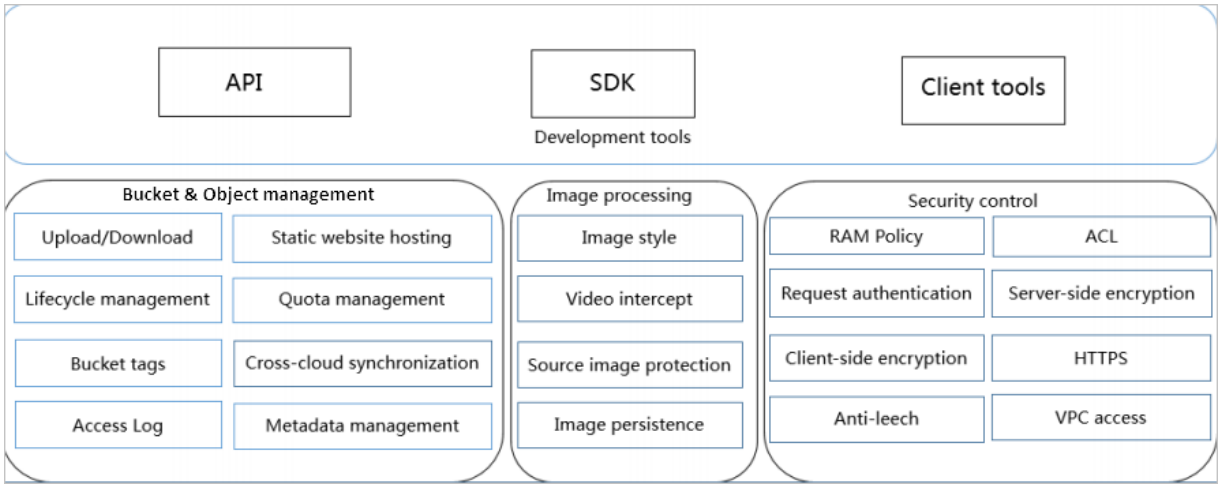
Data is transmitted as follows during the process: `User→RESTful API→SLB-Web server (WS) →Protocol module (PM)→KV Engine→Distributed storage`.

1. A user uses different clients such as browsers or SDKs to initiate a request that complies with the convention of OSS APIs to the OSS endpoint. The endpoint parses the request and sends it to the LVS VIP of SLB. The backend of the LVS VIP is bound to the actual WS. The request is forwarded to one of the WSs.
2. The PM parses the user request.The specific process is as follows: First, the request is authenticated. If the request fails the authentication, the corresponding error code is returned.
3. If the authentication succeeds, the request is parsed as the read/write operation on KV Engine and enters the partition layer.
4. The partition layer uses keys to query and store structured data. This layer also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point.
5. The data stored in KV Engine of the partition layer is written to the persistent storage layer.
6. The large-scale distributed file system is deployed at the persistent storage layer. Metadata is stored in masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way, efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any hardware or software fault.

6.4. Features and principles

6.4.1. Component

OSS is a storage solution that is built on the Apsara system. This topic describes the components of OSS.



OSS consists of the following three modules:

- Access layer: APIs, SDKs, and Apsara Stack Management Console
- Application layer: buckets, object management, IMG, and security modules
- Infrastructure layer: Apsara Distributed File System, Job Scheduler, and Apsara Name Service and Distributed Lock Synchronization System

6.4.2. Features

Object Storage Service (OSS) is a secure, cost-effective, and highly reliable cloud storage service provided by Alibaba Cloud. It enables you to store a large amount of data in the cloud. This topic describes the features provided by OSS.

Bucket and object management

- Bucket overview

  All buckets of the requester are displayed. By default, if you use HTTP to access an OSS endpoint, all of your buckets are displayed.

- Create a bucket

  By default, you can create a maximum of 100 buckets. Bucket names must comply with the naming rules.

  You may encounter the following scenarios when you create a bucket:

  - If the bucket you want to create does not exist, the system creates a bucket of a specified name and returns a flag, indicating that the bucket is created.
  - If the bucket you want to create exists and the requester is the original bucket owner, the original bucket is retained and a flag is returned, indicating that the bucket is created.
  - If the bucket you want to create exists and the requester is not the original bucket owner, a flag is returned, indicating that the bucket fails to create.

- Delete a bucket

  If you want to delete a bucket, ensure that the following conditions are met:

  - The bucket exists.
  - You have the permissions to delete the bucket.
  - The bucket is empty.

- List all objects in a bucket

  To list all objects in a specified bucket, you must have the corresponding operation permissions on the bucket. If the specified bucket does not exist, an error message is returned.

  OSS allows you to search for buckets by prefix and configure the number of objects that can be returned for each search. The maximum number of objects that can be returned for each search is 1,000.

- Upload an object

  You can upload an object to a specified bucket. You can upload objects to a bucket if the bucket exists and you have the corresponding operation permissions on the bucket. If the object you want to upload has the same name as that of an existing object in the bucket, the new object will overwrite the original object.

- Delete an object

  You can delete a specified object if you have the corresponding operation permissions on the object.

- Obtain an object

  To obtain the content or metadata of an object, you must have the corresponding operation permissions on the object.

- Access an object

  OSS allows you to use a URL to access an object.

Security control

- Configure and query the ACL of a bucket

  You can configure and view the ACL of a bucket. You can configure one of the following ACLs for a bucket:

  - Private: Only the owner or authorized users of this bucket can read and write objects in the bucket. Other users, including anonymous users cannot access the objects in the bucket without authorization.
  - Public read: Only the owner or authorized users of this bucket can write objects in the bucket. Other users, including anonymous users can only read objects in the bucket.
  - Public read/write: Any users, including anonymous users can read and write objects in the bucket.

- Access logging and monitoring

  You can enable logging for a bucket. After you enable this feature, OSS pushes the access logs on an hourly basis. You can view information such as buckets, traffic, and requests on the Object Storage Service homepage in the Apsara Stack Cloud Management (ASCM) console.

- Hotlink protection

  OSS provides hotlink protection to prevent other domain names from accessing your data in OSS. You can configure the Referer field in the HTTP header to implement hotlink protection. You can configure a Referer whitelist for a bucket and configure whether to allow access requests that have an empty Referer field in the OSS console. For example, you can add http://www.aliyun.com to the Referer whitelist for a bucket named oss-example. Then, requests whose Referer field is set to http://www.aliyun.com can access the objects in the oss-example bucket.

6.4.3. Terms

This topic describes several basic terms used in OSS.

Object

The basic unit for data operations in OSS. Objects are also known as OSS files. An object is composed of object metadata, object content, and a key. A key can uniquely identify an object in a bucket. Object metadata is a group of key-value pairs that define the properties of an object, such as the last modification time and the object size. You can also assign user metadata to the object.

The lifecycle of an object starts when the object is uploaded, and ends when it is deleted. During the lifecycle, the object cannot be modified. OSS does not support modifying objects. If you want to modify an object, you must upload a new object with the same name as the existing object to replace it.

> **Note** Unless otherwise stated, objects and files mentioned in OSS documents are collectively called objects.

Bucket

A container for OSS objects. Each object in OSS is contained in a bucket. You can configure and modify the attributes of a bucket to manage ACLs and lifecycle rules of the bucket. These attributes apply to all objects in the bucket. Therefore, you can create different buckets to meet different management requirements.

- OSS does not use a hierarchical structure for objects, but instead uses a flat structure. All elements are stored as objects in buckets. However, OSS supports folders as a concept to group objects and simplify management.
- You can create multiple buckets.
- A bucket name must be globally unique within OSS. Bucket names cannot be changed after the buckets are created.
- A bucket can contain an unlimited number of objects.

Strong consistency

A feature requires that object operations in OSS be atomic, which indicates that operations can only either succeed or fail. There are no intermediate states. To ensure that users can access only complete data, OSS does not return corrupted or partial data.

Object-related operations in OSS are highly consistent. For example, when a user receives an upload (PUT) success response, the uploaded object can be read immediately, and copies of the object have been written to multiple devices for redundancy. Therefore, there are no situations where data is not obtained when you perform the read-after-write operation. The same is true for delete operations. After you delete an object, the object and its copies no longer exist.

Similar to traditional storage devices, modifications are immediately visible in OSS while consistency is guaranteed.

Comparison between OSS and file systems

OSS is a distributed object storage service that stores objects based on key-value pairs. You can retrieve object content based on unique object keys. For example, object name *test1/test.jpg* does not necessarily indicate that the object is stored in a directory named test1. In OSS, *test1/test.jpg* is only a string. There is nothing essentially different between test1/test.jpg and *a.jpg*. Therefore, similar amounts of resources are consumed regardless of which object you access.

A file system uses a typical tree index structure. To access a file named *test1/test.jpg*, you must first access the test1 directory and then search for the *test.jpg* file in this directory. This makes it easy for a file system to support folder operations, such as renaming, deleting, and moving directories because these operations are only performed on directories. However, the performance of a file system depends on the capacity of a single device. The more files and directories that are created in the file system, the more resources and time are consumed.

You can simulate similar folder functions of a file system in OSS, but such operations are costly. For example, if you want to rename the test1 directory as test2, OSS must copy all objects whose names start with test1/ to generate objects whose names start with test2. This operation consumes a large amount of resources. Therefore, we recommend that you do not perform such operations in OSS.

Objects stored in OSS cannot be modified. A specific API operation must be called to append an object, and the generated object is different from objects uploaded by using other methods. To modify even a single byte, you must upload the entire object again. A file system allows you to modify files. You can modify the content at a specified offset location or truncate the end of a file. These features make file systems suitable for more general scenarios. However, OSS supports a large amount of concurrent access, whereas the performance of a file system is subject to the performance of a single device.

We recommend that you do not map operations on OSS objects to file systems because it is inefficient. If you attach OSS as a file system, we recommend that you only add new files, delete files, and read files. You can make full use of OSS advantages, such as the capability to process and store large amounts of unstructured data such as images, videos, and documents.

7.Apsara File Storage NAS

7.1. What is NAS?

7.1.1. Overview

Apsara File Storage NAS is a cloud service that provides file storage for compute nodes. These compute nodes include Elastic Compute Service (ECS) instances and Alibaba Cloud Container Service for Kubernetes (ACK) nodes.

NAS is a distributed file system that provides multiple benefits. These benefits include parallel shared access, auto scaling, high availability, and high reliability. Based on POSIX file APIs, NAS is compatible with native operating systems. This ensures data consistency and exclusive locks during shared access.

NAS provides scalable file systems and allows simultaneous access to a file system from multiple ECS instances. The storage capacity of the file system scales up or down when you add or remove files. NAS provides shared data sources for workloads and applications that run on multiple ECS instances or servers.

7.1.2. Benefits

This topic describes the benefits of Apsara File Storage NAS.

NAS has the following benefits:

- Parallel shared access

  Each file system can be mounted by a maximum of 10,000 clients at the same time. The file system shares data from the same data source by using the NFSv3 or NFSv4 protocol.

- High throughput

  When your data storage increases, NAS file systems provide a higher throughput to meet your demands. You do not need to purchase high-end NAS storage devices. This reduces a large amount of upfront investment.

- Auto scaling

  The storage capacity of a NAS file system scales with increasing or decreasing business data. Each file system can provide a maximum of 10 PB storage capacity and store a maximum of 1 billion files. The maximum size of a single file is 32 TB.

- High reliability

  Apsara File Storage NAS is based on Apsara Distributed File System. NAS maintains three copies for each data file across multiple storage nodes. This ensures data security of users.

- High security

  You can isolate your data by using VPCs, security groups, access control lists (ACLs), and RAM users.

- Global namespaces

  Data of a file system is stored on distributed nodes across the entire NAS cluster. This provides a unique namespace.

7.1.3. Scenarios

This topic describes the scenarios of Apsara File Storage NAS.

Scenario 1: shared storage and high availability for SLB

For example, assume that your Server Load Balancing (SLB) instance is connected to multiple Elastic Compute Service (ECS) instances. You can store the data of the applications on these ECS instances on a shared NAS file system. This data sharing method ensures high availability of the SLB instance.

Scenario 2: file sharing within an enterprise

For example, the employees of an enterprise need to access the same datasets. The administrator can create a NAS file system and configure different file or directory permissions for users or user groups.

Scenario 3: data backup

For example, you want to migrate your data from a data center to the cloud for backup. You want to use a standard interface to access the cloud storage service. You can back up your data in a NAS file system.

Scenario 4: server logs sharing

For example, you want to store the application server logs of multiple compute nodes to a shared file store. You can store these server logs in a NAS file system for centralized log processing and analysis.

7.2. Technical advantages

NAS has technical advantages in shared access and data security.

Shared access

NAS supports the standard NFS and SMB protocols and mainstream operating systems, such as Linux and Windows. You can mount NAS file systems on these operating systems.

Multiple compute instances can share access to the same data source. This guarantees strong data consistency.

Data security

NAS encrypts data during data transfer. This ensures data security.

NAS only allows access to a file system from dedicated networks, such as VPCs and VPNs. This ensures access security.

NAS saves multiple data copies and provides flexible backup policies to ensure data security.

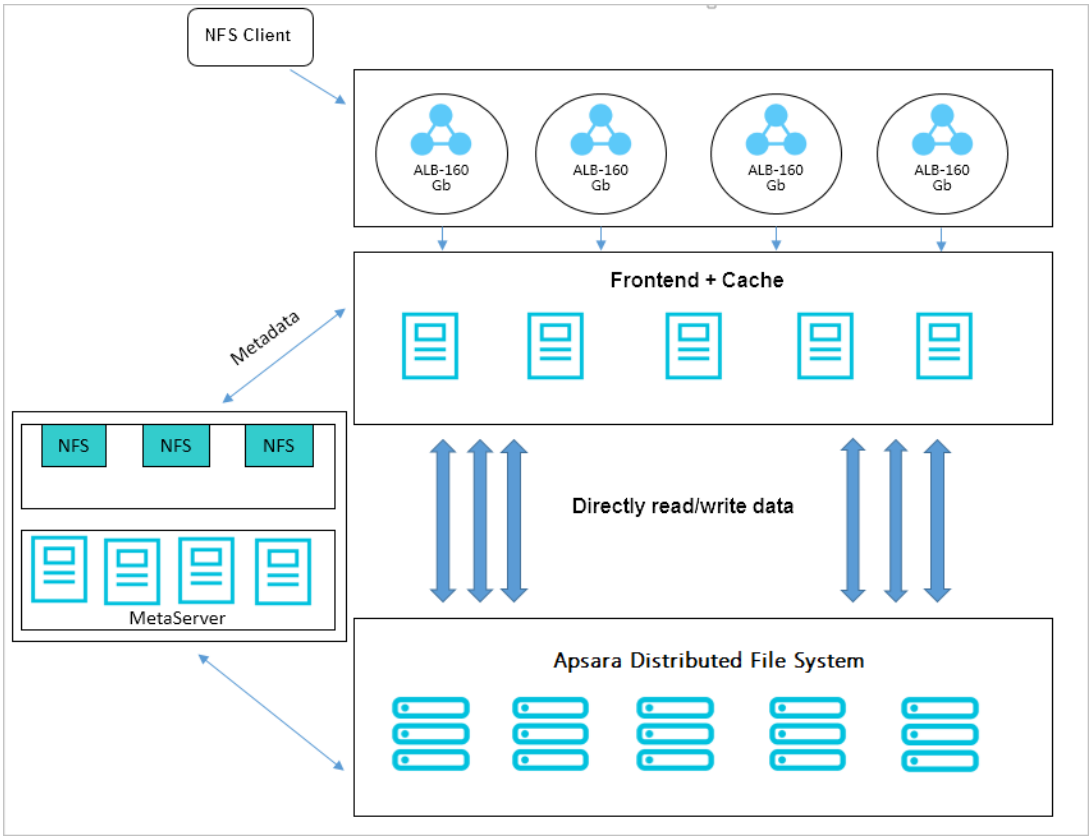
7.3. Architecture

Apsara File Storage NAS is based on Apsara Distributed File System. NAS maintains three copies for each data file across multiple storage nodes. Frontend nodes receive and cache connection requests from NFS clients. Frontend nodes are highly available because they are stateless and distributed.

The metadata of a NAS file system is stored on a MetaServer. When frontend nodes retrieve metadata from the MetaServer by using I/O requests, user data is read from and written to the backend nodes of Apsara Distributed File System.

The system architecture provides separate auto scaling of frontend and backend storage nodes. This ensures high availability, high concurrency, and low latency.

System architecture



7.4. Features and principles

7.4.1. Feature overview

NAS supports the NFSv3 and NFSv4 protocols. You can use NAS without making any changes to your existing applications. You can use either protocol to access NAS instances for the following purposes: business file sharing, backend file storage for office automation systems, enterprise database backup and storage, business system log storage and analysis, website data storage and distribution, and data storage during business system development and testing.

Features

7.4.2. Features

This topic describes the features of Apsara File Storage NAS.

Seamless integration

Apsara File Storage NAS supports the NFSv3 and NFSv4 protocols. NAS also allows you to access data by using standard file system interfaces. Mainstream applications and workloads can be seamlessly integrated with NAS without the need for modification.

Shared access

If you want to access a data source from multiple Elastic Compute Service (ECS) and Elastic Container Instance (ECI) instances, a NAS file system meets this need.

Access control

Apsara File Storage NAS uses multiple security mechanisms to guarantee system data security. These security mechanisms include network isolation based on VPCs, user isolation in classic networks, standard permission control for file systems, access control based on security groups, and RAM user authorization.

Scalable performance

Apsara File Storage NAS provides your applications with optimal storage performance, such as high throughput, high IOPS, and low latency. The storage performance linearly improves as the storage capacity increases. This meets your demands for higher storage performance as your business grows.

7.4.3. Terms

This topic describes the basic terms of Apsara File Storage NAS.

mount target

A mount target is the access address of a NAS file system in a VPC or classic network. Each mount target corresponds to a domain name. To mount a NAS file system to a local directory, you must specify the domain name of the mount target.

permission group

The permission group mechanism is a whitelist mechanism provided by NAS. You can add rules to a permission group of a NAS file system. You can allow users from specified IP addresses or CIDR blocks to access the NAS file system by using different permissions.

> **Note** Each mount target must be associated with a permission group.

authorized object

An authorized object is an attribute of a permission group rule. It specifies the IP address or CIDR block to which the permission group rule is applied. In a VPC, an authorized object can be a single IP address or a CIDR block. In a classic network, an authorized object must be a single IP address. In most cases, this IP address is the internal IP address of an Elastic Compute Service (ECS) instance.

8.Tablestore

8.1. What is Tablestore?

8.1.1. Technical background

This topic describes the data features in the data technology (DT) era and the challenges of traditional IT software solutions for you to better understand the technical background of Tablestore.

Data features in the DT era

As the mobile Internet becomes more common and widely adopted in various industries and fields, Internet applications present the following significant features and trends:

- The amount of data that needs to be stored and processed increases exponentially. The data includes microblogs, social events, pictures, and access logs.
- As the use of mobile and Internet of Things (IoT) devices increase, the requirements for concurrent writes of structured data storage also increase.
- The data has loose schemas and tends to be semi-structured, and data fields change dynamically.
- User access features hot spots and peak hours. For example, during promotional activities, user access soars within a few minutes.
- The mobile Internet allows users to connect to Internet applications at any time. Service instability caused by failures or even planned service failures greatly affects user experience. Therefore, high availability is required.
- Large amounts of data increase the requirements for the performance and scale of computing and analysis.

Challenges to traditional IT software solutions

Traditional IT software solutions face the following trends and challenges:

- Scalability

  Traditional software such as relational databases are incapable of handling such fast-growing data. It bottlenecks data write throughput and access efficiency. In traditional database solutions, databases and tables are manually and statically partitioned. This method requires large amounts of maintenance. In particular scenarios where nodes are added to increase the storage capacity, you must repartition and migrate existing data. During this process, it is difficult to guarantee service performance, stability, and availability. The whole process is complex.

- Data model changes

  Data in traditional databases is processed based on a schema. The number of columns for data storage is fixed and seldom modified. Frequent changes to the table schema and column count affect service availability. Therefore, traditional solutions are incapable of handling the increasing volumes of loosely structured data from Internet applications.

- Quick scaling

In traditional solutions, business access loads are stable, and the system is not required to scale resources in a short time. When resources need to be scaled, a large amount of labor is required to reparation and migrate data. Then, when business loads decline, the hosts added during scaling must be removed to avoid low resource usage, and data must be migrated again. This process is complex and inefficient.

- O&M guarantees

  In traditional software solutions, services are recovered when hardware (network devices or disks) failures occur. You must manually replace hardware, upgrade software, and configure tuning and updates. To ensure that applications are not aware of these processes and avoid deterioration of service availability, a special engineering team is required to implement O&M. Therefore, workloads caused from recruitment and fund investment bring a huge challenge to fast-developing enterprises.

- Computing bottlenecks

  The current business system uses Online Transaction Processing (OLTP) to process and analyze data in relational databases such as MySQL and Microsoft SQL Server. These relational databases are used to process transactions. Consistency and atomicity are maintained while data is frequently inserted and modified. However, if the amount of data that needs to be queried or calculated is too large, such as tens of millions or even billions of records of data, or if the computing is complex, the OLTP databases cannot meet the requirements.

8.1.2. Tablestore technologies

To improve scalability, Tablestore partitions tables and schedules data partitions to different nodes. When hardware failures occur on a single server, Tablestore uses heartbeat mechanism to find the node where failures occur. The partition in the node is migrated to a normal node, and the service is recovered and continues.

Data partitioning and load balancing

The first column of a primary key in each row of a table is the partition key. The system splits a table into multiple partitions based on the value of the partition key. These partitions are evenly scheduled across different storage nodes. When the data in a partition exceeds the limit size, the partition is split into two smaller partitions. The data and access loads are distributed to these two partitions. The partitions are scheduled to different nodes. As a result, access loads are scattered to different nodes. Eventually, the single-table data scale and access loads can be linearly scaled.

Technical indicator: Tablestore can store petabytes of data in a single table and allows you to simultaneously read/write millions of data.

Automatic recovery from single points of failure (SPOFs)

Each node in the storage engine of Tablestore provides services for multiple data partitions of different tables. The master node manages partition distribution and scheduling, and also monitors the health of each service node. If a service node fails, the master node migrates data partitions from the faulty node to other healthy nodes. The migration is logically performed, and does not involve physical entities. Therefore, services can rapidly recover from SPOFs.

Technical indicator: SPOFs affect services of only a part of data partitions and services can recover within single-digit minutes.

Zone-disaster recovery and geo-disaster recovery

To meet business security and availability requirements, Tablestore provides zone-disaster recovery and geo-disaster recovery based on primary and secondary clusters. Disaster recovery supports instance-based recovery. Any table operation on the primary instance, including insertion, update, or deletion, is synchronized to the table of the same name in the secondary instance. The duration of data synchronization between the primary and secondary instances depends on the network environment of the primary and secondary clusters. In the ideal network environment, the synchronization latency is within single-digit milliseconds. Before the manual failover, you must stop resource access to the primary cluster and wait for all data to be completely backed up. You can perform only one failover in an hour. After the failover, data in the original cluster is deleted, and a secondary cluster is configured.

In zone-disaster recovery based on primary and secondary clusters, the endpoints remain unchanged when applications access Tablestore in the primary and secondary clusters. In other words, the application endpoints do not need to be changed after the failover. In geo-disaster recovery based on primary and secondary clusters, the endpoints of the primary and secondary clusters are different. After the failover, endpoints need to be changed for applications.

Technical indicator: The RTO of Tablestore is smaller than 2 minutes, the RPO is smaller than 5 minutes, and the RCO is 1.

8.2. Benefits

Tablestore provides the following benefits:

Scalability

- Tablestore imposes no upper limit on the amount of data that can be stored in tables. When the amount of data increases, Tablestore adjusts partitions to provide more storage space for tables and improve the capability of handling access request bursts.
- Tablestore supports CPUs, disks, memory, and NICs of different specifications in a single-component cluster without affecting cluster running performance. This ensures maximum compatibility with existing devices.

High performance

If you use a high-performance instance, its average access latency of single rows is measured in single-digit milliseconds. The read/write performance is not affected by the size of data in a table.

Data reliability

- Tablestore provides high data reliability. It stores multiple data copies and restores data when any of the copies become invalid.
- Tablestore supports automatic fault tolerance for disk failures of servers in a cluster, and supports hot swapping of disks. The failures of a single disk do not affect the overall service. The failures of data disks do not affect the service of the server. If Redundant Array of Independent Disks (RAID) is not created and the system disk is damaged, the server is removed from the cluster. After the hardware failures are fixed, services can be restored within two minutes.
- Tablestore supports full or incremental backup and data recovery from storage.
- Tablestore supports the backup between data clusters in different data centers. The backup process is visualized.

High availability

Tablestore uses automatic failure detection and data migration to shield applications from host- and network-related hardware faults, which provides high availability for your applications.

Ease of management

- Tablestore automatically performs complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-out.
- You can store audit logs to Log Service and download logs from Log Service. This facilitates long-term storage and management of audit logs.

Access security

- Tablestore provides multiple permission management mechanisms. It verifies and authenticates the identity of each application request to prevent unauthorized data access, which improves the data security.
- Tablestore supports the management of data access permissions, including logon permissions, table creation permissions, read and write permissions, and whitelist-related permissions.
- Tablestore allows you to use the Apsara Stack Cloud Management (ASCM) console to manage administrative permissions, including administrator classification. You can use the ASCM console to manage user permissions in a centralized manner. You can manage the access control features of any component within the system. You can also block common users from querying access control details and simplify access control for administrators to improve the usability of access control.

Strong consistency

Tablestore ensures high data consistency for data writes. After a write operation succeeds, three replicas are written to a disk. Applications can read the latest data immediately.

Flexible data models

Tablestore tables do not require a fixed format. Each row can contain a different number of columns. Tablestore supports multiple data types, including Integer, Boolean, Double, String, and Binary.

Monitoring integration

You can log on to the Tablestore console to obtain monitoring information in real time, including the requests per second and average response latency.

Multi-tenancy

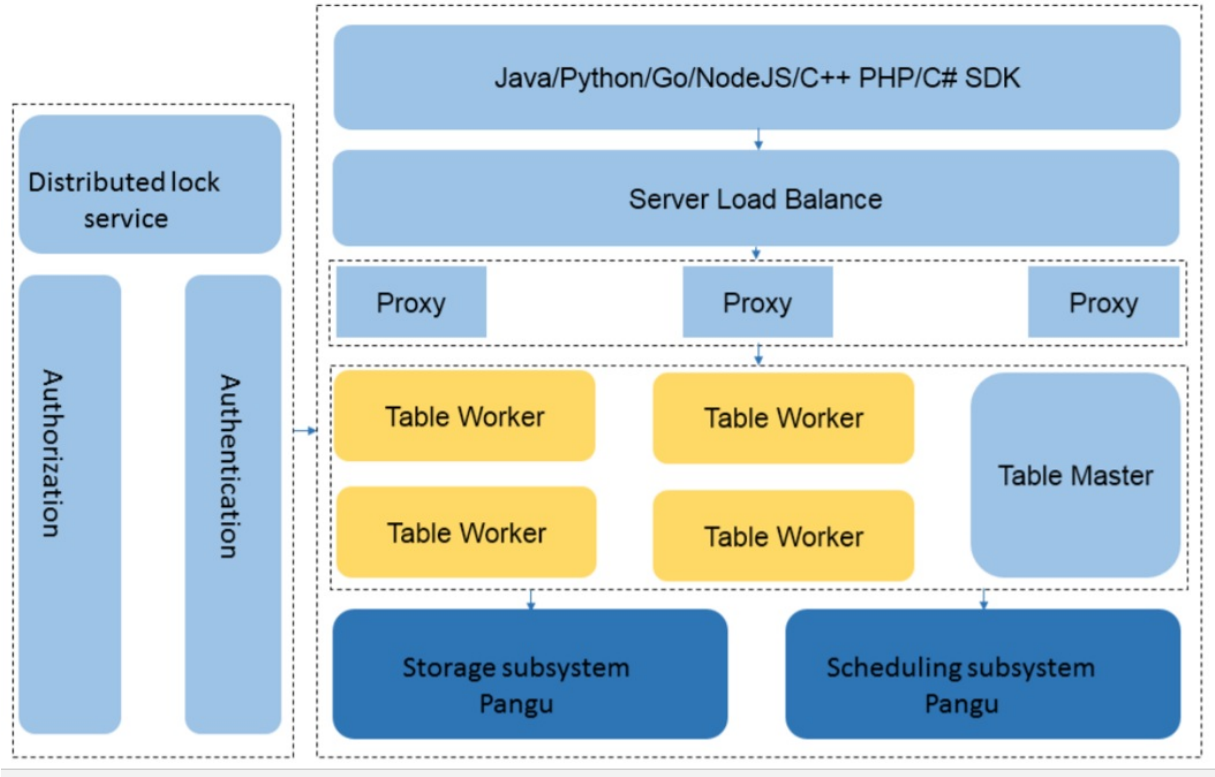
- Isolation: allows tasks of multiple tenants (projects) to be submitted to different queues and run separately. Resources are isolated among tenants.
- Permission: allows you to manage tenants in a centralized manner, dynamically configure and manage tenant resources, isolate resources, view statistics for resource usage, and manage tenants at multiple levels in the console.
- Scheduling: supports multi-tenant scheduling of multiple clusters and multiple resource pools.

8.3. Architecture

This topic describes the Tablestore architecture.

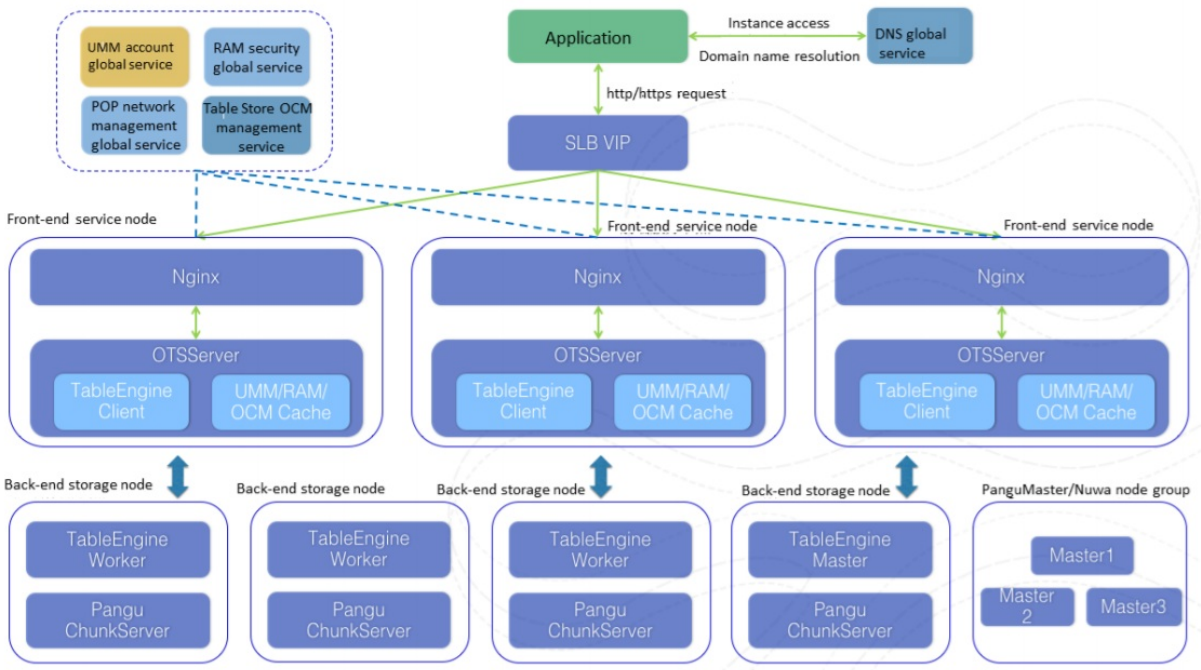
The architecture of Tablestore is referenced from Bigtable (one of the three core technologies of Google) and uses the log-structured merge-tree (LSM) storage engine to provide high write performance. The performance of primary key-based single-row queries and range queries is stable and predictable. The performance is not affected by the volume of data and access concurrency.

The following figure shows the basic architecture of Tablestore.



- The top layer is the protocol access layer. Server Load Balancer (SLB) distributes user requests to various proxy nodes. The proxy nodes receive requests that are sent by using the RESTful protocol and implement security authentication.
  - If the authentication succeeds, the user requests are forwarded to the corresponding data engine based on the value of the first primary key column for further operations.
  - If the authentication fails, error information is returned to the user.
- Table Worker is the data engine layer that processes structured data. It uses a primary key to search for or store data. Table Worker supports large-scale access request bursts.
- The bottom layer is the persistent storage layer. Apsara Distributed File System is deployed at this layer. Metadata is stored on masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way, efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any hardware or software fault.

The following figure shows the detailed architecture of Tablestore.

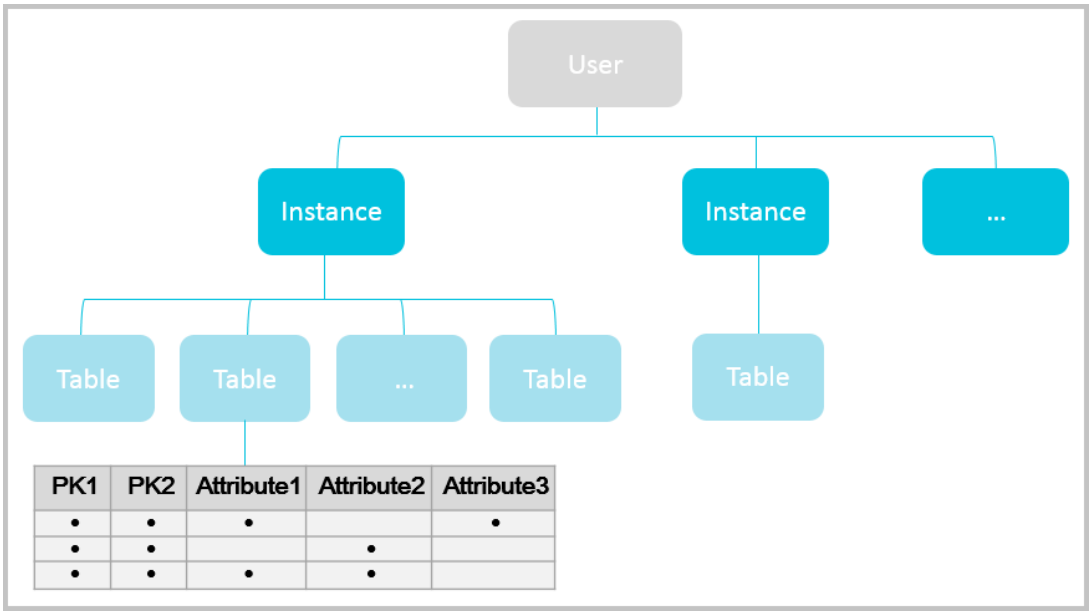


8.4. Features

8.4.1. Users and instances

This topic describes the architecture of users and instances.

The following figure shows the Tablestore architecture in relation to a user and instances.



- Users can log on with an Apsara Stack tenant account.
- User operations can be audited in fine granularity.
- Users organize resources based on instances. A user can create multiple instances and use each

instance to create and manage multiple data tables.

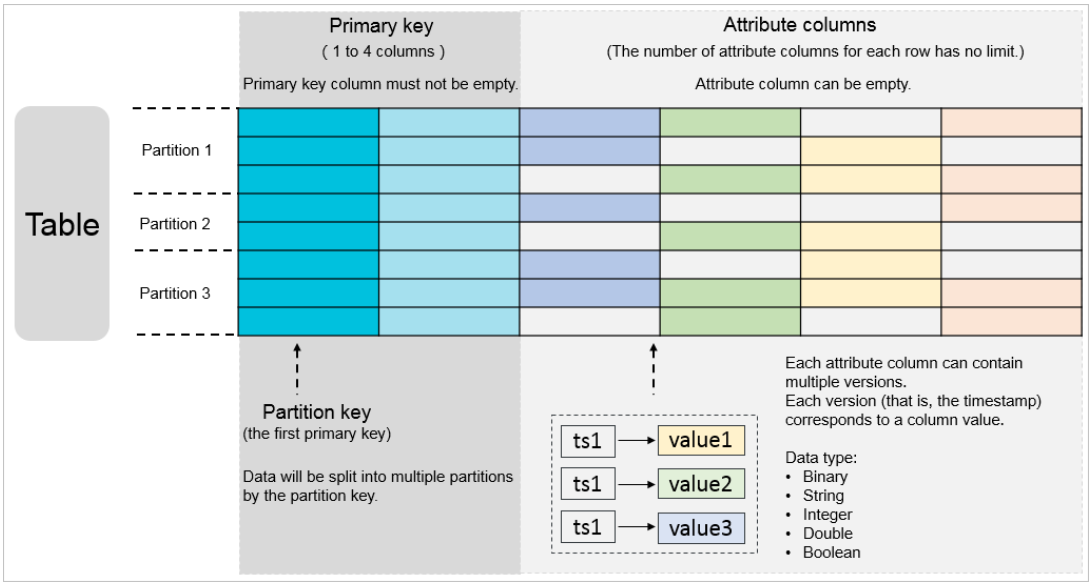
- An instance is the basic unit of multi-tenant isolation.
- User permissions can vary based on their roles.

8.4.2. Data tables

This topic describes the structure of data tables.

The following figure shows the data table structure.

Data table structure



- A data table is the basic unit of resource allocation.
- A table is a collection of rows. A row consists of primary key columns and attribute columns.
- A table partitions data based on the value of the first primary key column.
- All rows in a table must have the same quantity of primary key columns that share the same names.
- The quantity, names and data types of attribute columns in a row can be different.
- The number of attribute columns contained in a row is not limited. However, the maximum number of attribute columns that can be written in each request is 1,024.
- A table can contain at least hundreds of billions of rows of data.
- A table can store petabytes of data.

8.4.3. Data partitioning

This topic describes the features of data partitioning.

- A table partitions data based on the value of the first primary key column.
- The rows whose first primary key column values are within the same partition key value range are allocated to the same partition.
- To improve load balancing, Tablestore splits and merges partitions based on specific rules.
- We recommend that you do not store more than 10 GB of data in rows that share the same partition key.

8.4.4. Common commands and functions

This topic describes common commands used to manage tables and common functions used to manage data in tables.

Common commands used to manage tables

- ListTable: lists all tables in an instance.
- CreateTable: creates a table.
- DeleteTable: deletes a table.
- DescribeTable: queries the attributes of a table.
- UpdateTable: updates the reserved read/write throughput configuration of a table.
- ComputeSplitPointsBySize: logically partitions all table data into multiple partitions of a specified size, and returns the split points between these partitions and the prompt of the hosts where partitions reside.

Common functions used to manage data in tables

- GetRow: reads data from a single row.
- PutRow: inserts a row of data.
- UpdateRow: updates a row of data.
- DeleteRow: deletes a row of data.
- BatchGetRow: reads multiple rows in one or more tables simultaneously.
- BatchWriteRow: inserts, updates, or deletes multiple rows in one or more tables.
- GetRange: reads reads a range of data from a table.

8.4.5. Authorization and access control

This topic describes permissions for Tablestore and the operations you can perform in the Apsara Stack Cloud Management (ASCM) console.

Tablestore permissions

Tablestore integrates RAM and VPC to support the following access control mechanisms:

- Table-level authorization
- Operation-level access control
- Authentication based on IP address limits, HTTPS, multi-factor authentication (MFA), and access time limits
- Temporary access authorization of STS
- VPC-based access control

Operations in the ASCM console

- Account logons and authentication.
- Instance creation, management, and deletion in GUI.
- Table creation, management, deletion, and reserved read/write throughput adjustment in GUI.
- Display table-level monitoring information.

9.ApsaraDB for RDS

9.1. What is ApsaraDB for RDS?

ApsaraDB for RDS is a stable, reliable, and scalable online database service. Based on the distributed file system and high-performance storage, ApsaraDB for RDS allows you to perform database operations and maintenance with its set of solutions for disaster recovery, backup, restoration, monitoring, and migration.

ApsaraDB for RDS supports four storage engines, which are MySQL, SQL Server, PolarDB, and PostgreSQL. You can create database instances based on these storage engines to meet your business requirements.

RDS MySQL

Originally based on a branch of MySQL, ApsaraDB RDS for MySQL provides excellent performance. It is a tried and tested solution that handled the high-volume concurrent traffic during Double 11. ApsaraDB RDS for MySQL provides basic features, such as whitelist configuration, backup and restoration, Transparent Data Encryption (TDE), data migration, and management for instances, accounts, and databases. ApsaraDB RDS for MySQL also provides the following advanced features:

- **Read-only instance:** In scenarios where ApsaraDB for RDS handles a small number of write requests but a large number of read requests, you can create read-only instances to scale up the reading capability and increase the application throughput.
- **Read/write splitting:** The read/write splitting feature provides an extra read/write splitting endpoint. This endpoint enables an automatic link for the primary instance and all its read-only instances. An application can connect to the read/write splitting endpoint to read and write data. Write requests are automatically distributed to the primary instance while read requests are distributed to read-only instances based on their weights. To scale up the reading capacity of the system, you can add more read-only instances.

RDS SQL Server

ApsaraDB RDS for SQL Server provides strong support for a variety of enterprise applications under the high-availability architecture, has the capability of restoring data to any point in time.

ApsaraDB RDS for SQL Server provides basic features such as whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases.

PolarDB

PolarDB is a stable, secure, and scalable enterprise-class relational database. Based on PostgreSQL, PolarDB enhances performance, application solutions, and compatibility. It also provides the capability of directly running Oracle applications. You can run various enterprise applications on PolarDB stably at low costs.

PolarDB provides features such as account management, resource monitoring, backup and restoration, and security control. These features are under continuous improvement, and more features are under development to adapt to PolarDB.

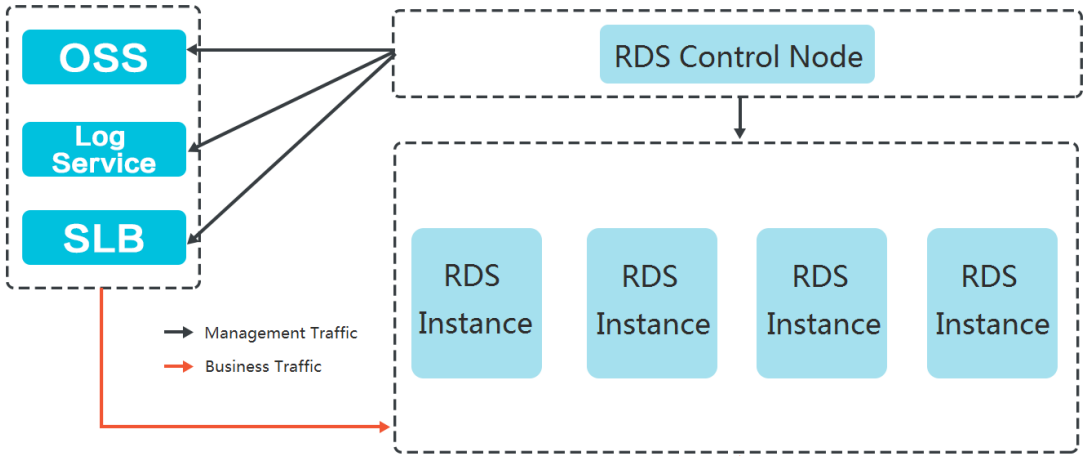
RDS PostgreSQL

ApsaraDB RDS for PostgreSQL is an advanced open source database that is fully compatible with SQL and supports a diverse range of data formats such as JSON, IP, and geometric data. In addition to support for features such as transactions, subqueries, multi-version concurrency control (MVCC), and data integrity check, ApsaraDB RDS for PostgreSQL integrates a series of features including high availability, backup, and restoration to ease operations and maintenance loads.

9.2. Architecture

The following figure shows the system architecture of ApsaraDB for RDS.

RDS system architecture

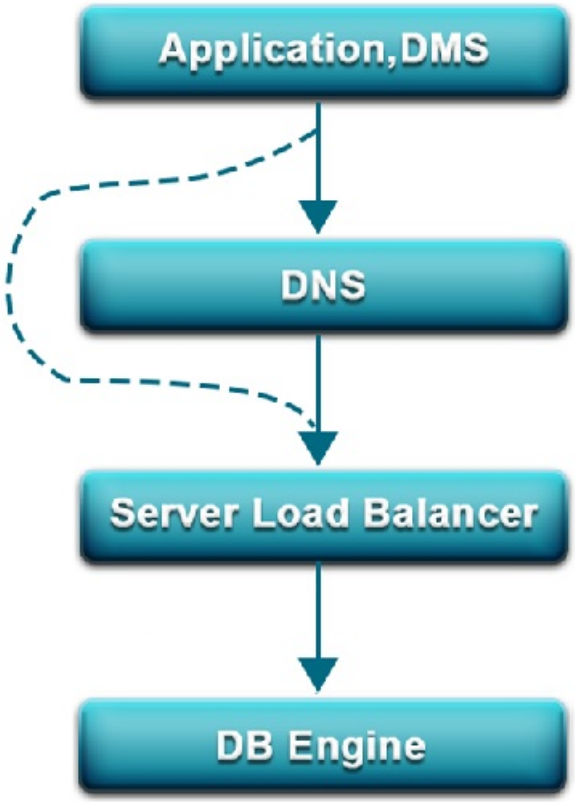


9.3. Features

9.3.1. Data link service

The data link service allows you to add, delete, modify, and query the table schema and data.

ApsaraDB for RDS data link service



DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore, IP address changes do not affect the performance of ApsaraDB for RDS instances. For example, assume that the domain name of an ApsaraDB for RDS instance is test.rds.aliyun.com, and its corresponding IP address is 10.1.1.1. The instance can be accessed when either test.rds.aliyun.com or 10.1.1.1 is configured in the connection pool of a program.

After a zone migration or version upgrade is performed for this ApsaraDB for RDS instance, the IP address may change to 10.1.1.2. If the domain name test.rds.aliyun.com is configured in the connection pool, the instance can still be accessed. However, if the IP address 10.1.1.1 is configured in the connection pool, the instance is no longer accessible.

SLB

The Server Load Balancer (SLB) module provides both the internal and public IP addresses of an ApsaraDB for RDS instance. Therefore, server changes do not affect the performance of the instance. For example, assume that the internal IP address of an RDS instance is 10.1.1.1, and the corresponding Proxy or DB Engine runs on 192.168.0.1. The SLB module typically redirects all traffic destined for 10.1.1.1 to 192.168.0.1. If 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 will take over for the initial server. In this case, the SLB module will redirect all traffic destined for 10.1.1.1 to 192.168.0.2, and the RDS instance will continue to provide services normally.

DB Engine

The following table describes the mainstream database protocols supported by ApsaraDB for RDS.

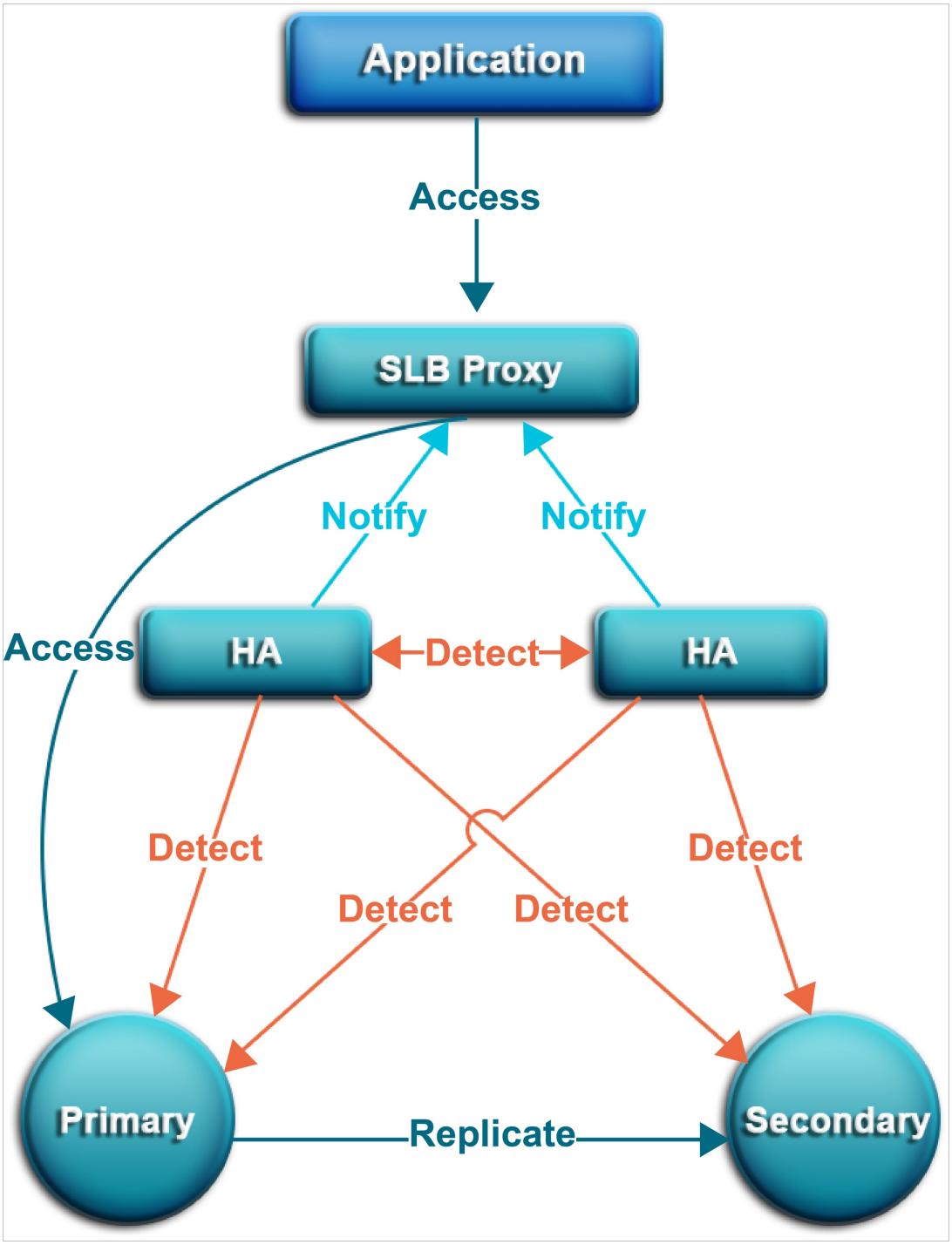
ApsaraDB for RDS database protocols

RDBMS	Version
MySQL	5.6 and 5.7
SQL Server	2012 and 2016
PostgreSQL	9.4 and 10
PolarDB	11

9.3.2. High-availability service

The high-availability (HA) service ensures the availability of data link services and processes internal database exceptions. The HA service is implemented by multiple HA nodes.

RDS HA service



Detection

The Detection module checks whether the primary and secondary nodes of the DB Engine are providing services normally.

The HA node uses heartbeat information taken at 8 to 10 second intervals to determine the health status of the primary node. This information, along with the health status of the secondary node and heartbeat information from other HA nodes, provides a reference for the Detection module. All this information helps the module avoid misjudgment caused by exceptions such as network jitter. Failover can be completed quickly.

Repair

The Repair module maintains the replication relationship between the primary and secondary nodes of the DB Engine. It can also correct errors that occur on either node during normal operations. For example:

- It can automatically restore primary/secondary replication after a disconnection.
- It can automatically repair table-level damage to the primary or secondary node.
- It can save and automatically repair the primary or secondary node in case of crashes.

Notice

The Notice module informs the SLB or Proxy module of status changes to the primary and secondary nodes to ensure that you always access the correct node.

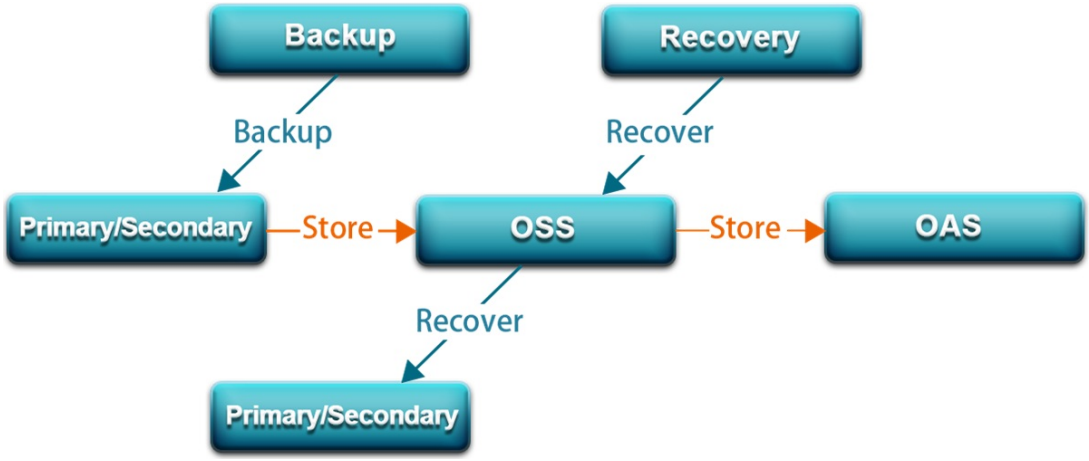
For example, the Detection module discovers problems with the primary node and instructs the Repair module to resolve these problems. If the Repair module fails to resolve a problem, it instructs the Notice module to perform traffic switchover. The Notice module forwards the switching request to the SLB or Proxy module, and then all traffic is redirected to the secondary node.

Meanwhile, the Repair module creates a new secondary node on a different physical server and synchronizes this change back to the Detection module. The Detection module rechecks the health status of the instance.

9.3.3. Backup service

The backup service supports offline data backup, storage, and recovery.

RDS backup service



Backup

The Backup module compresses and uploads data and logs on both the primary and secondary nodes. ApsaraDB for RDS uploads backup files to OSS and dumps the backup files to a more cost-effective and persistent Archive Storage system. When the secondary node is operating normally, backups are always created on the secondary node. This way, the services on the primary node are not affected. When the secondary node is unavailable or damaged, the Backup module creates backups on the primary node.

Recovery

The Recovery module restores backup files from OSS to a destination node. The Recovery module provides the following features:

- Primary node rollback: rolls back the primary node to a specified point in time when an operation error occurs.
- Secondary node repair: creates a new secondary node to reduce risks when an irreparable fault occurs on the secondary node.
- Read-only instance creation: creates a read-only instance from backup files.

Storage

The Storage module uploads, dumps, and downloads backup files.

All backup data is uploaded to OSS for storage. You can obtain temporary links to download backups as necessary.

In certain scenarios, the Storage module allows you to dump backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

9.3.4. Monitoring service

ApsaraDB for RDS provides multilevel monitoring services across the physical, network, and application layers to ensure service availability.

Service

The Service module tracks the status of services. For example, the Service module monitors whether SLB, OSS, and other cloud services on which RDS depends are operating normally. The monitored metrics include functionality and response time. The Service module also uses logs to determine whether the internal RDS services are operating properly.

Network

The Network module tracks statuses at the network layer. The monitored metrics include:

- Connectivity between ECS and RDS
- Connectivity between physical RDS servers
- Rates of packet loss on VRouters and VSwitches

OS

The OS module tracks the statuses of hardware and OS kernel. The monitored metrics include:

- Hardware maintenance: The OS module constantly checks the operating status of the CPU, memory, motherboard, and storage device. It can predict faults in advance and automatically submit repair reports when it determines a fault is likely to occur.
- OS kernel monitoring: The OS module tracks all database calls and analyzes the causes of slow calls or call errors based on the kernel status.

Instance

The Instance module collects the following information about ApsaraDB for RDS instances:

- Instance availability information
- Instance capacity and performance metrics

- Instance SQL execution records

9.3.5. Scheduling service

The scheduling service allocates resources and manages instance versions.

Resource

The Resource module allocates and integrates underlying RDS resources when you enable and migrate instances. When you use the RDS console or an API operation to create an instance, the Resource module calculates the most suitable host to carry traffic to and from the instance. A similar process occurs during ApsaraDB for RDS instance migration.

After repeated instance creation, deletion, and migration operations, the Resource module calculates the degree of resource fragmentation. It also regularly integrates resources to improve the service carrying capacity.

9.3.6. Migration service

The migration service can migrate data from your on-premises databases to ApsaraDB for RDS.

DTS

DTS can migrate data from on-premises databases to ApsaraDB RDS for MySQL without stopping services.

DTS is a data exchange service that streamlines data migration, real-time synchronization, and subscription. DTS is dedicated to implementing remote and millisecond-speed asynchronous data transmission in various scenarios. Based on the active geo-redundancy architecture designed for Double 11, DTS can implement security, scalability, and high availability by providing real-time data streams to up to thousands of downstream applications.

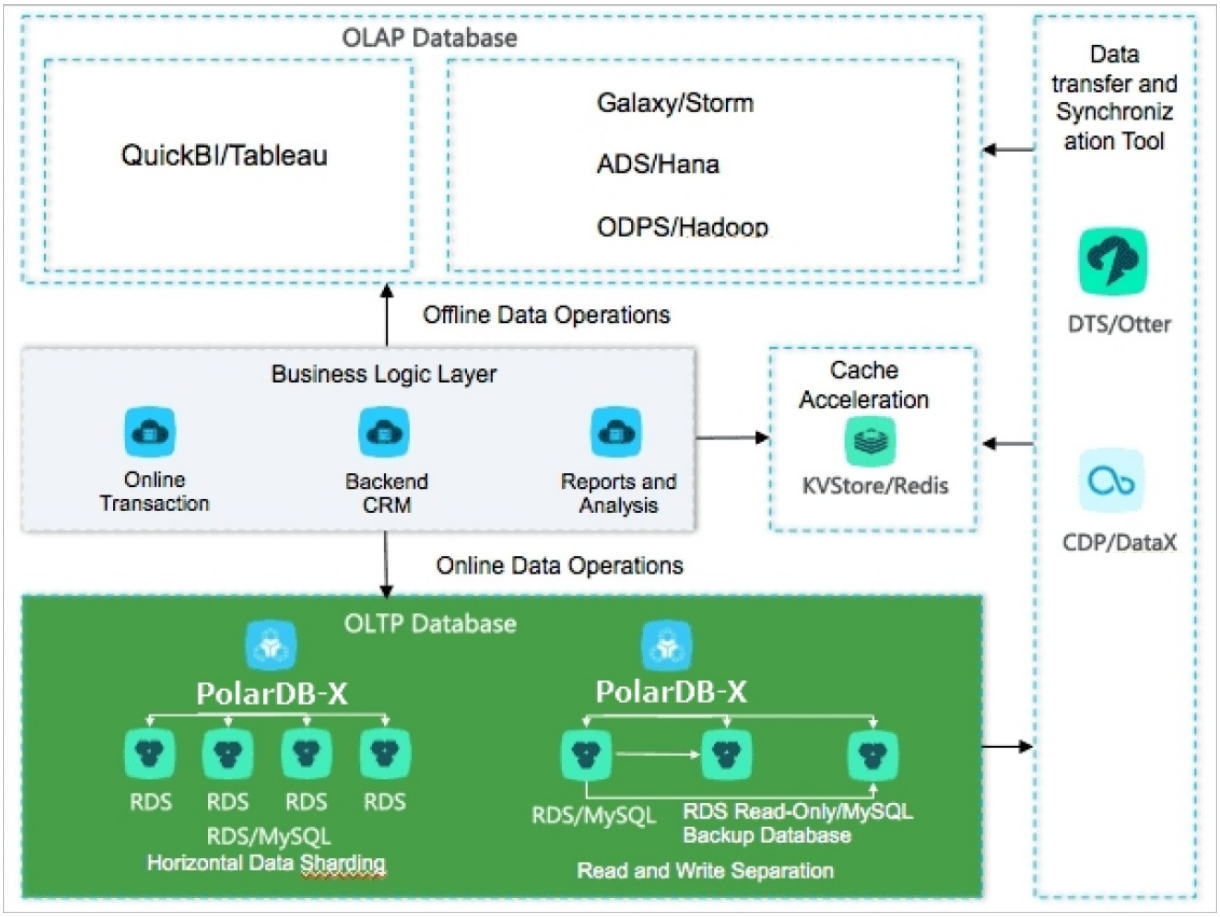
10.Cloud Native Distributed Database PolarDB-X

10.1. What is PolarDB-X?

Cloud Native Distributed Database PolarDB-X is a middleware service independently developed by Alibaba Group for scale-out of single-instance relational databases. It is compatible with Distributed Relational Database Service (DRDS).

PolarDB-X is the standard of relational database access for Alibaba Group. It shares the database sharding logic with Taobao Distributed Data Layer (TDDL). Compatible with the MySQL protocol, PolarDB-X supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It provides the core capabilities of distributed databases, such as database sharding, table sharding, smooth scale-out, configuration changing, and transparent read/write splitting. It is lightweight (stateless), flexible, stable, and efficient, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

PolarDB-X is mainly used for operations on large-scale online data, which focuses on frontend businesses for writing data to databases. By splitting data in specific business scenarios, it maximizes operation efficiency to meet the requirements of high-concurrency and low-latency database operations.



PolarDB-X mainly solves the following problems:

- Capacity bottleneck of single-instance databases: As the data volume and access volume increase, traditional single-instance databases encounter great challenges that cannot be completely solved by hardware upgrades. In distributed database solutions of PolarDB-X, multiple instances work jointly, which effectively resolves the bottlenecks of data storage capacity and access volumes.
- Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, data can be stored to different shards in PolarDB-X through smooth data migration, supporting the dynamic scale-out of relational databases.

10.2. Benefits

Distributed architecture

The distributed architecture of Cloud Native Distributed Database PolarDB-X allows horizontal partitioning of data and the cluster deployment of a single service. In this way, single-instance bottlenecks of Server Load Balancer (SLB), PolarDB-X, and ApsaraDB RDS for MySQL are resolved and service scalability is achieved.

High performance

PolarDB-X for RDS (MySQL) partitions data in specific business scenarios and clusters data based on major business operations, speeding up the response to online transactional operations. PolarDB-X for HiStore uses the columnar storage and knowledge grid to significantly speed up the response to common analytic operations such as large-scale data aggregation and ad hoc queries. It also helps reduce costs by achieving high compression ratio.

Security

PolarDB-X supports an account and permission system similar to that of single-instance databases, and provides useful functions, such as the IP address whitelist and default disabling of high-risk SQL statements. It offers comprehensive API operations for support even if they need to be integrated into the local management system. We also provide complete product support and architecture services.

10.3. Architecture

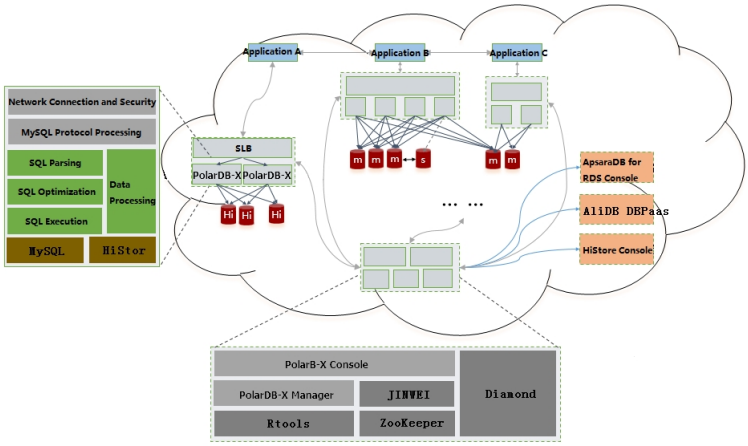
PolarDB-X supports two data output methods: overall output by Apsara Stack and separate output by Alibaba middleware. The two output methods differ in features and the components PolarDB-X depends on.

The following table describes the differences between them.

Item	Overall output by Apsara Stack	Separate output by Alibaba middleware
MySQL	ApsaraDB RDS for MySQL	Alibaba Group database system (DBPaaS)
Load balancing	Centralized Server Load Balancer (centralized SLB)	Client load balancer (VIPServer)
Special storage support	None	Storage with a high compression ratio (HiStore)

The following figure shows the architecture of PolarDB-X.

Figure of PolarDB-X architecture



PolarDB-X Server

PolarDB-X Server is the service layer of PolarDB-X. Multiple server nodes make up a server cluster to provide distributed database services, including the read/write splitting, routed SQL execution, result merging, dynamic database configuration, and globally unique ID (GUID).

ApsaraDB RDS for MySQL (marked by m and s in the figure)

ApsaraDB RDS for MySQL stores data and performs data operations online. It implements high availability through primary/secondary replication. It also implements dynamic database failover with the primary/secondary switchover mechanism.

You can implement management, monitoring, and alerting in the instance lifecycle in the ApsaraDB RDS for MySQL console.

HiStore

When PolarDB-X outputs data separately (not overall output by Apsara Stack), PolarDB-X supports HiStore as the physical storage. HiStore is a low-cost and high-performance database developed by Alibaba to support columnar storage. By using the columnar storage, knowledge grid, and multiple cores, HiStore provides higher data aggregation and ad hoc query capabilities, with lower costs than row storage (such as MySQL).

You can implement management, monitoring, and alerting in the HiStore instance lifecycle in the HiStore console.

DBPaaS

When PolarDB-X outputs data separately (not overall output by Apsara Stack), the MySQL O&M platform DBPaaS implements management, monitoring, and alerting in the MySQL lifecycle.

SLB

You do not need to install a client on user instances. Your requests are distributed through SLB. When an instance fails or a new instance is added, SLB ensures that traffic on the underlying instances is distributed evenly.

VIPServer

You need to install a client on user instances, with a weak dependency on the central controller (interaction is performed only when the load configuration changes). User requests are distributed through VIPServer. When an instance fails or a new instance is added, VIPServer ensures that traffic on the bound instances is distributed evenly.

Diamond

Diamond manages the configuration and storage of PolarDB-X. It provides the configuration functions for storage, query, and notification. In PolarDB-X, Diamond stores the source data of databases, and configuration data including the sharding rules, and PolarDB-X switches.

Data Replication System

Data Replication System migrates and synchronizes data for PolarDB-X. Its core capabilities include full data migration and incremental data synchronization. Its derived features include smooth data import, smooth scale-out, and global secondary index. Data Replication System requires the support of ZooKeeper and PolarDB-X Rtools.

PolarDB-X Console

PolarDB-X Console is designed for business database administrators (DBAs) to isolate resources and operations based on users. It provides functions such as instance management, database and table management, read/write splitting configuration, smooth scale-out, displaying monitored data, and IP address whitelist.

PolarDB-X Manager

PolarDB-X Manager is designed for global O&M personnel and DBAs. It manages the PolarDB-X resources and monitors the system. It provides the following main functions

- Manages all resources on which ApsaraDB RDS for MySQL instances depend, including virtual machines, SLB instances, and domain names.
- Monitors PolarDB-X instance statuses, including queries per second (QPS), active threads, connections, node network I/O, and node CPU utilization.

Rtools

Rtools is the O&M support system of PolarDB-X. It allows you to manage database configuration, read/write weights, connection parameters, topologies of databases and tables, and sharding rules.

10.4. Features

10.4.1. Horizontal partitioning (sharding)

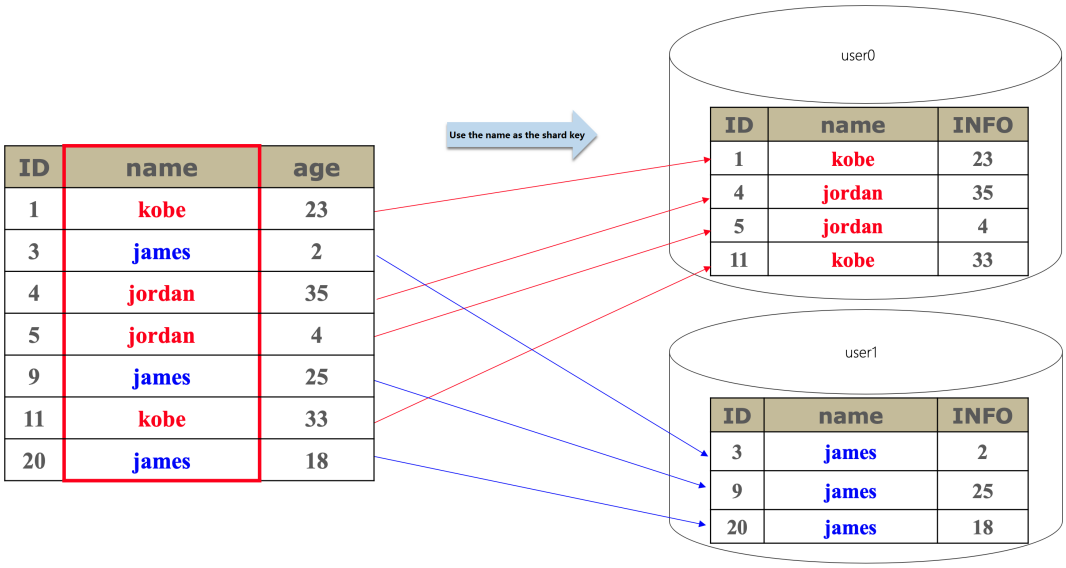
The core principle of PolarDB-X is horizontal partitioning of data, where data in a logical database is distributed and stored to multiple stable MySQL databases according to certain rules. These MySQL databases can be distributed across multiple instances or even across data centers, but provide external services (add, delete, modify, and query operations) as a single MySQL database. After partitioning, a physical database on an MySQL instance is called a database shard and a physical table is called a table shard (each table shard is a part of the complete data). By moving database shards on different MySQL instances, PolarDB-X implements database scale-out and improves the overall access to and the storage capacity of PolarDB-X databases.

PolarDB-X provides sharding rules, allowing you to select a partitioning policy that fits your business data characteristics. This ensures low latency for online database operations for transactions in high-concurrency scenarios. Therefore, when you use PolarDB-X, choosing the shard key is one of the important steps in database table structure design. The general principles are as follows:

- PolarDB-X performs well when writing data at the frontend. Most operations of such businesses are performed based on a specific database entity. For example, the business operations of the Internet are performed for users, the business operations of Internet of Things (IoT) are performed for devices and vehicles, the business operations of banks and government agencies are performed for customers, and the business operations of e-commerce independent software vendors (ISVs) and catering ISVs are performed for merchants. The data of such businesses can be partitioned by database entity. This, combined with global secondary indexes and eventually consistent transactions, can address the requirements on databases for large data volume, high concurrency, and low latency.
- For backend businesses, a batch of data is filtered and displayed on pages by condition and then processed and written back to the database. This is a business scenario in which PolarDB-X can partially address the needs. In this case, a large number of single-table associations and multi-table associations may exist, multiple filtering conditions are combined for DELETE and SELECT operations, and a large number of multi-table transactions are processed. Data partitioning by entity is recommended for such scenarios. If database processing is tightly related to time, data can be partitioned by time.

The following figure shows how data partitioning works.

Figure of data partitioning



ID	name	age
1	kobe	23
3	james	2
4	jordan	35
5	jordan	4
9	james	25
11	kobe	33
20	james	18

user0

ID	name	INFO
1	kobe	23
4	jordan	35
5	jordan	4
11	kobe	33

user1

ID	name	INFO
3	james	2
9	james	25
20	james	18

10.4.2. Smooth scale-out

To scale out a PolarDB-X instance, you can add ApsaraDB RDS for MySQL instances and migrate the original database shards to the new ApsaraDB RDS for MySQL instances.

Smooth scale-out is an online horizontal expansion method. It smoothly migrates the original database shards to the new ApsaraDB RDS for MySQL instances and increases the overall data storage capacity by adding ApsaraDB RDS for MySQL instances, which reduces the pressure on each RDS instance to process data.

How PolarDB-X scale-our works

Follow these steps:

1. Create a scale-out plan.

   Select a new ApsaraDB RDS for MySQL instance and database shards to be migrated. After the task is submitted, the system automatically creates a database and an account on the destination instance and submits a task for data migration and synchronization.

2. Perform full data migration.

   The system selects a time point before the current time and copies and migrates all data generated before this time point.

3. Perform incremental synchronization.

   After a full migration is completed, incremental data is synchronized according to the incremental change logs generated between a time point before the full migration and the current time, and eventually, the data is synchronized from the source database shard to the destination database shard in real time.

4. Verify data.

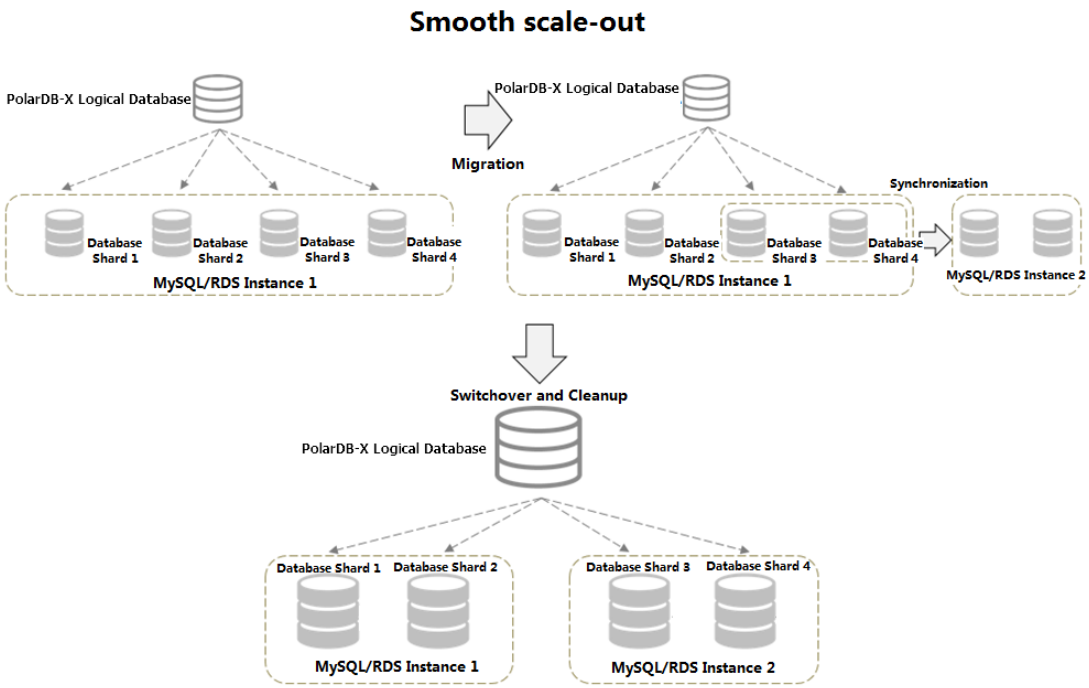
   When the incremental data is synchronized in quasi-real time, the system automatically performs full data verification and corrects inconsistent data caused by synchronization latency.

5. Disable the application service and switch routes.

   After verification, the incremental data is still synchronized in quasi-real time, and a specified time is selected for the switch. To ensure strict data consistency, we recommend that you disable the service (you can also not disable the service but the same data may be overwritten at a high concurrency). The engine layer switches routes based on database sharding rules to switch subsequent traffic to the new database. The switching process can be completed within seconds.

The following figure shows data migration between database shards.

Scale-out



To ensure data security and facilitate rollback of a scale-out task, data synchronization continues after the routing rule is switched. After the data O&M personnel confirm that the service is normal, you can clean up data in the source database shard in the console.

The whole scale-out process has little impact on services of the upper layer (some services may be affected if the instance type of the ApsaraDB RDS for MySQL instance is not satisfactory or its traffic pressure is high). If the service is not disabled during the switch, we recommend that you perform this operation when the database access traffic is low to reduce the possibility of concurrently updating the same data.

10.4.3. Read/write splitting

The read/write splitting function of PolarDB-X is a relatively transparent policy to switch over the read traffic for ApsaraDB RDS for MySQL instances.

You can add read-only ApsaraDB RDS for MySQL instances and adjust their read weights in the PolarDB-X console without code modification if your business applications can tolerate the latency of data synchronization between read-only instances and the primary instance. The read traffic is proportionally adjusted between the primary ApsaraDB RDS for MySQL instance and multiple read-only ApsaraDB RDS for MySQL instances. Write operations and transaction operations are performed on the primary ApsaraDB RDS for MySQL instance.

Note that a latency exists for data synchronization between the primary instance and read-only instances. When a large data definition language (DDL) statement is executed or a large volume of data is being corrected, the latency may be over one minute. Therefore, consider whether your business can tolerate the impact before using this function.

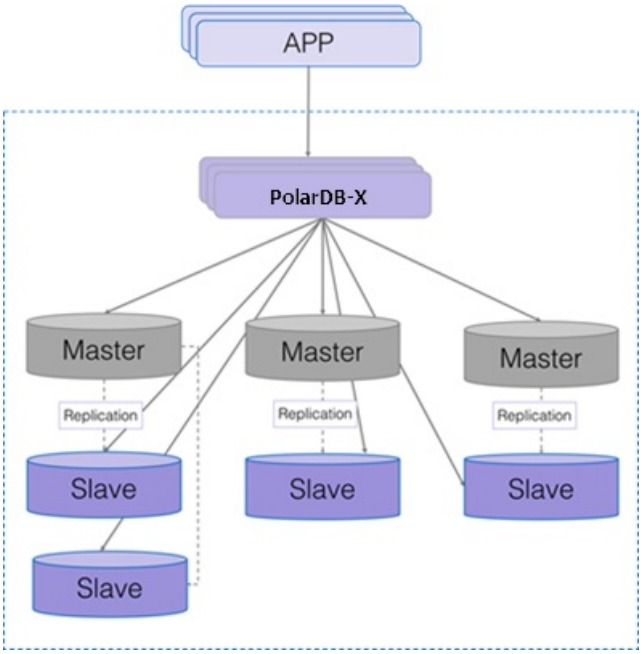
Adding read-only instances improves the read performance linearly. For example, if there is only one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added.

Traffic distribution and instance addition for read/write splitting

The read/write splitting function of PolarDB-X requires no modification of application code. You only need to add read-only instances and adjust the weights of read operations in the PolarDB-X console, to proportionally adjust the read traffic between the primary instance and multiple read-only instances. The write operations are performed on the primary instance.

Adding read-only instances improves the read performance linearly. For example, if there is one read-only instance, the read performance is doubled after one other read-only instance is added or tripled after two other read-only instances are added, as shown in the following figure.

Traffic distribution and expansion for read/write splitting



All data in the read operations on a read-only instance is asynchronously synchronized from the primary instance with a millisecond-level latency. For SQL statements that require high real-time performance, you can specify the primary instance through PolarDB-X Hint to execute these SQL statements, as shown in the following code:

```
/*TDDL:MASTER/select * from tddl5_users;
```

PolarDB-X allows you to run SHOW NODE to view the actual distribution of read traffic, as shown in the following figure.

SHOW NODE to view the actual distribution of read traffic

```
mysql> show node;
+------+------------------+-------------------+------------------+---------------------+---------------------+
| ID   | NAME             | MASTER_READ_COUNT | SLAVE_READ_COUNT | MASTER_READ_PERCENT | SLAVE_READ_PERCENT |
+------+------------------+-------------------+------------------+---------------------+---------------------+
| 0    | USERDATABASE_RDS |                10 |                2 |                 83% |                 17% |
+------+------------------+-------------------+------------------+---------------------+---------------------+
1 row in set (0.00 sec)
```

Read/write splitting in non-partition mode

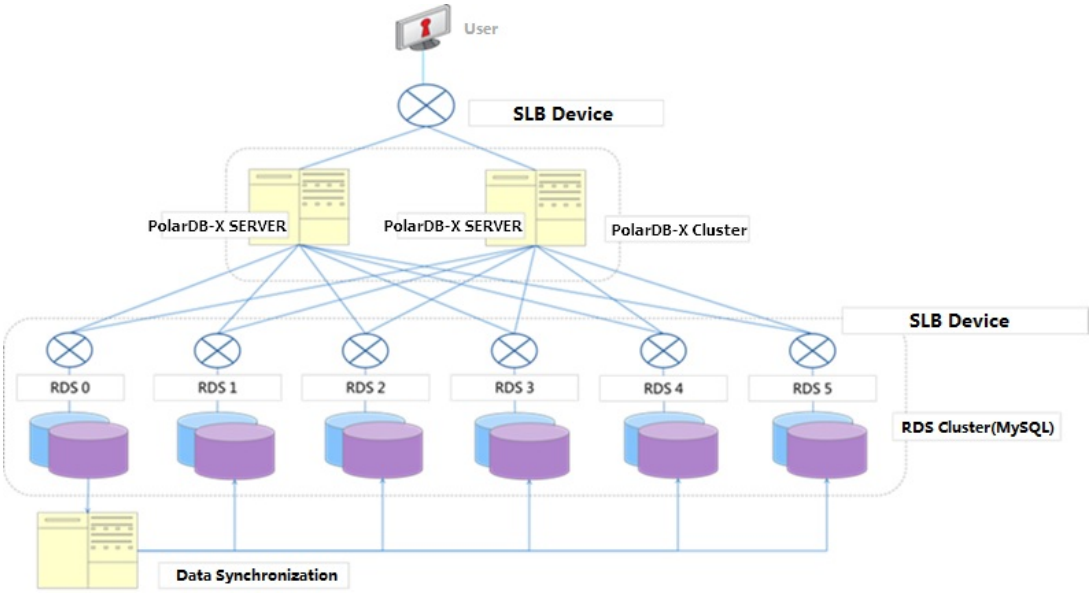
The read/write splitting function of PolarDB-X can be used independently in non-partition mode.

When you select an ApsaraDB RDS for MySQL instance for creating a PolarDB-X database in the PolarDB-X console, you can directly introduce a logical database on the ApsaraDB RDS for MySQL instance to the PolarDB-X database for read/write splitting without data migration.

10.4.4. Service upgrade and downgrade

A PolarDB-X instance consists of multiple server nodes that are deployed in a cluster and provides services externally through Server Load Balancer (SLB) and Domain Name System (DNS). The server nodes of PolarDB-X do not synchronize states, they process external requests in a balanced manner. When the processing capability of the server cluster is insufficient, server nodes can be added in real time to improve the service capability. If the resource utilization of all PolarDB-X server nodes in the cluster is low, you can remove some server nodes to downsize the cluster, and lower the capability of the service layer, for the elastic scaling of service capabilities. Figure of service upgrade and downgrade shows the details.

Figure of service upgrade and downgrade



10.4.5. Account and permission system

The account and permission system of PolarDB-X is used in the same way as MySQL, but does not support authorization across multiple databases, and has fewer permissions than MySQL. The system supports statements including GRANT, REVOKE, SHOW GRANTS, CREATE USER, DROP USER, and SET PASSWORD. It can be used to grant database-level and table-level permissions, but global and column-level permissions are not supported.

Account rules:

- The administrator account created in the console has all permissions.
- Only the administrator account can create and authorize accounts. Other accounts can only be created and authorized by the administrator account.
- The administrator account is bound to a database and does not have permissions on other databases. It can only access the bound database, and cannot grant permissions of other databases to an account. For example, the easydb administrator account can only connect to the easydb database, and can only grant permissions of the easydb database or tables in the easydb database to an account.

Currently, eight table-associated basic permissions are supported: CREATE, DROP, ALTER, INDEX, INSERT, DELETE, UPDATE, and SELECT. Among these operations:

- The TRUNCATE operation requires the table-level DROP permission.
- The REPLACE operation requires the table-level INSERT and DELETE permissions.

- The CREATE INDEX and DROP INDEX operations require the table-level INDEX permission.
- The CREATE SEQUENCE operation requires the database-level CREATE permission.
- The DROP SEQUENCE operation requires the database-level DROP permission.
- The ALTER SEQUENCE operation requires the database-level ALTER permission.
- The INSERT ON DUPLICATE UPDATE statement requires the table-level INSERT and UPDATE permissions.

10.4.6. PolarDB-X sequence

The PolarDB-X sequence of PolarDB-X (a 64-digit number of the signed BIGINT type in MySQL) aims to generate a globally unique number sequence (not necessarily in increments). This sequence is usually used to generate keys such as a primary key column and a unique key.

The PolarDB-X sequence of PolarDB-X can be implicitly used. When a table is partitioned to table shards, the primary key is auto_increment, and business data is inserted with no primary key specified, the globally unique primary key is automatically set, just like the single-instance MySQL database.

The PolarDB-X sequences can also be explicitly used. You can run `select xxx_seq.nextval from dual where count=?` to obtain one or more PolarDB-X sequences for other use in the application.

10.4.7. Second-level monitoring

PolarDB-X allows users to run **SHOW FULL STATS** to implement second-level monitoring. This command operates with the business monitoring system of PolarDB-X or third-party open-source monitoring software to provide better monitoring and alerting effect.

The following table describes the metrics supported by this command.

Metric	Description
QPS	The logical queries per second (QPS) specifying queries from an application to a PolarDB-X instance.
RDS_QPS	The QPS from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance is called physical QPS.
ERROR_PER_SECOND	The number of errors per second, which is the sum of SQL syntax errors, primary key conflicts, system errors, and connectivity errors.
VIOLATION_PER_SECOND	The number of primary key conflicts or unique key conflicts per second.
MERGE_QUERY_PER_SECOND	The number of queries merged from multiple table shards.
ACTIVE_CONNECTIONS	The number of connections in use.
CONNECTION_CREATE_PER_SECOND	The number of connections created per second.
RT(MS)	The logical response time (RT) for a response from an application to a PolarDB-X instance.

Metric	Description
RDS_RT(MS)	The physical RT for a response from a PolarDB-X instance to an ApsaraDB RDS for MySQL instance.
NET_IN(KB/S)	The network traffic received by PolarDB-X
NET_OUT(KB/S)	The network traffic sent by PolarDB-X
THREAD_RUNNING	The number of running threads.
HINT_USED_PER_SECOND	The number of queries with hints per second.
HINT_USED_COUNT	The total number of queries with hints since startup.
AGGREGATE_QUERY_PER_SECOND	The number of aggregate queries per second.
AGGREGATE_QUERY_COUNT	The total number of historical aggregate queries.
TEMP_TABLE_CREATE_PER_SECOND	The number of temporary tables created per second.
TEMP_TABLE_CREATE_COUNT	The total number of temporary tables created since startup.
MULTI_DB_JOIN_PER_SECOND	The number of cross-database JOIN queries per second.
MULTI_DB_JOIN_COUNT	The total number of cross-database JOIN queries since startup.

10.4.8. Distributed SQL engine

The distributed SQL engine of PolarDB-X is designed to achieve high compatibility with a single-instance MySQL database, to implement SQL push-down. PolarDB-X allows you to perform SQL operations, such as analysis, optimization, routing, and data aggregation.

The core principles of SQL push-down are as follows:

- Process data as close to the data as possible.
- Reduce data transfers over the network.
- Reduce data processing on PolarDB-X and offload the processing work to the lower-level data nodes whenever possible.
- Make full use of the features and capabilities of database storage.

10.4.9. High-availability architecture

Automatic traffic switchover of PolarDB-X Server

The PolarDB-X Server component of a PolarDB-X instance consists of multiple server nodes and provides services as a single connection through a load balancing service. When a PolarDB-X server fails, its traffic switches over to another PolarDB-X server in seconds. The entire failover process is transparent to users, with no need to change the application code or restart the application.

Automatic traffic switchover

PolarDB-X supports the read/write splitting function. You can sign in to the console, choose **DRDS Database > Read/Write Splitting**, and configure the function. This function allocates some read traffic to the secondary instances. PolarDB-X identifies the read SQL requests and distributes them to the primary and secondary ApsaraDB RDS for MySQL instances based on the configured ratio, to implement read/write splitting. A secondary instance allocated with only read traffic is called a read-only instance.

If multiple read-only instances are configured but one of them fails (the connection fails), PolarDB-X automatically withdraws the read traffic from the failed instance, and then re-allocates the traffic based on the ratio of read traffic in the remaining normal read-only instances.

The automatic traffic switchover process of read-only instances is transparent to users. You do not need to restart applications. When no read-only instance is available, read requests are still allocated proportionally to both the read-only and the primary instances, to prevent the primary instance from being overloaded. Errors are reported for the read requests allocated to read-only instances .

> **Note** All write requests and transactions are automatically routed to the primary instance for execution, regardless of the availability of read-only instances.

10.4.10. Software upgrade

- PolarDB-X automatically provides new versions of installed database software.
- Software upgrade is optional. It is carried out only upon your request.
- If PolarDB-X determines that your version has major security risks, it will notify you to schedule the upgrade. The PolarDB-X team will provide support during the entire upgrade process.
- The PolarDB-X upgrade process is generally completed within 5 minutes. During the upgrade process, there may be several transient database disconnections. There is minimal interruption to applications if the database reconnection (or connection pool) is properly configured for applications.

10.4.11. SQL compatibility

PolarDB-X is compatible with the MySQL protocols and supports most MySQL query syntax, common data manipulation language (DML) syntax, and data definition language (DDL) syntax. However, the pronounced architectural differences between distributed databases and single-instance databases restrict the usage of SQL. The compatibility and SQL restrictions are described as follows.

> **Note** Since there are many MySQL versions and the MySQL syntax and PolarDB-X versions are constantly updating, the compatibility discussed in this document is for reference only. Determine whether the selected MySQL version matches your business according to the actual test results.

PolarDB-X SQL restrictions

SQL restrictions are as follows:

- Custom data types and functions are not supported at present.
- Views, stored procedures, triggers, and cursors are not supported at present.
- Compound statements such as `BEGIN...END, LOOP...END LOOP, REPEAT...UNTIL...END REPEAT, and WHILE...DO...END WHILE` are not supported at present.

- Process control statements such as IF and WHILE are not supported at present.

Small syntax restrictions

DDL:

- CREATE TABLE tbl_name LIKE old_tbl_name does not support table sharding.
- CREATE TABLE tbl_name SELECT statement does not support table sharding.

DML:

- SELECT INTO OUTFILE, SELECT INTO DUMPFILE, and SELECTvar_name are not supported at present.
- INSERT DELAYED is not supported at present.
- Subqueries irrelevant to the WHERE condition are not supported at present.
- SQL subqueries that contain aggregation conditions are not supported at present.
- Variable references and operations in SQL statements are not supported at present, for example, `SET @c=1, @d=@c+1; SELECT @c, @d`.

Database management:

- SHOW WARNINGS does not support the LIMIT/COUNT combination.
- SHOW ERRORS does not support the LIMIT/COUNT combination.

Compatibility of PolarDB-X with SQL

Compatibility with MySQL protocols

PolarDB-X supports mainstream clients such as MySQL Workbench, Navicat For MySQL, and SQLyog.

> **Note** PolarDB-X supports the add, delete, modify, and query operations on databases. However, other special functions (such as import and diagnosis) have not been thoroughly tested.

PolarDB-X is compatible with the following DDL statements:

- `CREATE TABLE`
- `CREATE INDEX`
- `DROP TABLE`
- `DROP INDEX`
- `ALTER TABLE`
- `TRUNCATE TABLE`

PolarDB-X is compatible with the following DML statements:

- INSERT
- REPLACE
- UPDATE
- DELETE
- Subquery
- Scalar subquery
- Comparisons subquery
- Subquery with ANY, IN, or SOME
- Subquery with ALL

- Subquery by column
- Subquery with EXISTS or NOT EXISTS
- Subquery in the FROM clause
- SELECT

PolarDB-X is compatible with the following PREPARE statements:

- PREPARE
- EXECUTE
- DEALLOCATE PREPARE

PolarDB-X is compatible with the following database management statements

- SET
- SHOW
- KILL 'PROCESS_ID' (PolarDB-X only supports the KILL 'PROCESS_ID' command but does not support the KILL QUERY command.)
- SHOW COLUMNS
- SHOW CREATE TABLE
- SHOW INDEX
- SHOW TABLES
- SHOW TABLE STATUS
- SHOW TABLES
- SHOW VARIABLES
- SHOW WARNINGS
- SHOW ERRORS

> **Notice** Other SHOW commands are delivered to the database for processing by default, and the returned result data in different shards are not merged.

PolarDB-X is compatible with the following database tool statements:

- DESCRIBE
- EXPLAIN
- USE

Custom instructions of PolarDB-X are as follows:

- SHOW SEQUENCES, CREATE SEQUENCE, ALTER SEQUENCE, and DROP
- SEQUENCE. It manages PolarDB-X sequences.
- SHOW PARTITIONS FROM TABLE. It queries table shard keys.
- SHOW TOPOLOGY FROM TABLE. It queries the physical topology of a table.
- SHOW BROADCASTS. It queries all broadcast tables.
- SHOW RULE [FROM TABLE]. It queries the table sharding rule.
- SHOW DATASOURCES. It queries data sources of the backend database connection pool.
- SHOW DBLOCK/RELEASE DBLOCK. It defines the distributed LOCK.
- SHOW NODE. It queries the database read and write traffic.

- SHOW SLOW. It queries the slow SQL statements.
- SHOW PHYSICAL_SLOW. It queries slow SQL statements executed in the physical database.
- TRACE SQL_STATEMENT/SHOW TRACE. It traces the SQL statement execution process.
- EXPLAIN [DETAIL/EXECUTE] SQL_STATEMENT. It analyzes the SQL execution plans of PolarDB-X and physical databases.
- RELOAD USERS. It synchronizes the user information from PolarDB-X Console to the PolarDB-X Server.
- RELOAD SCHEMA. It clears data caches in the corresponding PolarDB-X database, such as cache of SQL parsing, syntax tree, and table structure.
- RELOAD DATASOURCES. It rebuilds a connection pool that connects the backend to all databases.

Database functions:

- SQL statements with shard keys are supported by all MySQL functions.
- SQL statements without a shard key are supported by only some functions.
- Operator functions

Function	Description
AND, &&	Logical AND
=	Assigns a value (a part of the SET statement or a part of the SET clause in the UPDATE statement).
BETWEEN... AND...	Determines a certain range of a value.
BINARY	Converts a string into a binary string.
&	Bitwise AND
~	Bitwise negation
^	Bitwise Exclusive OR (XOR)
DIV	Returns an integer obtained from integer division.
/	Division operator
<=>	NULL-safe equal operator
=	Equal operator, it compares the equality of two strings
>=	Operator, greater than or equal to
>	Greater than
IS NOTNULL	Tests for a non-NULL value.
ISNOT	Tests for a non-Boolean value.
ISNULL	It tests for a NULL value.
IS	Tests for a Boolean value.

Function	Description
<<	Bitwise left shift operator
<=	Operator, less than or equal to
<	Operator, less than
LIKE	Compares a character string to a specified string pattern.
-	Minus operator
%,	Returns the remainder of a number divided by another number.
NOT BETWEEN... AND...	Determines a certain range that a value is not in.
! =, <>	Operator, not equal to
NOT LIKE	Finds a specific character string that does not match a specified pattern.
NOT REGEXP	NOT operator in regular expressions
NOT, !	NOT
OR	Logical OR
+	Plus operator
REGEXP	Uses a regular expression for matching.
>>	Bitwise right shift operator
RLIKE	Uses a regular expression for matching. It is the same as REGEXP。
*	Multiplication operator
-	Takes the opposite value of the parameter.
XOR	Logical XOR
Coalesce	Returns the first non-NULL parameter.
GREATEST	Returns the largest parameter value.
LEAST	It returns the smallest parameter value.
STRCMP	Compares two strings.

- Process control functions

Function	Description
CASE	Case operator

Function	Description
IF()	If/else structure
IFNULL()	Null if/else structure
NULLIF()	If expr1 = expr2, NULL is returned.

- Numeric functions

Function	Description
ABS()	Returns the absolute value.
ACOS()	Returns the arc cosine of a number.
ASIN()	Returns the arc sine of a number.
ATAN2()	Returns the arc tangent of two parameters.
ATAN()	Returns the arc tangent of a parameter.
CEIL()	Obtains the smallest integer greater than or equal to a number.
CEILIG()	Obtains the smallest integer greater than or equal to a number.
CONV()	Converts a number between different number bases.
COS()	Returns the cosine of a number.
COT()	Returns the cotangent of a number.
CRC32()	Calculates the cyclic redundancy check (CRC) value.
DEGREES()	Converts a radian to a degree.
DIV	Returns an integer obtained from integer division.
EXP()	Returns e raised to the power of the specified number.
FLOOR()	Obtains the largest integer less than or equal to a number.
LN()	Returns the natural logarithm of a parameter.
LOG10()	Returns the logarithm with the base 10 of the parameter.
LOG2()	Returns the logarithm with the base 2 of the parameter.
LOG()	Returns the natural logarithm of the first parameter.
MOD()	Returns the remainder of a number.
%,MOD	Returns the remainder of a number divided by another number.

Function	Description
PI()	Returns the value of Pi.
POW()	Returns N power of the first parameter, where N is the second parameter.
POWER()	Returns N power of the first parameter, where N is the second parameter.
RADIANS()	Converts a parameter into a radian.
RAND()	Returns a random floating-point number.
ROUND()	Rounds up or down to an integer.
SIGN()	Returns the positive or negative sign of a parameter.
SIN()	Returns the sine value of a parameter.
SQRT()	Returns the square root of a parameter.
TAN()	Returns the tangent of a parameter value.
TRUNCATE(	Truncates to the specified decimal place.

- String functions

Function	Description
ASCII()	Returns the ASCII value of a character.
BIN()	Returns the binary value of a character.
BIT_LENGTH()	Returns the bit length of a string.
CHAR_LENGTH()	Returns the number of characters in a string.
CHAR()	Converts an input integer into a character.
CHARACTER_LENGTH()	Returns the number of characters in a string. It is the same as CHAR_LENGTH().
CONCAT_WS()	Connects the input parameters by using the specified separator.
CONCAT()	Returns a connection string.
ELT()	Returns the string at the index number.
EXPORT_SET()	-
FIELD()	Returns the index position of the first parameter in subsequent parameters.
FIND_IN_SET()	Returns the index position of the first parameter in the second parameter.

Function	Description
FORMAT()	Returns the formatted numbers of the specified decimal places.
HEX()	It converts a decimal number or string into a hexadecimal number.
INSERT()	Inserts a substring of a specified number of characters at the specified place.
INSTR()	Returns the index position where the substring appears for the first time.
LCASE()	Converts to lowercase letters. It is the same as LOWER().
LEFT()	Returns the characters of the specified number that is the furthest left.
LENGTH()	Returns the number of bytes of a string.
LIKE	Finds a specific character string matches a specified pattern.
LOCATE()	Returns the position where the substring appears for the first time.
LOWER()	Converts to lowercase letters.
LPAD()	Pads the left side of a string with a specific set of characters.
LTRIM()	Removes spaces at the beginning.
MAKE_SET()	Returns a set value (a string containing substrings separated by , characters) consisting of the characters specified in the first argument.
MID()	Extracts a substring from a string (starting at the specified position).
NOT LIKE	Finds a specific character string that does not match a specified pattern.
NOT REGEXP	Performs a pattern match of a string expression against a pattern.
OCT()	Converts a number to an octal number by a string.
OCTET_LENGTH()	Returns the number of bytes of a string. It is the same as LENGTH().
ORD()	Returns the code for the leftmost character of the given parameter.
POSITION()	Returns the position where the sub-string occurs for the first time. It is the same as LOCATE().
QUOTE()	Escapes parameters for use in SQL statements.
REPEAT()	Repeats a string for a specified number of times.
REPLACE()	Replaces the specified string in all the places where it appears.

Function	Description
REVERSE()	Reverses characters in a string.
RIGHT()	Returns the characters of the specified number that is the furthest right.
RPAD()	Pads strings for the specified number of times from the right.
RTRIM()	Removes spaces at the end.
SPACE()	Returns a string consisting of specified spaces.
STRCMP()	Compares two strings.
SUBSTR()	Returns the specified substring.
SUBSTRING_INDEX()	Returns the substring that appears for a specified number of times and in front of a separator in a string.
SUBSTRING()	Returns the specified substring.
TRIM()	Removes spaces at the beginning and end.
UCASE()	Converts a string to all uppercase. It is the same as UPPER().
UNHEX()	Returns a string that is the hexadecimal value of the parameter.
UPPER()	Converts a string to uppercase.

- Time functions

Function	Description
ADDDATE()	Adds a time value (an interval) to a date.
ADDTIME()	Adds a time interval to a time/datetime and then returns the time/datetime.
CURDATE()	Returns the current date.
CURRENT_DATE()	Returns the current date. It is the same as CURDATE().
CURRENT_TIME()	Returns the current time. It is the same as CURTIME().
CURRENT_TIMESTAMP()	Returns the current date and time. It is the same as NOW().
CURTIME()	Returns the current time.
DATE_ADD()	Adds a time value (an interval) to a date.
DATE_FORMAT()	Formats the date as required.
DATE_SUB()	Subtracts a specified time value (an interval) from a date.

Function	Description
DATE()	Extracts the date from the date expression or datetime expression.
DATEDIFF()	Subtracts one date from the other date.
DAY()	Returns the day (0-31) of a month for the specified date. It is the same as DAYOFMONTH().
DAYNAME()	Returns the weekday for a date.
DAYOFMONTH()	Returns the day (0-31) of a month for the specified date.
DAYOFWEEK()	Returns the day of a week (1 for Sunday and 7 for Saturday) for a date.
DAYOFYEAR()	Returns the day (1-366) of a year for a date.
EXTRACT()	Extracts a part of a date.
FROM_DAYS()	Converts a day to a date.
FROM_UNIXTIME()	Formats a UNIX timestamp as a date.
GET_FORMAT()	Returns a string of the date format.
HOUR()	Extracts hours from input time parameters.
LAST_DAY()	Returns the last day of the month for the parameter.
LOCALTIME()	Returns the current date and time. It is the same as NOW().
LOCALTIMESTAMP, LOCALTIMESTAMP()	Returns the current date and time. It is the same same as NOW().
MAKEDATE()	Returns the date, containing the year and the number of days.
MAKETIME()	Constructs a time containing the hour, minute, and second.
MICROSECOND()	Returns the microsecond of a parameter.
MINUTE()	Returns the minute of a parameter.
MONTH()	Returns the month of the input date.
MONTHNAME()	Returns the name of a month.
NOW()	Returns the current date and time.
PERIOD_ADD()	Adds a period to a date containing the year and month.
PERIOD_DIFF()	Returns the number of months between two periods.
QUARTER()	Returns the quarter of the date parameter.

Function	Description
SEC_TO_TIME()	Converts the second into the time in 'HH:MM:SS' format.
SECOND()	Returns the second (0-59) of a minute.
STR_TO_DATE()	Converts the string to a date.
SUBDATE()	Subtracts a specified time value (an interval) from a date when three parameters are called. It is the same as DATE_SUB().
SUBTIME()	Subtracts a time interval from a time/datetime and then returns the time/datetime.
SYSDATE()	Returns the function execution time.
TIME_FORMAT()	Formats the time.
TIME_TO_SEC()	Converts a parameter into a second.
TIME()	Extracts the time of an input parameter.
TIMEDIFF()	Returns the difference between two time/datetime expressions.
TIMESTAMP()	Returns a datetime value or expression based on a date or datetime value. If there are two arguments specified with this function, it first adds the second argument to the first, and then returns a datetime value.
TIMESTAMPADD()	Adds a time interval to the datetime expression.
TIMESTAMPDIFF()	Subtracts a time interval from the datetime expression.
UNIX_TIMESTAMP()	Returns the UNIX timestamp.
UTC_DATE()	Returns the current UTC date.
UTC_TIME()	Returns the current UTC time.
UTC_TIMESTAMP()	Returns the current UTC date and time.
WEEKDAY()	Returns the weekly index, where 1 indicates Sunday and 7 indicates Saturday.
WEEKOFYEAR()	Returns the number of the week on the calendar for a date.
YEAR()	It returns the year.

- Type conversion functions

Function	Description
BINARY	Converts a string into a binary string.

Function	Description
CAST()	Converts a value into a type.
CONVERT()	It converts a value into a type.

10.4.12. Table sharding

PolarDB-X provides convenient table sharding and changing functions, allowing you to flexibly partition a table into table shards, to glue table shards to a table, and to transfer data from one table shard to another.

10.4.13. Multi-zone instances

PolarDB-X allows you to select a multi-zone PolarDB-X instance. This ensures the PolarDB-X instance availability when one of the zones is unavailable.

10.4.14. Zone-disaster recovery

PolarDB-X provides the zone-disaster recovery function, supporting migration between single-zone instances and dual-zone instances. Zone-based disaster recovery can be performed if an inappropriate zone is selected for the target PolarDB-X instance or the available ApsaraDB RDS for MySQL instances in the target PolarDB-X zone are insufficient.

11.AnalyticDB for MySQL

11.1. What is AnalyticDB for MySQL?

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service that is developed by Alibaba Cloud to analyze large amounts of data at high concurrency. AnalyticDB for MySQL can analyze hundreds of billions of data records across multiple dimensions within milliseconds and provide you with data-driven insights into your business.

Background information

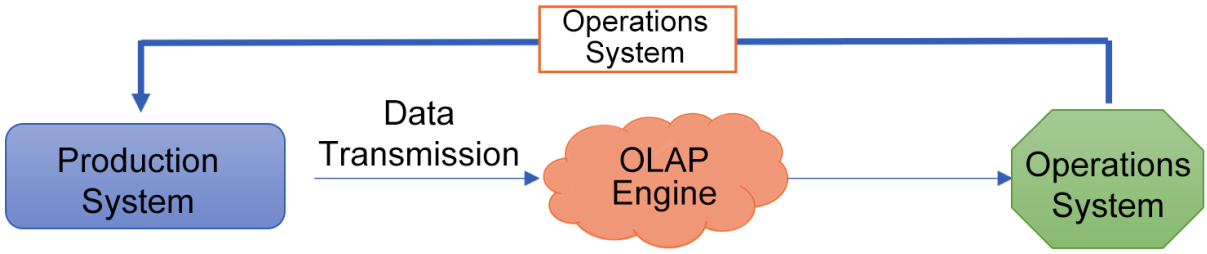
To keep up with the rapid development of technology and the demands of processing ever-increasing amounts of data, big data analytics services are changing to make data processing independent of business systems.

Database management systems ideal for OLTP, such as MySQL and PostgreSQL, are typically used in business systems. OLTP is ideal for processing transactions and allows you to frequently insert and modify data. However, once your data grows to billions of rows or you require complex computational functions, OLTP becomes less ideal. In these cases, OLAP is a better solution for processing such data.

Current developments

AnalyticDB for MySQL uses SQL to build relational data warehouses. You can manage databases, scale nodes in or out, and scale specifications up or down. AnalyticDB for MySQL provides various visualization and ETL tools to simplify enterprise data processing.

You can use AnalyticDB for MySQL to refine business operations, gain real-time insights into data values, and promote continuous digital transformation for enterprises. A growing number of industries, such as logistics, transport, and new retail, use OLAP to refine their business operations and optimize production guidelines, operation efficiencies, and enterprise decisions.



AnalyticDB for MySQL supports a large number of concurrent queries and ensures high system availability through dynamic multi-copy storage and computing technology. Therefore, AnalyticDB for MySQL can serve as a backend system for various products, including user-facing and enterprise-facing products. AnalyticDB for MySQL is used in Internet business systems that have hundreds of thousands to tens of millions of users, such as Data Cube, Taobao Index, Kuaidi Dache, Alimama DMP, and Taobao Groceries.

11.2. Benefits

AnalyticDB for MySQL is a real-time online analytical processing (RT-OLAP) service that is developed by Alibaba Cloud to analyze large amounts of data at high concurrency. AnalyticDB for MySQL can analyze hundreds of billions of data records across multiple dimensions within milliseconds and provide you with data-driven insights into your business.

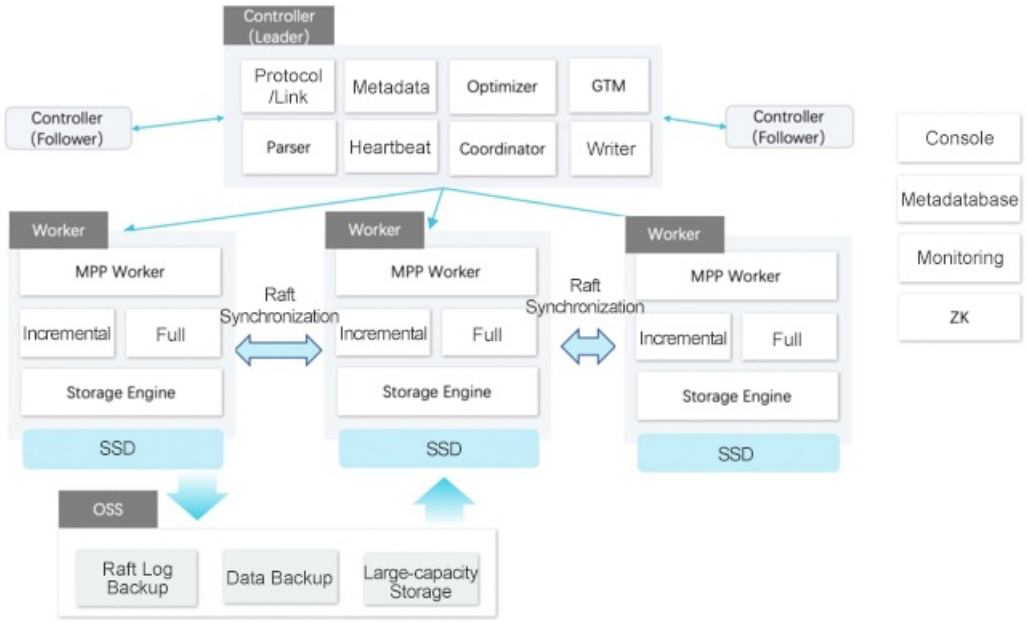
AnalyticDB for MySQL uses the Full MPP Mode (MPP) technology that features hybrid row-column storage to solve the technical limits of OLTP and traditional data warehouses. AnalyticDB for MySQL builds high-performance and cost-effective data warehouses to process petabytes of data.

AnalyticDB for MySQL is fully compatible with MySQL protocols and the SQL:2003 standard. You can migrate your business to AnalyticDB for MySQL with few to no changes to your existing business.

11.3. Architecture

AnalyticDB for MySQL is a distributed real-time computing system based on the MPP architecture. AnalyticDB for MySQL is built on top of the Apsara system and incorporates distributed retrieval technology.

AnalyticDB for MySQL consists of the underlying infrastructure, compute nodes, controllers, and storage nodes.



Underlying infrastructure

The underlying infrastructure consists of the following parts:

- Apsara system: used for logical isolation, persistence data storage, and to construct schemas and indexes.
- Metadatabase: refers to ApsaraDB for RDS or Tablestore that is used to store metadata of AnalyticDB for MySQL.

  > **Note** Metadata is not involved in actual computations.

- Apache ZooKeeper module: performs distributed coordination among components.

Controllers

A controller is used to control the allocation of database resources in compute nodes and the distribution of computing resources. The controller can also manage compute nodes and tasks that run in the database background. A controller consists of multiple modules:

- Alibaba Cloud Server Load Balancer (SLB): groups controllers and implements load balancing among controllers.
- Client access manager.
- SQL parser.
- AnalyticDB for MySQL console.

AnalyticDB for MySQL supports the following clients, drivers, programming languages, and middleware:

- Clients that support MySQL 5.1, 5.5, or 5.6 protocols and drivers: MySQL 5.1.x Connector/J, MySQL 5.3.x Connector/ODBC, and MySQL 5.1.x, 5.5.x, or 5.6.x client.
- Programming languages: JAVA, Python, C/C++, Node.js, PHP, and R (RMySQL).
- Middleware: Websphere Application Server 8.5, Apache Tomcat, and JBoss.

Compute nodes

Compute nodes run computing tasks that are issued by controllers to read, filter, merge, and compute data.

Storage nodes

Storage nodes write data, save data to disk storage, and replicate data between nodes. Storage nodes support data backup and restoration.

11.4. System features

Compatibility with MySQL

- Supports MySQL and standard JDBC and ODBC interfaces.
- Supports several MySQL development tools such as the Data Management (DMS) console, MySQL command line client, DBeaver, Navicat, and SQL Workbench/J.
- Uses relational models to store data and provides SQL statements to flexibly compute and analyze data. You do not need to create a data model in advance.
- Supports major data types to display numbers, characters, dates, and binary data.

INSERT, UPDATE, and DELETE operations

AnalyticDB for MySQL supports concurrent INSERT and DELETE operations on one or more tables.

- Allows you to perform INSERT and DELETE operations on the real-time tables that have the primary key defined.
- Provides multiple mechanisms to ensure that the written data is not lost. Both REPLACE INTO/INSERT OVERWRITE and INSERT IGNORE INTO statements are supported.
- Supports `INSERT INTO...SELECT FROM` statements.

AnalyticDB for MySQL supports SELECT operations on one or more tables.

- Supports multiple column mapping methods such as expressions, functions, aliases, column names, and CASE WHEN. The supported SELECT operations are 90% compatible with standard MySQL queries.
- Supports clauses such as FROM table name AS alias and JOIN table name AS alias.

- Supports JOIN operations between real-time tables, between real-time tables and dimension tables, and between ON conditions.
- Supports subqueries with up to three levels, JOIN operations between subqueries under specific conditions, and queries with IN operators for dimension table data.

  > **Note** MPP engines allow you to query with IN operators for data of dimension and real-time tables.

- Supports WHERE clauses that are combined with AND and OR operators, function expressions, or BETWEEN and IS operators.
- Supports GROUP BY operations for multiple columns and aliases that are generated from column mapping expressions such as CASE WHEN.
- Supports ORDER BY operations for expressions and columns in ascending or descending order.
- Supports common aggregate functions and HAVING operations.
- Supports COUNT(DISTINCT) operations for multiple columns that contain hash partition columns, and for any columns if the Full MPP Mode is used.
- Supports UNION, UNION ALL, MINUS, and INTERSECT operators for multiple SELECT statements.

Major data types and diversified OLAP functions

AnalyticDB for MySQL supports major data types and a wide range of OLAP functions to display numbers, characters, dates, and binary data.

Hybrid row-column storage

AnalyticDB for MySQL supports hybrid row-column storage within a table to handle the following hybrid load scenarios:

- Detail query in OLTP: requires you to read and write the detailed data of an entire row by using a single SELECT statement. AnalyticDB for MySQL can return query results at low I/O costs.
- Large-scale multidimensional analysis in OLAP: includes statistical analysis and JOIN operations for large amounts of data, typically for several columns in a wide table. AnalyticDB for MySQL has load processing capabilities in OLAP scenarios.
- High throughput: AnalyticDB for MySQL allows you to write hundreds of billions of data entries in real time each day.

High data compression rate

AnalyticDB for MySQL provides adaptive compression algorithms. AnalyticDB for MySQL can automatically select an optimal algorithm that delivers a compression ratio of up to 1:20 and provide DML operations in data compression status based on different data distribution methods and data types.

Data import

AnalyticDB for MySQL provides multiple methods to import data.

- Allows you to batch import files in parallel by using multiple nodes.
- Allows you to import data in the format of CSV files, TEXT files, or files with multiple delimiters.
- Allows you to import data by using multiple nodes in real-time streaming mode.

Data export

AnalyticDB for MySQL allows you to specify a destination and batch export data in parallel by using multiple nodes.

- Provides SELECT statements to export query results.
- Provides DUMP DATA statements to export large amounts of data to Object Storage Service (OSS) and MaxCompute.

Load balancing management

AnalyticDB for MySQL integrates with Alibaba Cloud Server Load Balancer (SLB) to implement load balancing. When a server fails, SLB automatically redirects client requests to an available server. Even if two or more servers fail, AnalyticDB for MySQL can still provide services.

11.5. Unique features

11.5.1. Full-text indexing

This topic describes the features of full-text indexing in AnalyticDB for MySQL.

Full-text indexing has the following features:

- AnalyticDB for MySQL supports the SQL-92 standard and MySQL protocols. AnalyticDB for MySQL provides full-text retrieval based on SQL statements, which reduces learning costs. AnalyticDB for MySQL also unifies common structured data analysis with flexible unstructured data analysis and uses the same SQL language to manage multiple types of data, which reduces development costs.
- AnalyticDB for MySQL can perform integrated retrieval and multi-modal analysis for structured and unstructured data. Most solutions in the industry focus only on creating full-text indexes on text data to retrieve unstructured data, and not structured data. AnalyticDB for MySQL supports full-text retrieval and provides a variety of classic index structures in traditional databases, such as B-tree index, bitmap index, and inverted index. You can use multiple indexes within a table to meet a variety of retrieval requirements.
- AnalyticDB for MySQL provides comprehensive distributed computing capabilities. Services such as Elasticsearch and Solr cannot provide distributed JOIN solutions. AnalyticDB for MySQL uses the MPP and DAG architecture to provide distributed JOIN, GROUP BY, and aggregation capabilities such as COUNT(DISTINCT). AnalyticDB for MySQL also provides computation based on partition and non-partition keys.

11.5.2. Data consistency

AnalyticDB for MySQL provides strong data consistency. Data that is written or updated can take effect immediately.

12.AnalyticDB for PostgreSQL

12.1. What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL is a distributed analytic database service that leverages the massively parallel processing (MPP) architecture, where each instance is composed of multiple compute nodes. AnalyticDB for PostgreSQL provides MPP warehousing services that support horizontal scaling of storage and compute capabilities, online analysis of petabytes of data, and offline processing of Extract, Transform, and Load (ETL) tasks.

AnalyticDB for PostgreSQL is developed based on the PostgreSQL kernel and has the following features:

- Supports the SQL:2003 standard, OLAP aggregate functions, views, Procedural Language for SQL (PL/SQL), user-defined functions (UDFs), and triggers. AnalyticDB for PostgreSQL is partially compatible with the Oracle syntax.
- Supports horizontal scaling of storage and compute capabilities based on the MPP architecture. AnalyticDB for PostgreSQL also supports range and list partitioning.
- Supports row store, column store, and multiple indexes. AnalyticDB for PostgreSQL also supports multiple compression methods based on column store to reduce storage costs.
- Supports standard database isolation levels and distributed transactions to ensure data consistency.
- Provides the vector computing engine and the CASCADE-based SQL query optimizer to ensure high-performance SQL analysis.
- Uses a primary/secondary architecture to ensure dual-copy data storage and service availability.
- Provides online scaling, system monitoring, and disaster recovery to reduce O&M costs.

12.1.1. Scenarios

This topic describes the OLAP data analysis services that AnalyticDB for PostgreSQL supports.

- Extract, Transform, and Load (ETL) for offline data processing

  AnalyticDB for PostgreSQL provides the following benefits that make it ideal to optimize complex SQL queries as well as aggregate and analyze large amounts of data:

  - Supports standard SQL syntax, OLAP window functions, and stored procedures.
  - Provides the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.
  - Built on the MPP architecture that supports horizontal scaling of storage and compute capabilities to analyze and process petabytes of data.
  - Provides column store-based high-performance aggregation of large tables at a high compression ratio to maximize storage capacity.

- Online high-performance query

  AnalyticDB for PostgreSQL provides the following benefits for real-time exploration, warehousing, and updating of data:

  - Allows you to write and update high-throughput data by performing INSERT, UPDATE, and DELETE operations.

- Allows you to query data based on row store and multiple indexes to obtain results within milliseconds. These indexes include B-tree indexes, bitmap indexes, and hash indexes.
- Supports distributed transactions, standard database isolation levels, and HTAP.

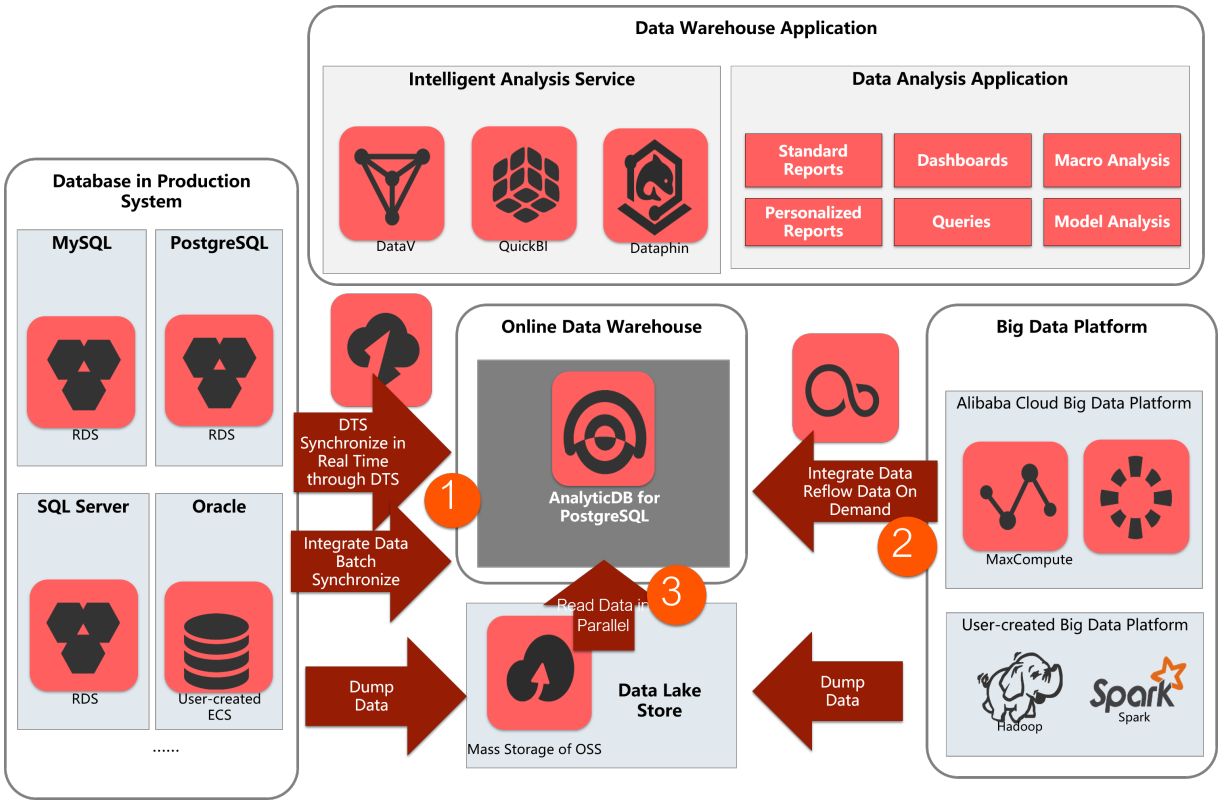
- Multi-model data analysis

  AnalyticDB for PostgreSQL provides the following benefits for processing unstructured data from a variety of sources:

  - Supports the PostGIS extension for geographic data analysis and processing.
  - Uses the MADlib library of in-database machine learning algorithms to implement AI-native databases.
  - Provides high-performance retrieval and analysis of unstructured data such as images, speech, and text by means of vector retrieval.
  - Supports formats such as JSON and can process and analyze semi-structured data such as logs.

Typical scenarios

AnalyticDB for PostgreSQL is applicable to the following scenarios:



- Data warehousing service

  Data Transmission Service (DTS) can synchronize data in real time in production system databases such as ApsaraDB RDS for MySQL, ApsaraDB RDS for PostgreSQL, and PolarDB as well as traditional databases such as Oracle and SQL Server. Data can also be batch synchronized to AnalyticDB for PostgreSQL by using Data Integration. AnalyticDB for PostgreSQL supports ETL operations on large amounts of data. You can also use DataWorks to schedule these tasks. AnalyticDB for PostgreSQL also provides high-performance online analysis capabilities and can use Quick BI, DataV, Tableau, and FineReport for report presentation and real-time query.

- Big data analytics platform

You can use Data Integration or OSS to import large amounts of data from MaxCompute, Hadoop, and Spark to AnalyticDB for PostgreSQL for high-performance analysis, processing, and exploration.

- Data lake analytics

  AnalyticDB for PostgreSQL can use foreign tables to access the large amounts of data stored in OSS in parallel and build an Alibaba Cloud data lake analytics platform.

12.2. Benefits

This topic describes the benefits of AnalyticDB for PostgreSQL.

- Real-time analysis

  Built on the MPP architecture that supports horizontal scaling and can respond to queries on petabytes of data within seconds. AnalyticDB for PostgreSQL supports the leading vector computing feature and intelligent indexes of column store. It also supports the CASCADE-based SQL query optimizer to enable complex queries without the need for tuning.

- Stability and reliability

  Provides ACID properties for distributed transactions. Transactions are consistent across nodes and all data is synchronized between primary and secondary nodes. AnalyticDB for PostgreSQL supports distributed deployment and provides transparent monitoring, switching, and restoration to secure your data infrastructure.

- Easy to use

  Supports rich SQL syntax and functions, Oracle functions, stored procedures, user-defined functions (UDFs), and isolation levels of transactions and databases. You can use popular business intelligence (BI) software and ETL tools online.

- Ultra-high performance

  Supports row store, column store, and multiple indexes. The vector engine provides high-performance analysis and computing capabilities. The CASCADE-based SQL query optimizer enables complex queries without the need for tuning. AnalyticDB for PostgreSQL supports high-performance parallel import of data from OSS.

- Flexible scalability

  Enables you to scale out compute nodes as well as CPU, memory, and storage resources on demand to improve OLAP performance.

  Supports transparent OSS operations. OSS offers a larger storage capacity for cold data that does not require online analysis.

  Supports online scaling to add, remove, modify, and query data during data redistribution.

- Resource isolation

  Supports multi-tenant parallel execution on a cluster by using multiple instances. Tasks from tenants are submitted to queues on different instances for execution. Resources of each AnalyticDB for PostgreSQL instance are isolated among tenants.

- Permission management

  Allows you to configure and manage tenants in a dynamic and centralized manner. You can also isolate resources and query usage statistics of resources. Management of multi-level tenants is supported.

- Resource scheduling

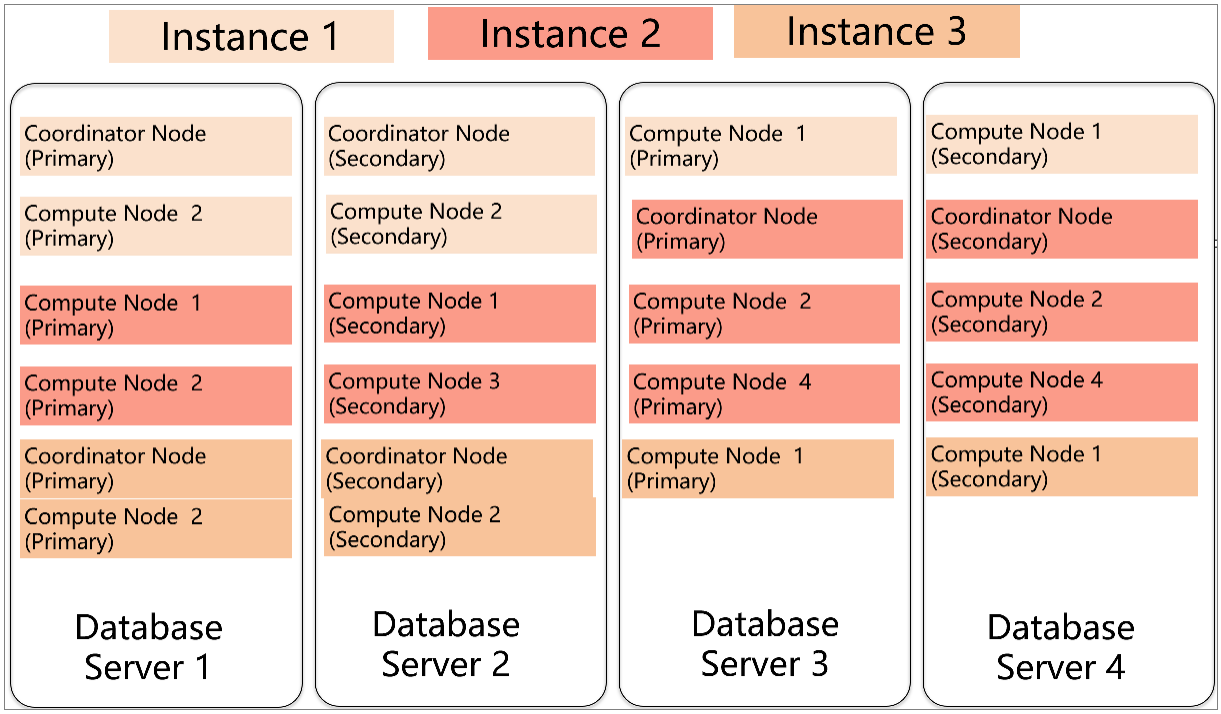
  Supports multi-tenant scheduling of multiple clusters and resource pools.

12.3. Architecture

This topic describes the physical architecture and logical architecture of an AnalyticDB for PostgreSQL cluster.

Physical architecture of a cluster

The following figure shows the physical architecture of an AnalyticDB for PostgreSQL cluster.

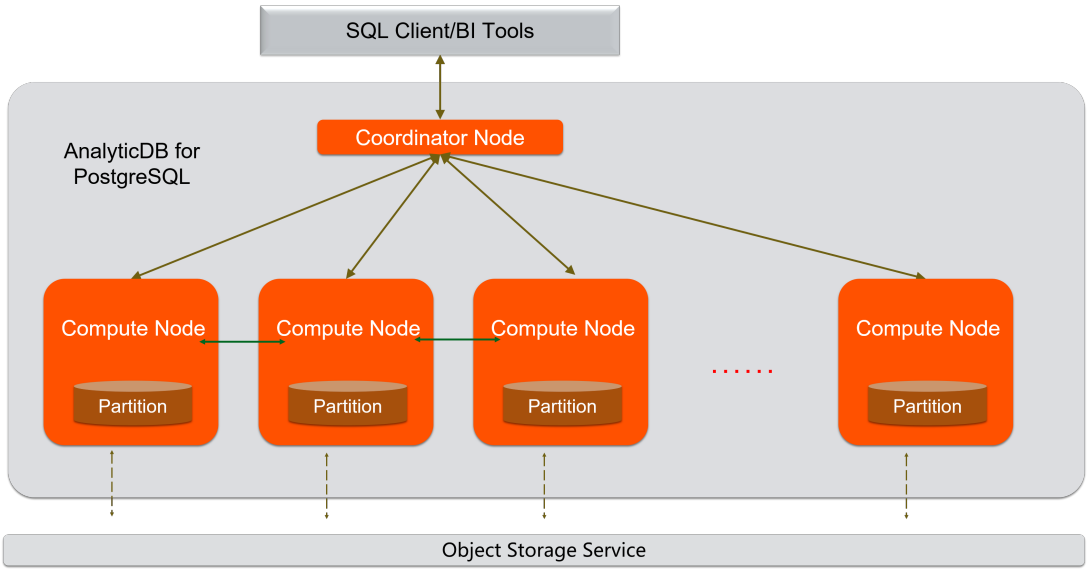


You can create multiple AnalyticDB for PostgreSQL instances in a physical cluster of AnalyticDB for PostgreSQL by using the management and control system. Each instance consists of a coordinator node and multiple compute nodes.

- The coordinator node is used for access from applications. It receives connection requests and SQL query requests from clients and dispatches computing tasks to compute nodes. The cluster deploys a secondary node of the coordinator node on an independent physical server and replicates data from the primary node to the secondary node for failover. The secondary node does not accept external connections.
- Compute nodes are independent instances in AnalyticDB for PostgreSQL. Data is evenly distributed across compute nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Each compute node uses a primary/secondary architecture for automatic failover.

Logical architecture of an instance

You can create multiple instances within an AnalyticDB for PostgreSQL cluster. The following figure shows the logical architecture of an instance.AnalyticDB for PostgreSQL



Data is distributed across compute nodes by hash value or RANDOM function of a specified distribution column. Each compute node uses a primary/secondary architecture to ensure dual-copy storage. High-performance network communication is supported across nodes. When the coordinator node receives a request from an application, the coordinator node parses and optimizes SQL statements to generate a distributed execution plan. After the coordinator node sends the execution plan to the compute nodes, the compute nodes perform massively parallel processing of the plan.

12.4. Features

This topic describes the features of AnalyticDB for PostgreSQL.

Distributed architecture

AnalyticDB for PostgreSQL is built on the massively parallel processing (MPP) architecture. Data is distributed evenly across nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Storage and compute capabilities are scaled out by adding nodes to ensure a quick response as the data volume increases.

AnalyticDB for PostgreSQL supports distributed transactions to ensure data consistency among nodes. It supports three transaction isolation levels: SERIALIZABLE, READ COMMITTED, and READ UNCOMMITTED.

High-performance data analysis

AnalyticDB for PostgreSQL supports column store and row store for tables. Row store provides high update performance. Column store provides high OLAP aggregate analysis performance for tables. AnalyticDB for PostgreSQL supports B-tree indexes, bitmap indexes, and hash indexes to enable high-performance analysis, filtering, and query.

AnalyticDB for PostgreSQL uses the CASCADE-based SQL query optimizer. AnalyticDB for PostgreSQL combines the cost-based optimizer (CBO) with the rule-based optimizer (RBO) to provide SQL optimization features such as automatic subquery decorrelation. These features enable complex queries without the need for tuning.

High-availability service

AnalyticDB for PostgreSQL builds a system for automatic monitoring, diagnosis, and troubleshooting based on the Apsara system. This helps reduce O&M costs.

The coordinator node compiles and optimizes SQL statements by storing database metadata and receiving query requests from clients. The coordinator node uses a primary/secondary architecture to ensure strong consistency of metadata. If the primary coordinator node fails, the service is automatically switched to the secondary coordinator node.

Data synchronization methods and tools

You can use Data Transmission Service (DTS) or DataWorks Data Integration to synchronize data from MySQL or PostgreSQL databases to AnalyticDB for PostgreSQL. You can use popular Extract, Transform, and Load (ETL) tools to import ETL data to and schedule jobs in AnalyticDB for PostgreSQL databases. You can also use standard SQL syntax to query data from formatted files stored in OSS by using foreign tables in real time.AnalyticDB for PostgreSQL

AnalyticDB for PostgreSQL supports popular business intelligence (BI) reporting tools such as Quick BI, DataV, Tableau, and FineReport. It also supports ETL tools, including Informatica and Kettle.

Data security

AnalyticDB for PostgreSQL supports the configuration of whitelists. You can add up to 1,000 IP addresses of servers to a whitelist to allow access to your instance and control risks from access sources. AnalyticDB for PostgreSQL also supports Anti-DDoS to monitor inbound traffic in real time. When large amounts of malicious traffic is identified, the traffic is scrubbed by means of IP filtering. If traffic scrubbing is insufficient, blackhole filtering is triggered.

Supported SQL features

- Supports row store and column store.
- Supports multiple indexes, including B-tree indexes, bitmap indexes, and hash indexes.
- Supports distributed transactions and standard isolation levels to ensure data consistency among nodes.
- Supports character, date, and arithmetic functions.
- Supports stored procedures, user-defined functions (UDFs), and triggers.
- Supports views.
- Supports range partitioning, list partitioning, and the definition of multi-level partitions.
- Supports multiple data types. The following table provides a list of data types and their information.

Data type	Alias	Storage size	Range	Description
bigint	int8	8 bytes	-9223372036854775808 to 9223372036854775807	An integer within a large range.
bigserial	serial8	8 bytes	1 to 9223372036854775807	A large auto-increment integer.
bit [(n)]	None	n bits	A bit string constant	A bit string with a fixed length.
bit varying [(n)]	varbit	A bit string with a variable length.	A bit string constant	A bit string with a variable length.

Data type	Alias	Storage size	Range	Description
boolean	bool	1 byte	true/false, t/f, yes/no, y/n, 1/0	A boolean value (true or false).
box	None	32 bytes	((x1,y1),(x2,y2))	A rectangular box on a plane, which is not allowed in a column that is used as the distribution key.
bytea	None	1 byte + binary string	Sequence of octets	A binary string with a variable length.
character [(n)]	char [(n)]	1 byte + n	A string up to n characters in length	A blank-padded string with a fixed length.
character varying [(n)]	varchar [(n)]	1 byte + string size	A string up to n characters in length	A string with a limited variable length.
cidr	None	12 or 24 bytes	None	IPv4 and IPv6 networks.
circle	None	24 bytes	<(x,y),r> (center and radius)	A circle on a plane, which is not allowed in distribution key columns.
date	None	4 bytes	4713 BC - 294,277 AD	Calendar date (year, month, day).
decimal [(p, s)]	numeric [(p, s)]	variable	No limits	User-specified precision, which is exact.
double precision	float8	8 bytes	Precise to 15 decimal digits	Variable precision, which is inexact.
	float			
inet	None	12 or 24 bytes	None	IPv4 and IPv6 hosts and networks.
integer	int, int4	4 bytes	-2.1E+09 to +2147483647	An integer in typical cases.
interval [(p)]	None	12 bytes	-178000000 years - 178000000 years	A time range.
json	None	1 byte + json size	JSON string	A string with an unlimited variable length.
lseg	None	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane, which is not allowed in distribution key columns.
macaddr	None	6 bytes	None	A Media Access Control (MAC) address.

Data type	Alias	Storage size	Range	Description
money	None	8 bytes	-92233720368547758.08 to +92233720368547758.07	Currency amount.
path	None	16+16n bytes	[(x1,y1),...]	A geometric path on a plane, which is not allowed in distribution key columns.
point	None	16 bytes	(x,y)	A geometric point on a plane, which is not allowed in distribution key columns.
polygon	None	40+16n bytes	((x1,y1),...)	A closed geometric path on a plane, which is not allowed in a column that is used as the distribution key.
real	float4	4 bytes	Precise to 6 decimal digits	Variable precision, which is inexact.
serial	serial4	4 bytes	1 to 2147483647	An auto-increment integer.
smallint	int2	2 bytes	-32768 to 32767	An integer within a small range.
text	None	1 byte + string size	A string with a variable length	A string with an unlimited variable length.
time [(p)] [without time zone]	None	8 bytes	00:00:00[.000000] - 24:00:00[.000000]	The time of a day without the time zone.
time [(p)] with time zone	timetz	12 bytes	00:00:00+1359 - 24:00:00-1359	The time of a day with the time zone.
timestamp [(p)] [without time zone]	None	8 bytes	4713 BC - 294,277 AD	The date and time without the time zone.
timestamp [(p)] with time zone	timestamptz	8 bytes	4713 BC - 294,277 AD	The date and time with the time zone.

Data type	Alias	Storage size	Range	Description
xml	None	1 byte + xml size	Variable-length XML string	A string with an unlimited variable length.

13.KVStore for Redis

13.1. What is KVStore for Redis?

KVStore for Redis is an online key-value storage service compatible with open-source Redis protocols. KVStore for Redis supports various types of data, such as strings, lists, sets, sorted sets, and hash tables. The service also supports advanced features, such as transactions, message subscription, and message publishing. Based on the hybrid storage of memory and hard disks, KVStore for Redis can provide high-speed data read/write capability and support data persistence.

As a cloud computing service, KVStore for Redis works with hardware and data deployed in the cloud, and provides comprehensive infrastructure planning, network security protections, and system maintenance services.

13.1.1. Scenarios

Game industry applications

KVStore for Redis can be an important part of the business architecture for deploying a game application.

Scenario 1: KVStore for Redis works as a storage database

The architecture for deploying a game application is simple. You can deploy a main program on an ECS instance and all business data on a KVStore for Redis instance. The KVStore for Redis instance works as a persistent storage database. KVStore for Redis supports data persistence, and stores redundant data on primary and secondary nodes.

Scenario 2: KVStore for Redis works as a cache to accelerate connections to applications

KVStore for Redis can work as a cache to accelerate connections to applications. You can store data in a Relational Database Service (RDS) database that works as a backend database.

Reliability of the KVStore for Redis service is vital to your business. If the KVStore for Redis service is unavailable, the backend database is overloaded when processing connections to your application. KVStore for Redis provides a two-node hot standby architecture to ensure high availability and reliability of services. The primary node provides services for your business. If this node fails, the system automatically switches services to the secondary node. The complete failover process is transparent.

Live video applications

In live video services, KVStore for Redis works as an important measure to store user data and relationship information.

Two-node hot standby ensures high availability

KVStore for Redis uses the two-node hot standby method to maximize service availability.

Cluster editions eliminate the performance bottleneck

KVStore for Redis provides cluster instances to eliminate the performance bottleneck that is caused by Redis single-thread mechanism. Cluster instances can effectively handle traffic bursts during live video streaming and support high-performance requirements.

Easy scaling relieves pressure at peak hours

KVStore for Redis allows you to easily perform scaling. The complete upgrade process is transparent. Therefore, you can easily handle traffic bursts at peak hours.

E-commerce industry applications

In the e-commerce industry, the KVStore for Redis service is widely used in the modules such as commodity display and shopping recommendation.

Scenario 1: rapid online sales promotion systems

During a large-scale rapid online sales promotion, a shopping system is overwhelmed by traffic. A common database cannot properly handle so many read operations.

However, KVStore for Redis supports data persistence, and can work as a database system.

Scenario 2: counter-based inventory management systems

In this scenario, you can store inventory data in an RDS database and save count data to corresponding fields in the database. In this way, the KVStore for Redis instance reads count data, and the RDS database stores count data. KVStore for Redis is deployed on a physical server. Based on solid-state drive (SSD) high-performance storage, the system can provide a high-level data storage capacity.

13.2. Benefits

High performance

- Supports cluster features and provides cluster instances of 128 GB or higher to meet large capacity and high performance requirements.
- Provides primary/secondary instances of 32 GB or smaller to meet general capacity and performance requirements.

Elastic scaling

- Easy scaling of storage capacity: you can scale instance storage capacity in the KVStore for Redis console based on business requirements.
- Online scaling without interrupting services: you can scale instance storage capacity on the fly. This does not affect your business.

Resource isolation

Instance-level resource isolation provides enhanced stability for individual services.

Data security

- Persistent data storage: based on the hybrid storage of memory and hard disks, KVStore for Redis can provide high-speed data read/write capability and support data persistence.
- Primary/secondary backup and failover: KVStore for Redis backs up data on both a primary node and a secondary node and supports the failover feature to prevent data loss.
- Access control: KVStore for Redis requires password authentication to ensure secure and reliable access.
- Data transmission encryption: KVStore for Redis supports encryption based on Secure Sockets Layer (SSL) and Secure Transport Layer (TLS) to secure data transmission.

High availability

- Primary/secondary structure: each instance runs in this structure to eliminate the possibility of single points of failure (SPOFs) and guarantee high availability.
- Automatic detection and recovery of hardware faults: the system automatically detects hardware faults and performs the failover operation within several seconds. This can minimize your business losses caused by unexpected hardware faults.

Easy to use

- Out-of-the-box service: KVStore for Redis requires no setup or installation. You can use the service immediately after purchase to ensure efficient business deployment.
- Compatible with open-source Redis: KVStore for Redis is compatible with Redis commands. You can use any Redis clients to easily connect to KVStore for Redis and perform data operations.

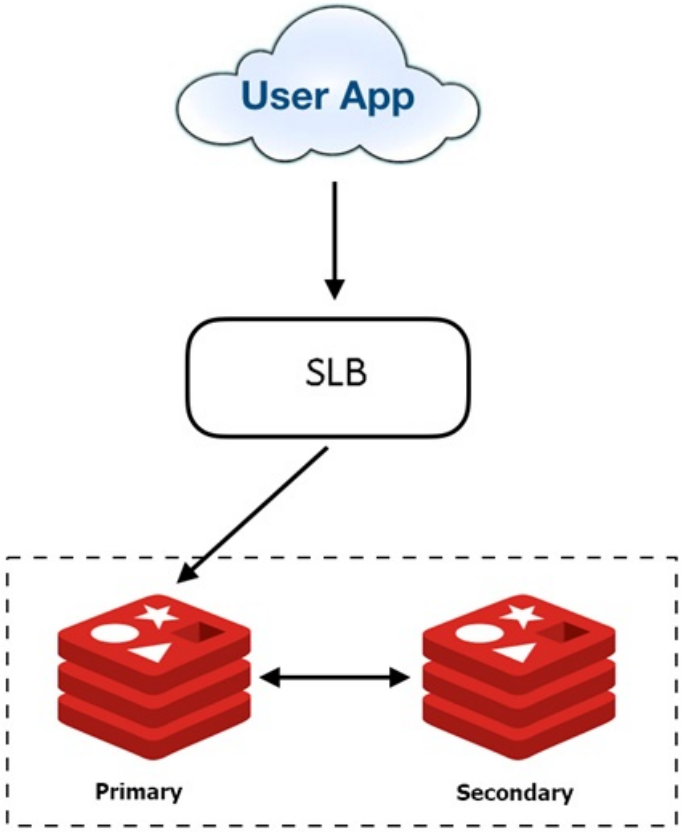
13.3. Architectures

13.3.1. Overall system architecture

KVStore for Redis provides primary/secondary and cluster architecture modes.

Primary/secondary architecture

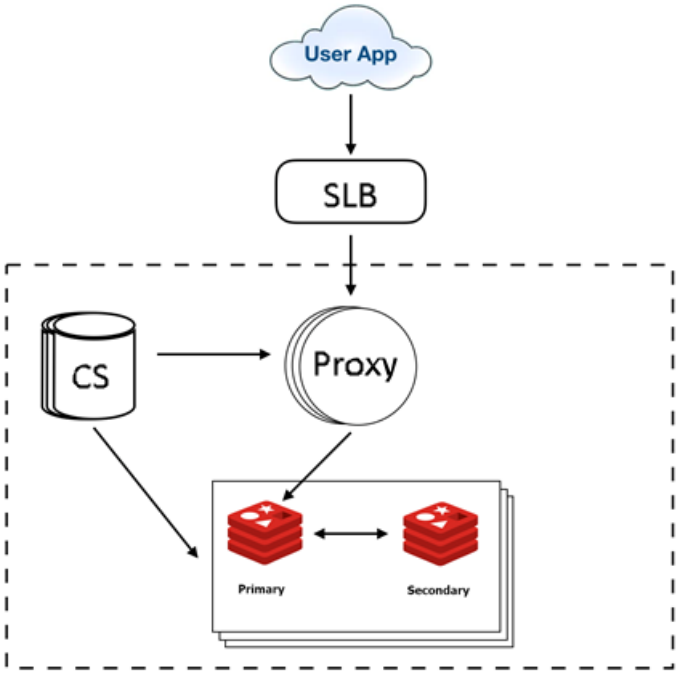
The following figure shows the primary/secondary architecture.



The primary/secondary architecture consists of a primary KVStore for Redis database and a secondary KVStore for Redis database. You can directly access the primary database through an SLB connection.

Cluster architecture

The following figure shows the cluster architecture.



The cluster architecture consists of three components: redis-config (cs), redis-proxy (proxy), and Redis.

The cluster architecture consists of multiple cs nodes, proxy nodes, and primary/secondary Redis nodes. After you access the proxy component through an SLB connection, the proxy component forwards request routes to a shard of the primary Redis database.

13.3.2. Components

This topic describes the components of KVStore for Redis and how these components provide services.

redis-config

The redis-config (cs) component stores the metadata and topology information of the cluster, and performs cluster operations and maintenance. The cs component keeps checking heartbeat messages with the Redis and proxy components, and synchronizes metadata and topology information of clusters to redis and proxy.

redis-proxy

The redis-proxy (proxy) component is the proxy server that connects your client to a Redis server and that implements Redis protocols. The proxy component can authenticate user identities, forward request routes, provide slow and audit logs, and collect monitoring data at an interval of several seconds.

Redis kernel

Alibaba Cloud has optimized the proprietary Redis kernel and developed cloud features based on the open-source kernel from the Redis community.

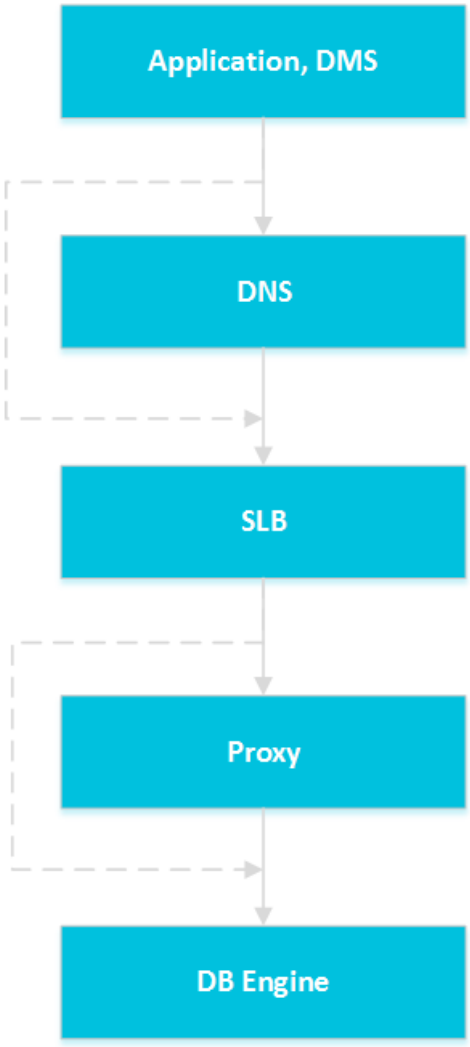
13.4. Features

13.4.1. Data link service

13.4.1.1. Overview

The data link service allows you to add, delete, modify, and search data.

You can connect to the KVStore for Redis service by using your application.



13.4.1.2. DNS

The Domain Name System (DNS) module can dynamically resolve domain names to IP addresses. Therefore, IP address changes cannot affect the performance of KVStore for Redis.

For example, the domain name of a KVStore for Redis instance is test.kvstore.aliyun.com, and the IP address corresponding to this domain name is 10.1.1.1. You can connect to the KVStore for Redis instance if you add test.kvstore.aliyun.com or 10.1.1.1 to the connection pool of your application. If you migrate the KVStore for Redis instance to another host after a failure occurs or upgrades the instance version, the IP address may change to 10.1.1.2. You can connect to the KVStore for Redis instance if you add test.kvstore.aliyun.com to the connection pool of your application. However, if you add 10.1.1.1 to the connection pool, you cannot connect to the instance.

13.4.1.3. SLB

The Server Load Balancer (SLB) module can forward traffic to available instance IP addresses. Therefore, physical server changes cannot affect the performance of KVStore for Redis.

For example, the private IP address of a KVStore for Redis instance is 10.1.1.1. The IP address of the Proxy or DB Engine module is 192.168.0.1. The SLB module forwards all traffic destined for 10.1.1.1 to 192.168.0.1. When the Proxy or DB Engine module fails, the secondary Proxy or DB Engine module with the IP address 192.168.0.2 takes over for 192.168.0.1. The SLB module redirects access traffic from 10.1.1.1 to 192.168.0.2 and the KVStore for Redis instance continues to run normally.

13.4.1.4. Proxy

The Proxy module provides some features such as data routing, traffic detection, and session persistence.

- Data routing: supports partition policies and complex queries for distributed routes based on a cluster architecture.
- Traffic detection: reduces the risks from cyberattacks that exploit Redis vulnerabilities.
- Session persistence: prevents connection interruptions in the case of failures.

13.4.1.5. DB Engine

KVStore for Redis supports standard Redis protocols of the corresponding engine versions as described in the following table.

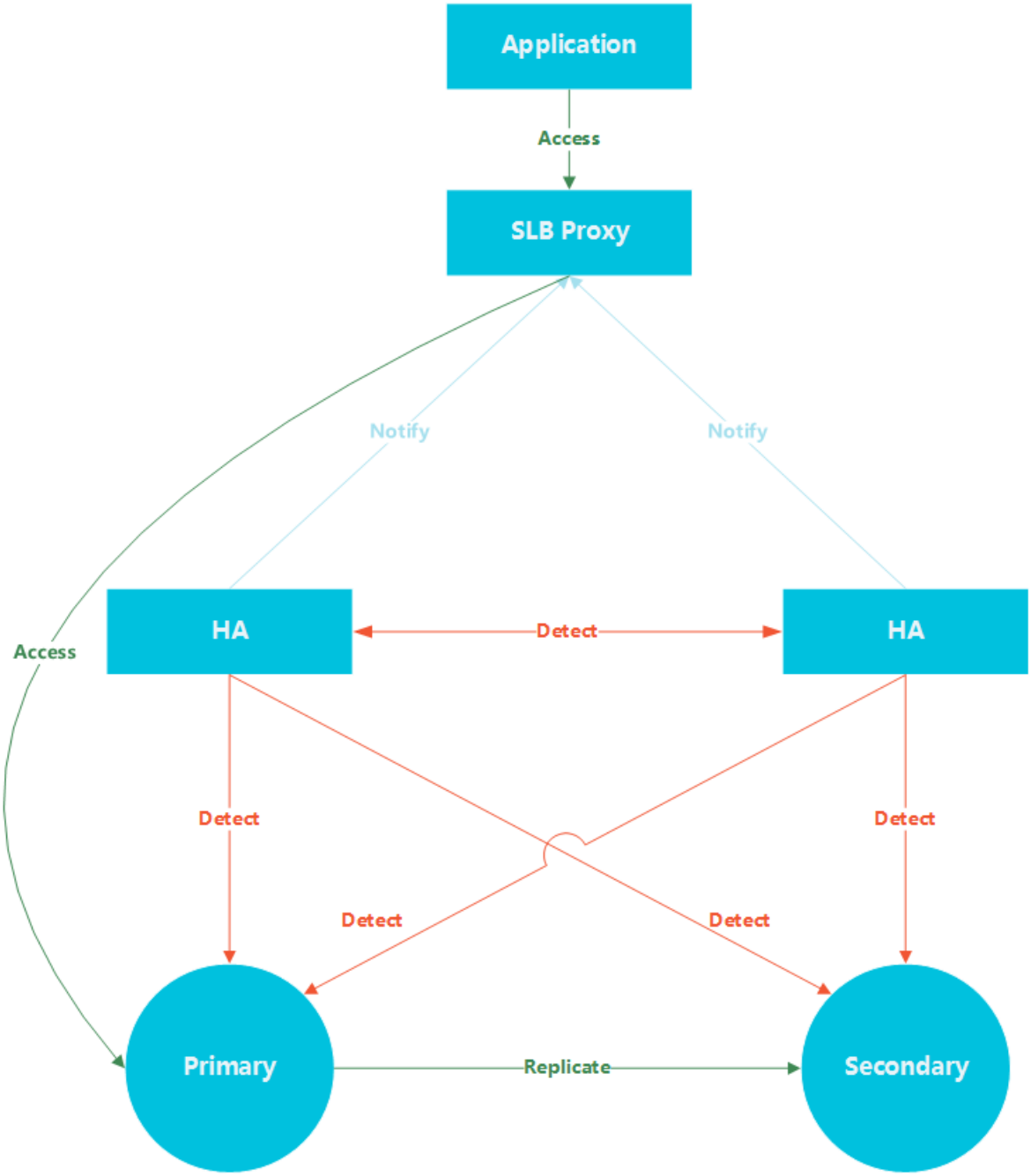
Engine	Version
Redis	Compatible with Redis 2.8 and Redis 3.0 GEO.
Redis	Redis 4.0

13.4.2. HA service

13.4.2.1. Overview

The high-availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

The HA service is also highly available because this service contains multiple HA nodes.



13.4.2.2. Detection

The Detection module checks whether the primary and secondary nodes of the database engine are operating normally.

An HA node receives the heartbeat from the primary database engine node at an interval of 8 to 10 seconds. This information, combined with the heartbeat information of the secondary and other HA nodes, allows the Detection module to eliminate false negatives and positives caused by exceptions such as network jitter. As a result, switchover can be completed within 30 seconds.

13.4.2.3. Repair

The Repair module maintains replications between the primary node and the secondary node of DB Engine. This module also fixes errors that occur on either node during normal operations as follows:

- Automatically fixes exceptionally disconnected replications between these nodes.
- Automatically fixes table-level damages on both nodes.
- Automatically saves crash events and fixes the failures on both nodes.

13.4.2.4. Notice

The Notice module notifies the SLB or Proxy module of status changes of primary and secondary nodes. Therefore, you can connect to available nodes.

For example, the Detection module locates an exception on a primary node and notifies the Repair module to fix the exception. If the Repair module fails to resolve the issue, the Repair module notifies the Notice module to perform failover. Afterward, the Notice module forwards the failover request the Server Load Balancer (SLB) or Proxy module to switch all traffic to the secondary node. Meanwhile, the Repair module creates a secondary node on a different physical server and synchronizes this change to the Detection module. The Detection module checks the health status of the instance again to verify that the instance is healthy.

13.4.3. Monitoring service

13.4.3.1. Service-level monitoring

The independent Service module provides service-level monitoring. The Service module of KVStore for Redis monitors features, response time, and other metrics of other dependent cloud services such as Server Load Balancer (SLB), and checks whether these services run normally.

13.4.3.2. Network-level monitoring

The Network module traces the network status. The monitoring metrics include:

- Connection conditions between ECS instances and KVStore for Redis instances.
- Connection conditions between physical servers of KVStore for Redis.
- Packet loss rates of routers and VSwitches.

13.4.3.3. OS-level monitoring

The operating system (OS) module traces status of hardware and the kernel of an operating system. The monitoring metrics include:

- Hardware inspection: the OS module regularly checks the running status of devices such as CPUs, memory modules, motherboards, and storage devices. When locating any potential hardware failures, the module automatically raises a request for repair.
- OS kernel monitoring: the OS module traces all kernel requests for databases, and analyzes the cause of a slow or error response to a request according to the kernel status.

13.4.3.4. Instance-level monitoring

The Instance module collects information of KVStore for Redis instances. The monitoring metrics include:

- Instance availability.
- Instance capacity.

13.4.4. Scheduling service

The scheduling service allocates and integrates underlying resources of KVStore for Redis, so you can activate and migrate instances.

When you create an instance in the console, the scheduling service computes and selects the most suitable physical server to handle the traffic.

After long-term operations such as instance creation, deletion, and migration, a data center generates resource fragments. The scheduling service can calculate resource fragmentation in the data center and regularly initiates resource integration to improve service performance of the data center.

14.ApsaraDB for MongoDB

14.1. What is ApsaraDB for MongoDB?

ApsaraDB for MongoDB is a stable, reliable, and scalable database service fully compatible with MongoDB protocols. This service provides a full range of database solutions, such as disaster recovery, data backup, data recovery, monitoring, and alerts.

ApsaraDB for MongoDB uses the three-node replica set architecture by default. The primary node supports read/write access, the secondary node provides routine read-only operations, and the standby node is hidden to ensure high availability.

ApsaraDB for MongoDB supports multiple features to ensure the security and availability of services, including:

- Access control: database credential management and IP address whitelist
- Network isolation
- Data backup
- Version maintenance
- Service authorization

14.2. Benefits

High availability

- Three-node replica set high-availability architecture.

  The ApsaraDB for MongoDB service uses a high-availability architecture that features a three-node replica set. The three data nodes are located on different physical servers and automatically synchronize data. The primary and secondary nodes provide services. When the primary node fails, the system automatically selects a new primary node. When the secondary node fails, the standby node takes over the services.

- Automatic backup with a single click.

  Data is automatically backed up and uploaded to Object Storage Service (OSS) every day. This improves data disaster recovery capabilities while effectively reducing disk space consumption. The backup files can be used to restore the instance data to the original instance. This effectively prevents irreversible effects on service data caused by incorrect operations or other reasons.

High security

- Anti-DDoS: This feature monitors traffic at the network ingress in real time. When heavy traffic is identified as an attack, traffic from the source IP addresses is scrubbed. If scrubbing is ineffective, the black hole mechanism is triggered.
- IP whitelist configuration: A maximum of 1,000 IP addresses are allowed to connect to an ApsaraDB for MongoDB instance, directly controlling risks at the source.

Ease of use

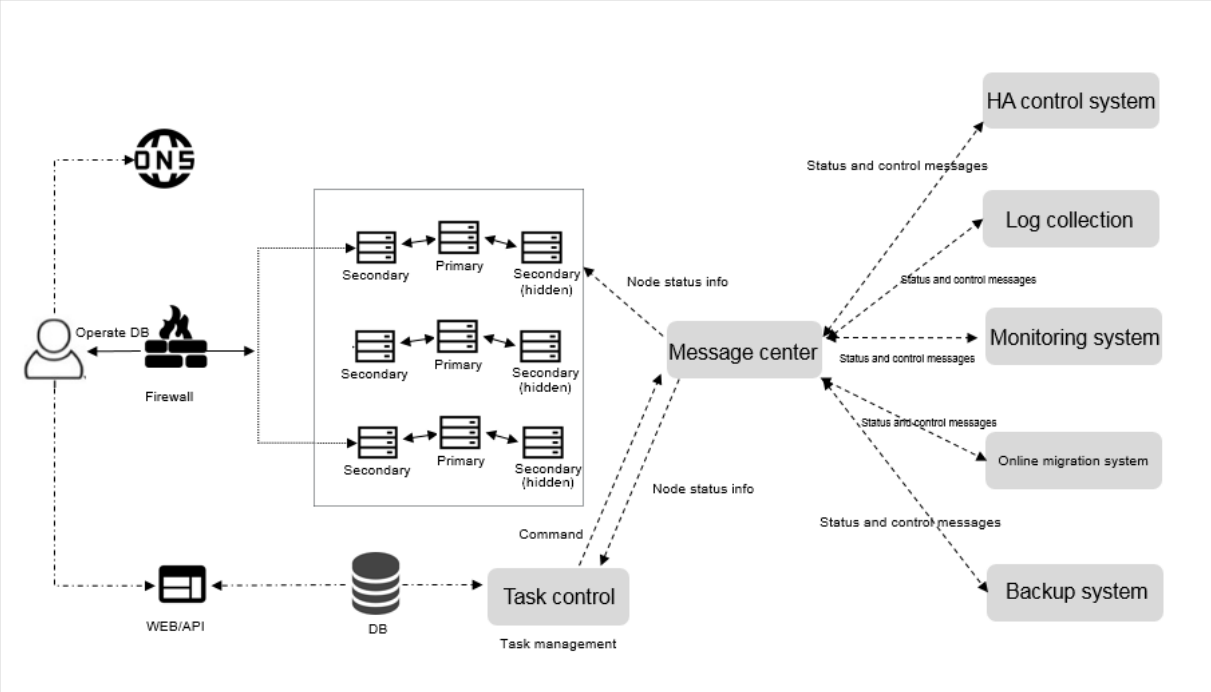
ApsaraDB for MongoDB provides sound performance monitoring. It provides monitoring information about the CPU utilization, IOPS, connections, and disk capacity as well as real-time monitoring and alerting. It enables you to be aware of all instance statuses.

Scalability

ApsaraDB for MongoDB supports three-node replica sets that can be elastically scaled out. You can change the configuration of your instance if the current configuration is too high or is insufficient to meet the performance requirements of your application. The configuration change process is completely transparent and will not affect your business.

14.3. Architecture

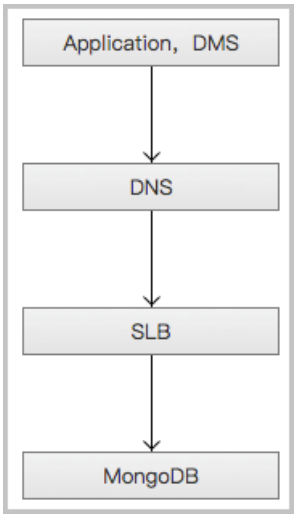
ApsaraDB for MongoDB supports six core services: data link, scheduling, backup, high availability, monitoring, and migration.



14.4. Features

14.4.1. Data link service

This topic describes the data link service, which allows you to perform operations on data.



DNS

For example, the endpoint of a node in an ApsaraDB for MongoDB instance is mongodb.aliyun.com, and the IP address that corresponds to this endpoint is 10.1.1.1. To connect an application to the instance, you can create a connection to mongodb.aliyun.com or 10.1.1.1 in the connection pool.

However, the IP address may change to 10.1.1.2 if the instance is upgraded or migrated. In this case, if mongodb.aliyun.com is configured in the connection pool, the application can still access the instance. If 10.1.1.1 is configured in the connection pool, the application can no longer access the instance.

SLB

The SLB module uses both the private and public IP addresses of an ApsaraDB for MongoDB instance, so server changes do not affect the performance of the instance.

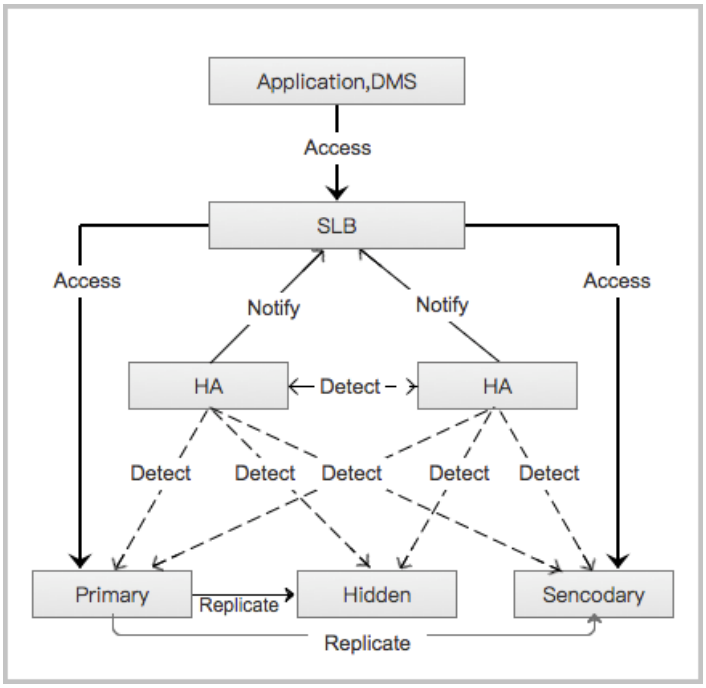
For example, the private IP address of a node in an ApsaraDB for MongoDB instance is 10.1.1.1, and this ApsaraDB for MongoDB instance actually runs on a server whose IP address is 192.168.0.1. Typically, the SLB module forwards all traffic destined for 10.1.1.1 to 192.168.0.1.

If the server with the IP address 192.168.0.1 fails, another server in the hot standby state with the IP address 192.168.0.2 takes over services from the server with the IP address 192.168.0.1. The SLB module then redirects all traffic destined for 10.1.1.1 to 192.168.0.2.

14.4.2. High availability service

The high availability (HA) service guarantees the availability of data link services and handles internal database exceptions.

In addition, this service is based on multiple HA nodes that are also highly available.



Detection

The Detection module detects the running or faulty status of the primary, secondary, and hidden nodes for ApsaraDB for MongoDB. An HA node uses heartbeat information, which is acquired at an interval of 8 to 10 seconds, to determine the health status of the primary node. This information, combined with the heartbeat information of the secondary and hidden nodes, allows the Detection module to eliminate any risk of false negatives and positives caused by exceptions such as network jitters. Switchover can be completed quickly.

Repair

The Repair module maintains the replication relationship among the primary, secondary, and hidden nodes, and fixes faulty nodes or creates new nodes.

Notice

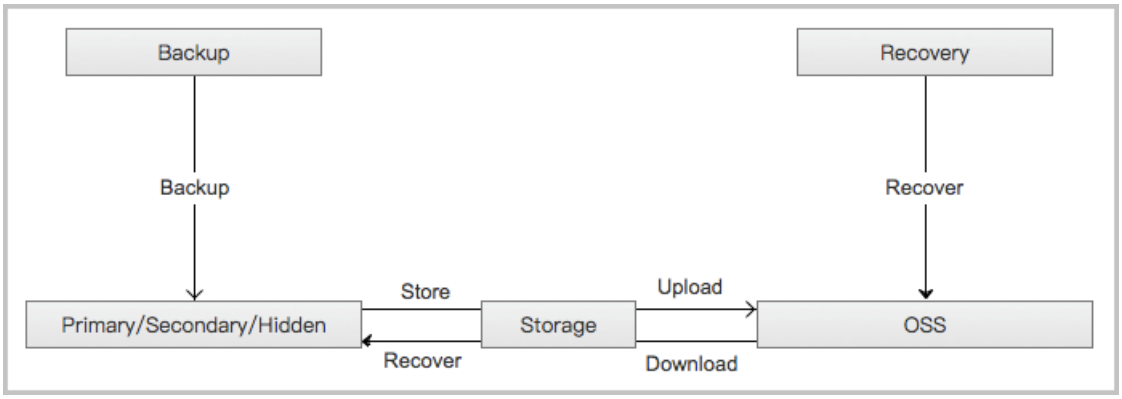
The Notice module informs SLB of node status changes to ensure that you can access the available node.

For example, the Detection module will instruct the Notice module to switch traffic if the Detection module discovers that an exception occurs with the primary node. The Notice module then forwards the switched traffic request to SLB, which redirects traffic from the primary node to the secondary node or from the secondary node to the hidden node. In this circumstance, the secondary node becomes the primary node and the hidden node becomes the secondary node.

During this process, the Repair module attempts to fix the original primary node and convert it to a new hidden node. If the Repair module fails to fix the original primary node, the Repair module will create a new hidden node on another physical server and synchronize the change to the Detection module. The Detection module then incorporates the information and rechecks the health status of the instance.

14.4.3. Backup service

The backup service supports offline data backup, transfer, and recovery.



Backup

The Backup module backs up and compresses data and logs of an instance, and uploads the compressed files to OSS. Data backup in ApsaraDB for MongoDB is performed on the hidden node to avoid affecting services on the primary and secondary nodes.

Recovery

The Recovery module restores backup files stored in OSS to a specified node.

Primary node rollback: You can roll back the settings on the primary node to a specific point in time if you mistakenly perform operations on data.

Secondary and hidden node restore: The system automatically selects a new secondary node to reduce risks when an irreparable failure occurs with the original secondary node.

Storage

The Storage module uploads, dumps, and downloads backup files. Currently, all backup data is uploaded to OSS for storage. You can obtain temporary links to download the data as needed.

14.4.4. Monitoring service

The monitoring service tracks the status of services, networks, operating systems, and instances.

Service

The Service module tracks the status of Alibaba Cloud services. For example, the Service module can monitor SLB, OSS, and SLS services and check whether their functions work as expected and the response time is acceptable. ApsaraDB for MongoDB is dependent on these services. The module also uses corresponding logs to check whether the internal services of ApsaraDB for MongoDB are running properly.

Network

The Network module tracks the status of networks. For example, the Network module can monitor the connectivity between ECS and ApsaraDB for MongoDB instances and among ApsaraDB for MongoDB physical machines. It can also monitor packet loss rates of VRouters and VSwitches.

OS

The OS module tracks the status of hardware and OS kernels.

Examples:

- Hardware inspection: The OS module regularly checks the running status of components such as CPUs, memory modules, motherboards, and storage devices. If the module detects any potential hardware failures, it automatically submits a repair ticket.
- OS kernel monitoring: The OS module tracks all kernel invocations of databases and analyzes the cause of a slow or faulty invocation based on the kernel status.

Instance

The Instance module supports the following features:

- Collects ApsaraDB for MongoDB instance information.
- Provides instance availability information.
- Monitors instance capacity and performance metrics.
- Records statement executions for instances.

14.4.5. Scheduling service

The scheduling service allocates resources and manages instance versions.

Resource

The Resource module allocates and integrates underlying ApsaraDB for MongoDB resources. This module allows you to create and modify instances.

For example, when you use the ApsaraDB for MongoDB console or API to create an instance, the Resource module will calculate and then select the most suitable server to handle the network traffic. After you have created, deleted, and migrated instances multiple times, the Resource module can calculate the resource fragmentation rate in a zone and periodically integrates the resources to improve service capabilities of the zone.

Version

The Version module allows you to upgrade ApsaraDB for MongoDB instances. For example, you can upgrade an ApsaraDB for MongoDB instance to a major version, such as from version 3.2 to version 3.4. You can also upgrade an instance to a minor version that has optimized the source code or kernel as required.

14.4.6. Migration service

The migration service enables you to migrate data from a user-created database to ApsaraDB for MongoDB.

Data Transmission Service (DTS) is a data stream service provided by Alibaba Cloud for data exchanges between data sources. It supports full and incremental migration for ApsaraDB for MongoDB.

- Full data migration: DTS migrates all data from source databases to destination instances.
- Incremental data migration: During incremental migration, the updated data in the local MongoDB database is synchronized to an ApsaraDB for MongoDB instance. Ultimately, the local MongoDB database and the ApsaraDB for MongoDB instance enter the dynamic synchronization process. Incremental migration enables data migration from a local MongoDB database to an ApsaraDB for MongoDB instance without interrupting the services provided by the local MongoDB database.

15.ApsaraDB for OceanBase

15.1. What is ApsaraDB for OceanBase?

ApsaraDB for OceanBase is a distributed relational database service that is developed by Ant Financial. Ant Financial is an affiliate company of Alibaba Group. ApsaraDB for OceanBase incorporates the advantages of traditional relational databases and distributed systems. This allows ApsaraDB for OceanBase to provide high-availability, high-performance, and high-scalability services. ApsaraDB for OceanBase is compatible with most of the MySQL features. ApsaraDB for OceanBase uses general hardware devices to provide financial-grade database services that have high availability.

The data that is stored in ApsaraDB for OceanBase includes baseline data and incremental data. Baseline data is read-only data and incremental data is the data that is added, deleted, or modified. ApsaraDB for OceanBase performs major freeze operations to merge incremental data with baseline data on a regular basis.

ApsaraDB for OceanBase is a cloud-based database service that implements the multitenancy architecture. The architecture allows a single instance to provide services for multiple tenants. In this architecture, each tenant is allocated with a specified set of tenant resources based on the business requirements of the tenant. You can allocate the following resources to each tenant: CPU cores, memory size, and storage space.

15.2. Technical benefits

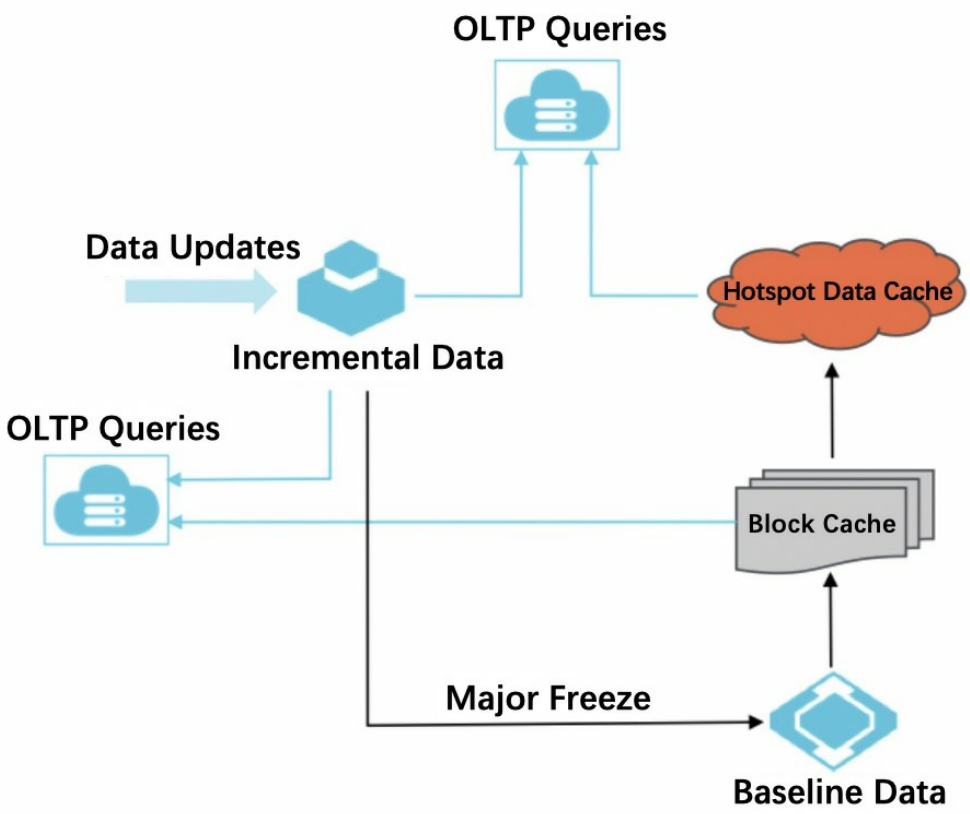
15.2.1. High-efficiency storage engine

ApsaraDB for OceanBase uses a shared-nothing distributed architecture. In this architecture, each OBServer is independent and provides the same features. The partitions that each OBServer manages are different from the partitions that each of the other OBServers manages.

The storage engine for a single OBServer uses a read/write splitting architecture. The storage engine uses MemTables to store updated dynamic data in the memory. The storage engine uses sorted string tables (SSTables) to store baseline data in disks.

All the data for each partition is stored on an OBServer, such as baseline data, incremental data, and transaction log records. Therefore, each data read and write operation for a partition is performed on only one OBServer. If you need to write data to partitions on multiple OBServers, you can perform concurrent write operations.

High-efficiency storage engine in ApsaraDB for OceanBase



The storage engine uses a read/write splitting architecture. This architecture offers diverse benefits. For example, you can compress large amounts of static baseline data to reduce storage costs. This architecture also eliminates your concerns on row cache expiration that is caused by data writes. The disadvantage is that the architecture results in complex procedures of data reads. To reduce the complexity, the system must merge data in real time. However, this may compromise system performance. To resolve this issue, ApsaraDB for OceanBase provides various optimization methods. For example, the Bloom filter cache filter outs the rows that do not exist. The Bloom filter cache executes the INSERT ROW statement to check whether the row exists. If the row does not exist, the I/O operation for this row is not required. The system preferably reads the updated data from active MemTables. If the system retrieves the required columns from the active MemTables, the system does not need to read baseline data or merge incremental data with baseline data.

Incremental data is written in the memory. After the amount of the stored incremental data reaches a specified threshold, the system merges the incremental data and the baseline data to generate new baseline data. This process is known as major freeze. Major freeze operations cause additional loads. This may affect the requests from clients. To resolve this issue, you can use a rotated policy to perform polling major freeze operations. To be specific, if you need to merge incremental data with baseline data in multiple data centers of ApsaraDB for OceanBase, you can switch the requests for one of the data centers to another data center. After the major freeze operation is complete in the original data center, you can switch the requests back to the original data center. This policy allows you to eliminate the impact of major freeze operations on your business. You can use the rotated policy during system upgrades and maintenance. Before you upgrade a version, you can switch the requests from one OBServer to another OBServer. After the upgrade is complete, you can perform a phased switchover to switch the requests back to the original OBServer based on the percentages of requests. This way, you can immediately perform a rollback after a failure occurs during the switchover. This prevents data losses.

15.2.2. Scalability

In ApsaraDB for OceanBase, you can use table partitioning methods of traditional relational databases to divide a table into partitions. The syntax for table partitioning in ApsaraDB for OceanBase is compatible with that in traditional relational databases. This allows you to use the benefits of distributed systems and relational databases. Distributed systems provide scalability capabilities and relational databases provide the following benefits: easy-to-use features and flexibility. The compatibility allows database administrators (DBAs) to use their familiar operations to manage partitions.

ApsaraDB for OceanBase supports online linear scaling. If the storage capacity or the processing capability of a cluster cannot meet the business requirements, you can add new OBServers to meet the requirements. Then, the system automatically migrates appropriate partitions to the added OBServer based on the processing capability of the server. If a cluster provides more services than required based on its storage capacity and the processing capability, you can delete OBServers to reduce costs. In sales promotion events, such as the Double 11, ApsaraDB for OceanBase provides high-scalability services.

All the data reads and writes for a partition are performed on the OBServer where the partition is stored. If a transaction needs to be performed on multiple partitions, the two-phase commit protocol is used to perform the transaction on the OBServers where the partitions are deployed.

In ApsaraDB for OceanBase, transactions are divided into the following types:

- Single-partition transactions: The transactions of this type are similar to those in traditional relational databases. Single-partition transactions are performed on a single server and the two-phase commit protocol is not required.
- Single-server multi-partition transactions: The transactions of this type are performed on multiple partitions of a single server based on the two-phase commit protocol. The protocol is optimized for the transactions of this type.
- Multi-server multi-partition transactions: The transactions of this type are performed on multiple partitions of serves based on the two-phase commit protocol.

The standard two-phase commit protocol requires two phases:

1. Prepare phase: The coordinator sends a prepare request to all the participants. After the participants replicate redo logs and prepare logs to replicas and write the logs to disks of replicas, the participants send a response to the coordinator. This process is known as log durability.

   > **Note** Log durability is successful only after the logs are replicated to a majority of replicas and are written to the disks of the replicas.

2. Commit or abort phase: If the operations in the prepare phase are successful for all the participants, the coordinator sends a commit request to each participant. Otherwise, the coordinator sends an abort request to each participant. After the participants replicate commit logs or abort logs to replicas and write the logs to disks of replicas, the participants send a response to the coordinator.

The status of distributed transactions can be determined only after the commit or abort phase is complete. ApsaraDB for OceanBase optimizes the two-phase commit protocol by adding the clear phase. The operations in the clear phase are asynchronously performed in the background. The clear phase is added to recycle the resources that are requested in the prepare and commit or abort phases.

The result of a multi-server multi-partition transaction is sent to the client only after the operations in the commit or abort phase are complete.

The result of a single-server multi-partition transaction is sent to the client after the operations in the prepare phase are complete. This reduces the latency of the single-server multi-partition transaction.

ApsaraDB for OceanBase uses the multiversion concurrency control (MVCC) method to manage concurrent operations. The method ensures that read transactions and write transactions can be performed in a concurrent way and do not affect each other. If a read request retrieves data that is stored in one or more partitions of a single OBServer, the system needs only to read a specific snapshot on the OBServer. The specific snapshot is created at the required point in time. If a read request retrieves data that is stored in multiple partitions of OBServers, the system must read the snapshots that are distributed across the partitions.

15.2.3. Paxos-based log synchronization

Each partition in ApsaraDB for OceanBase has multiple replicas. In most cases, each partition has three replicas. The three replicas are deployed in three different data centers or zones.

In ApsaraDB for OceanBase, tens of millions of partitions may exist. ApsaraDB for OceanBase uses the Paxos protocol to replicate and synchronize logs among the replicas of these partitions.

An independent Paxos replication group consists of a partition and the replicas of the partition. In each replication group, one replica is a leader replica and the other replicas are follower replicas.

The partitions on each OBServer are divided into two types: leader partitions and follower partitions. When an OBServer is faulty, the follower partitions are not affected and the write services of the leader partitions are affected for a short period. For each affected partition, a follower replica on the server is elected as a new leader replica based on the Paxos protocol. The failover process requires about 20 seconds to 30 seconds.

The Paxos protocol is used for Paxos-based distributed election and log replication.

Paxos-based distributed election ensures that only one leader replica is always elected for each partition. The leader replica replicates logs to the follower replicas for synchronization. The system considers a transaction successful only after the log entries of the transaction are replicated to a majority of replicas and are written to the required disks.

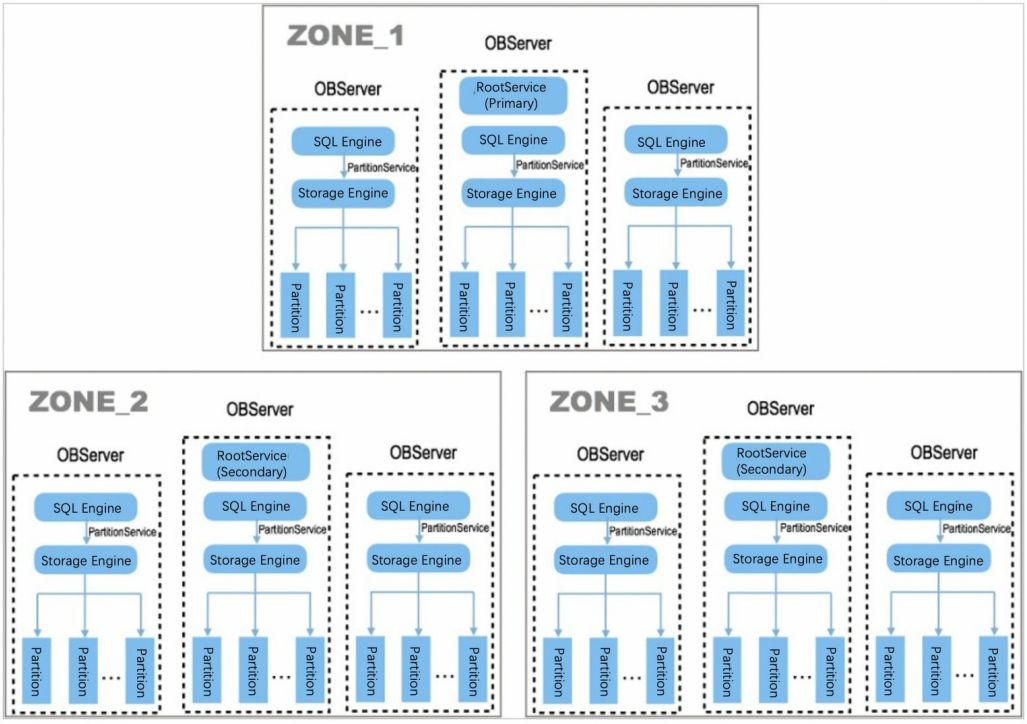
Assume that a partition has three replicas. For each committed transaction, you must ensure that the transaction log entries are written to the disk in the leader replica. You must also ensure that the log entries are replicated from the leader replica to one of the follower replicas and are written to the disk of the follower replica. If a follower replica is faulty, the leader replica and the remaining follower replica can form a majority of replicas. This allows the system to continue providing services. If the leader replica is faulty, a few latest transactions that are performed in the leader replica may not be synchronized to a majority of replicas. These transactions are known as in-doubt transactions because the status of the transactions cannot be determined. The status of the transactions can be determined only after a follower replica serves as a new leader replica. The first operation that the new leader replica performs is to reconfirm the status of the in-doubt transactions. After the reconfirm process is complete, the new leader replica starts to provide services and the system recovers.

After a partition is replicated or migrated, a change occurs on the members of the Paxos replication group for the partition. In this scenario, the system must change the members in the group. A member change requires a voting process. The member change must be approved by two groups of members. One member group consists of the majority of members before the change. The other member group consists of the majority of members after the change.

15.3. Architecture

Architecture of ApsaraDB for OceanBase shows the architecture of ApsaraDB for OceanBase.

Architecture of ApsaraDB for OceanBase



In most cases, three replicas are deployed in ApsaraDB for OceanBase. Each replica is stored in a zone. Each zone hosts multiple physical OBServers. The roles of nodes in each zone are divided into two types: RootService and PartitionService. Each PartitionService node manages multiple partitions and consists of two components: SQL engine and storage engine. The lease mechanism is used between all the OBServers in a cluster and the RootService nodes in the cluster. If an OBServer is faulty, the RootService node can detect the fault and rectify the fault to recover the server. RootService and PartitionService nodes are deployed on OBServers. You can deploy a RootService node and a PartitionService node on the same OBServer. The system automatically elects the RootService node. When you deploy the service, you must specify the list of IP addresses for the RootService nodes.

15.4. Principles

15.4.1. Multitenancy

The multitenancy architecture of ApsaraDB for OceanBase offers the following benefits:

- Multitenancy allows you to reduce the operations and maintenance (O&M) costs and simplify the O&M operations on multiple database instances. The costs and complexity of O&M operations for multiple database instances are similar to those for a single database instance.

  ApsaraDB for OceanBase provides large clusters. This allows ApsaraDB for OceanBase to offer cost advantages that are brought by the service scale. Each ApsaraDB for OceanBase cluster can provide multiple database instances whose management costs are the same as those for one database instance.

- Multitenancy helps you maximize the utilization of resources. Compared with non-multitenancy architectures, multitenancy allows the same set of the resources to support more services. If your service systems have different peak hours and off-peak hours, you can deploy the service systems in the same cluster. This maximizes the utilization of system resources.

- Tenants in each cluster are isolated from each other.

  Cross-tenant data access is not allowed. This eliminates the risks of data leakage and data breach for each tenant.

  Each tenant has exclusive use of the resources that are allocated to the tenant. The frontend applications of each tenant offer stable performance. The performance is measured by the following performance metrics: response time, transactions per second (TPS), and queries per second (QPS). These performance metrics for each tenant are not affected by the service loads of the other tenants.

- If a worker thread of an OBServer processes an SQL request and the processing time exceeds the duration of a time slice, the request is automatically distributed to the scheduler. Note that the duration of a time slice is about 10 milliseconds. Then, based on service loads, the scheduler determines whether to continue processing the request or move the worker thread of the SQL request to the task queue. If the latter option is used, the scheduler runs other threads to continue processing other requests.

15.4.2. Compatibility with MySQL

ApsaraDB for OceanBase is compatible with MySQL. You can run MySQL applications in ApsaraDB for OceanBase and do not need to modify the applications.

ApsaraDB for OceanBase is compatible with MySQL in the following aspects:

- Interfaces

  The Java Database Connectivity (JDBC) interface is widely used in ApsaraDB for OceanBase. ApsaraDB for OceanBase improves the compatibility with MySQL in terms of frontend and backend protocols. This allows you to use the MySQL driver to access ApsaraDB for OceanBase.

- Data objects

  ApsaraDB for OceanBase supports standard SQL objects and objects that are specific to MySQL. The standard SQL objects include databases, tables, views, and auto-increment columns. ApsaraDB for OceanBase implements multitenancy in database systems.

  The service that runs on different MySQL instances can be seamlessly migrated to an ApsaraDB for OceanBase cluster.

- Statements and data types

  The SQL statements of major database products comply with the specifications that are defined in the ISO/IEC 9075 standard. The latest SQL standard version is SQL:2011.

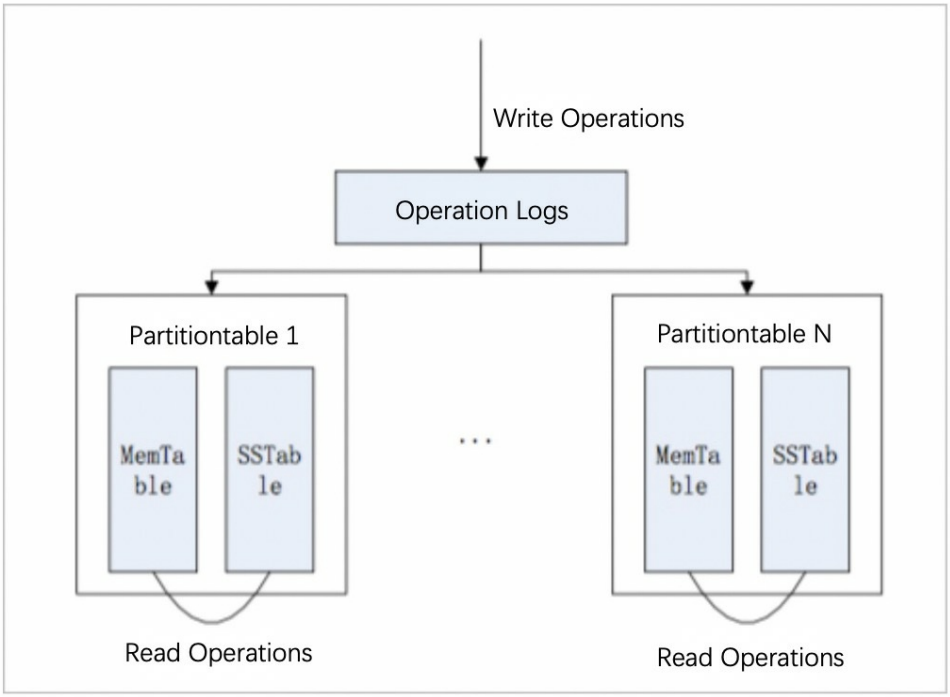
- Transactions

  ApsaraDB for OceanBase is compatible with MySQL in terms of transaction isolation levels and transaction concurrency control.

  ApsaraDB for OceanBase uses the MVCC protocol. This allows you to perform data reads or writes in parallel. ApsaraDB for OceanBase supports the read committed isolation level.

15.4.3. Engine of a single OBServer

Schematic diagram: Engine of a single OBServer



Schematic diagram: Engine of a single OBServer shows the working principle for the engine of a single OBServer in ApsaraDB for OceanBase. Each partition uses a MemTable to store incremental modification data. In ApsaraDB for OceanBase, the data that must be replicated and migrated consists of the baseline data in SSTables and the incremental data in MemTables.

Before data is written to partition replicas, the OBServer uses a redo log to record data changes. Then, the OBServer replicates the redo log to a majority of partition replicas over the Paxos protocol. After the replication is completed, the OBServer writes the data to the memory of the partition replicas. The group commit method is used to improve the I/O throughput of the system. ApsaraDB for OceanBase provides services that are similar to in-memory databases. Log writes in the service must be asynchronously replicated. To be specific, an SQL thread can process the next task immediately after the SQL thread commits a log write task. After the log write task is completed, the thread sends a response to the client. This ensures that log write tasks are committed in an optimal way.

If the amount of incremental modified data in the memory exceeds a specified threshold, a minor freeze or a major freeze operation is triggered. In a minor freeze operation, the data in the memory is written to a disk to generate compaction SSTables. In a major freeze operation, data in baseline SSTables, compaction SSTables, and MemTables are merged into a new baseline SSTable. To improve read efficiency, you must limit the maximum number of compaction SSTables. In ApsaraDB for OceanBase, only one valid compaction SSTable is reserved. If a valid compaction SSTable already exists when you perform the next minor freeze operation, the data that is written to the MemTable is merged into the compaction SSTable.

In most cases, the amount of daily incremental modification data is lower than that of baseline data. Therefore, a minor freeze or a major freeze operation is triggered at a low probability. Most of the operations that are involved in transactions are performed in the MemTable of each partition. Each partition is equivalent to an in-memory database engine.

15.4.4. Memory transaction engine

The memory transaction engine consists of two components: memory index structure and MVCC engine.

In ApsaraDB for OceanBase, MemTables use B+ tree indexes and hash indexes. B+ tree indexes are used to optimize range queries. Hash indexes are used to optimize queries that retrieve data from a single row.

MemTables automatically ensures the consistency between B+ tree indexes and hash indexes each time transactions are performed.

The B+ tree indexes that are used in the memory of ApsaraDB for OceanBase provide high-performance services. Another benefit of B+ tree indexes is that few concurrency conflicts occur. In performance tests, the B+ tree indexes show higher performance than the alternative structures that are used in other in-memory databases, such as lock-free skip lists in MemSQL databases.

MemTables record historical transactions in the memory. The operations that are specified in historical transactions for each row are sorted in chronological order. These operations are organized as a row operation chain. Each time a new transaction is committed, the information about the new operation is added to the end of the row operation chain. If the row operation chain records an excessively large number of historical transactions, the read performance is compromised. In this scenario, a compaction operation is triggered to merge the historical transactions into a new row operation chain.

For example, the following three row operations are performed on a row: `{(c1, 1), (c2, 'a')}`, `{(c1, 2), (c3, true)}`, and `{(c1, 3), (c2, "b")}`. After the compaction operation is performed to merge the three row operations, the new row operation chain records only one row operation: `{(c1, 3), (c2, "b"), (c3, true)}`. Each compaction operation does not delete the original row operation chain. The new row operation chain provides a backward pointer that points to the original chain. To read the earlier historical snapshots, you need only to use the backward pointer in the memory to trace the historical snapshots. This operation is similar to the undo operation in databases.

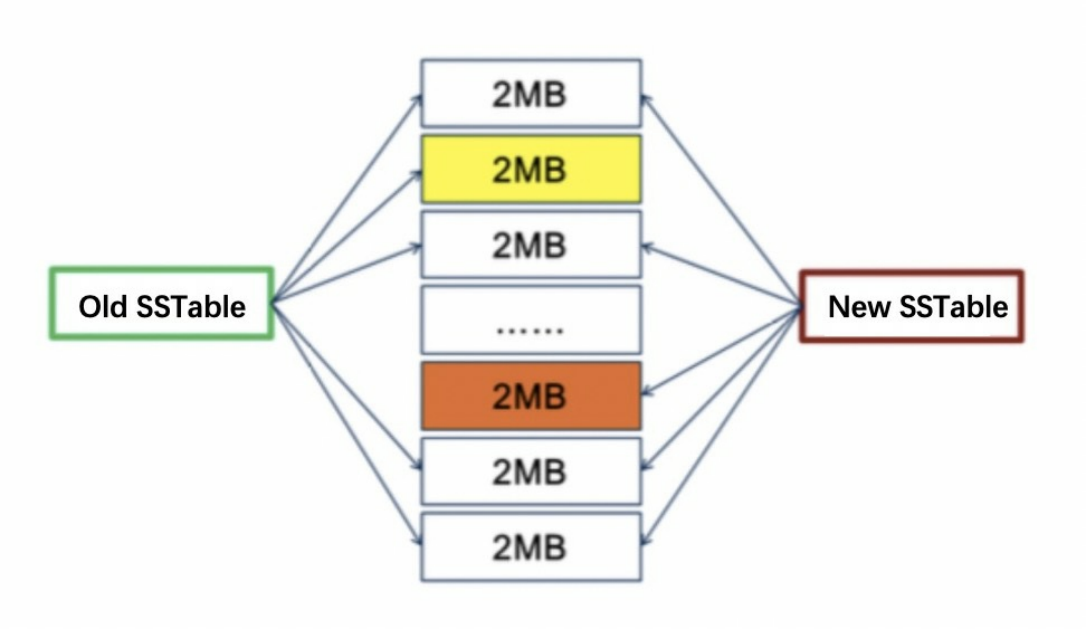
Follower replicas receive the redo logs that leader replicas send over the Paxos protocol and replay the redo logs to MemTables.

Concurrent threads can be used to replay logs to MemTables. The replay overhead that is generated by follower replicas is significantly lower than the overhead of the transactions that are performed by leader replicas. This ensures that the data in the follower replicas is consistent with the data in the leader replicas during peak hours.

15.4.5. Baseline data storage

ApsaraDB for OceanBase uses the SSTable data structure to store baseline data. The name SSTable has its origin in the Google Bigtable system.

Baseline data storage



Baseline data storage shows that an SSTable is divided into multiple 2 MB macro-blocks. This reduces the overhead of major freeze operations for SSTables. After each major freeze operation, the data in a previous SSTable and the data in a MemTable are merged into a new SSTable. If a macro-block of the previous SSTable is not modified, the data in the macro-block does not need to be rewritten. If requests are sent to read the data from this macro-block, the new SSTable needs only to forward the read requests to the previous macro-block. If a macro-block of the previous SSTable is modified, the modifications of this macro-block and the corresponding data in the MemTable are merged into a new macro-block. For most of the services, the ratio of modified macro-blocks is low. Therefore, the macro-block method reduces the overhead of major freeze operations and the disk space that is occupied by SSTables.

Each macro-block is divided into multiple micro-blocks. The size of each micro-block ranges from 4 KB to 64 KB. The data rows in micro-blocks are sorted based on the primary key. SSTables store the index structure of macro-blocks. Macro-blocks store the index structure of micro-blocks. The index structures are sorted arrays. Micro-blocks are similar to pages in a relational database. Micro-blocks are the basic unit of reading data from the block cache.

SSTables store the information about row cache, block index cache, and Bloom filter cache, and block cache.

If you need to query data from a single row in an SSTable and the row exists, a cache hit occurs for the row cache. If the row does not exist, a cache hit occurs for the Bloom filter cache. For most of the queries on a single row, data needs to be retrieved only once from the cache in baseline data storage. This does not cause additional overheads.

15.4.6. RootService nodes

RootService nodes are feature modules of OBServer processes in ApsaraDB for OceanBase. RootService nodes are similar to PartitionService nodes in this aspect.

The ApsaraDB for OceanBase system has an initial partition: __all_core_table. The initial partition can be used to index the other partitions based on different index levels. By default, the OBServer where the leader replica of the initial partition resides provides RootService features.

RootService nodes provide the following features: server and zone management, partition management, management of daily major freeze operations, system bootstrapping, and data definition language (DDL) operations.

Server and zone management

The lease mechanism is used between all the OBServers in a cluster and the RootService nodes in the cluster. You can use the RootService nodes to enable or disable OBServers and zones.

Partition management

A RootService node manages the distribution of data across partitions in a cluster and initiates operations, such as replication, migration, splitting, and major freeze for partitions.

If an OBServer is faulty, the Paxos protocol is used to switch the services of the leader replicas for partitions on the failed server to other replicas in another OBServer. The new replicas serve as the leader replicas.

If the failed OBServer recovers within a short period after the failure occurs, the partition replicas on this server are added to the Paxos replication group. If the failed OBServer is unavailable for a long period, the partition replicas on this server are permanently removed. Then, the RootService node initiates partition replication to create new partition replicas on another OBServer.

When a RootService node manages partitions, the node must implement load balancing based on the following factors of each OBServer: CPU utilization, disk usage, memory usage, and IOPS. Load balancing prevents the partitions of a table from being distributed to only a few OBServers.

Management of daily major freeze operations

In ApsaraDB for OceanBase, the system performs major freeze operations to merge dynamic data and static data at the specified time every day. In most cases, the operations are performed during off-peak hours.

A RootService node coordinates the major freeze operations and ensures that the operations are simultaneously performed on all the OBServers in each cluster.

To mitigate the impact of daily major freeze operations on your business, ApsaraDB for OceanBase allows you to perform major freeze operations during off-peak hours by zone. In this method, before the major freeze operations are performed, the RootService node switches the services in all the leader replicas of partitions from a zone to another zone. After the major freeze operations are complete, the RootService node switches the services back to the original zone. After the leader replicas of partitions are switched to another zone, read and write requests are also automatically switched to the new leader replicas. During the major freeze operations, make sure that few external requests are routed to the zones where the operations are being performed.

DDL statements

A RootService node allows you to execute DDL statements. For example, you can execute DDL statements to create tables, delete tables, add columns, delete columns, modify column attributes, and add indexes.

System bootstrapping

Bootstrapping is an installation process to initialize a system. Bootstrapping is initiated to create internal system tables and initialize system settings.

15.4.7. OBProxy

The applications of ApsaraDB for OceanBase can use the OBProxy to access ApsaraDB for OceanBase services. The applications communicate with the OBProxy over the standard MySQL protocol. For applications, ApsaraDB for OceanBase is similar to MySQL, but ApsaraDB for OceanBase provides higher performance than MySQL.

The OBProxy is a reverse proxy server that offers high-performance services. You can manage and perform O&M operations on the OBProxy in an easy way. The OBProxy provides the following features:

- Reverse proxy services

  The OBProxy forwards a request from a client to the OBServer where the requested data resides and sends the response from the OBServer to the client. The OBProxy must also prevent transient connections when the OBServer breaks down or is upgraded. This ensures that the requests from the client can be processed as expected. The OBProxy is compatible with all the MySQL clients. When you initiate an SQL request, the OBProxy parses the request and finds the OBServer based on the requested partition. The OBProxy also processes system events such as OBServer failures and off-peak major freeze operations in ApsaraDB for OceanBase clusters.

- O&M requirements

  The OBProxy can meet the following O&M requirements:

  - The OBProxy supports hot upgrades.
  - The OBProxy supports automatic configuration updates.
  - The OBProxy provides a wide range of security features. For example, you can configure the IP address whitelist, prevent SQL injection, and traffic shaping.
  - You can deploy the OBProxy on an independent server or on a client.

15.4.8. Backup and restoration

ApsaraDB for OceanBase is a distributed relational database that offers high-availability and high-performance services. Backup and restoration is an essential feature of ApsaraDB for OceanBase. The feature allows you to ensure high-availability and high-performance services. In ApsaraDB for OceanBase, you can use various storage tools such as Alibaba Cloud Object Storage Service (OSS) and disk arrays for backup and restoration. You can use a backup and restoration tool of ApsaraDB for OceanBase to back up data for multiple ApsaraDB for OceanBase clusters at a time. This is known as the 1+N backup. You can use the backup and restoration feature to back up and restore data by cluster or by tenant. You can also restore data to a specific point in time. The backup and restoration feature allows you to back up and restore the data of all the operations that are performed in databases. The backup data supports all the physical data and some logical data. The supported logical data includes user permissions, table definitions, system variables, user information, and view information.

High availability

The backup and restoration feature allows you to use a third-party metadatabase to record the information about all of the backup and restoration tasks. No dependencies exist among these tasks. AgentServer is a stateless backup and restoration tool. This tool can be deployed on one or more servers. If you deploy the tool on multiple servers, high availability can be ensured. If the physical server where AgentServer is deployed is faulty, the other AgentServer servers continue to run the backup tasks. This ensures that data backups for the cluster are not affected.

High performance

The AgentServer tool obtains information about multiple tasks at a time from the metadatabase. The backup rate depends on the upload and download rates of the tools that you use to back up data. The AgentServer tool does not cause performance bottlenecks for the backup rate. If you run backup tasks on a 10 GE server, the transmission rate of the network interface controller (NIC) on the server can reach up to 700 Mbit/s.

Ease of use

ApsaraDB for OceanBase supports multitenancy. You can run backup and restoration tasks to back up and restore data by cluster or by tenant. If you back up or restore data by tenant, you can run the tasks for a specified tenant. You can also use the AgentServer tool to back up data for multiple ApsaraDB for OceanBase clusters. This reduces the costs of backup and recovery tools if multiple clusters run in your system.

Various application scenarios

The backup and restoration feature is applicable to various business scenarios, such as cold backups of database data. For example, a database can serve as an image database. You can use the image database to restore online service data of a tenant to a specified cluster. Then, you can verify the services in the staging phase for a tenant of the cluster. In this process, online service data instead of offline data is used for verification. This ensures the accuracy of verification results because the online service data of the tenant is generated from production environments. This also allows you to avoid issues that occur due to the data amount during the execution of SQL plans.

15.5. Disaster recovery solutions and deployment architectures

15.5.1. Overview

ApsaraDB for OceanBase uses the Paxos protocol at the underlying layer to ensure high availability and strong consistency.

Paxos is a protocol for synchronous replication. This protocol cannot eliminate network latencies in theory. Therefore, when you design a disaster recovery solution, you must consider the impacts of data center architectures and network architectures on service deployment.

Some traditional databases use a primary/secondary architecture that requires only two replicas. ApsaraDB for OceanBase uses the Paxos protocol that requires at least three replicas. In some scenarios, five replicas are required. To prevent replicas from occupying large amounts of resources, ApsaraDB for OceanBase must provide appropriate disaster recovery solutions and deployment architectures.

15.5.2. Disaster recovery

ApsaraDB for OceanBase provides capabilities of automatic non-disruptive disaster recovery. The disaster recovery capabilities are divided based on the following levels:

- Non-disruptive disaster recovery across servers: If a single server is faulty, the services on the faulty server are automatically switched to another server and no data loss occurs.
- Non-disruptive disaster recovery across data centers or zones: If a single data center is faulty, the services in the faulty data center are automatically switched to another data center and no data loss occurs.
- Non-disruptive disaster recovery across regions: If all the data centers in a region are faulty, the

services in the region are automatically switched to the data centers in another region and no data loss occurs.

15.5.3. Deployment architectures

15.5.3.1. Three Data Centers Across Two Regions

Disaster recovery across regions is an optimal disaster recovery solution.

For example, one production system is deployed around the globe for Google Cloud Spanner. Each write is replicated to multiple cities across regions over the Paxos protocol. This deployment method causes long write latencies. In the Cloud Spanner system, the replication latency for each write ranges from tens of milliseconds to hundreds of milliseconds. The latency varies based on the geological distances between the regions where replicas reside.

If you use the Three Data Centers Across Two Regions architecture, the replication latency across regions increases each time you commit a transaction.

The replication latency between data centers across regions such as China (Shanghai) and China (Shenzhen) in China is about tens of milliseconds. In most cases, each business process involves multiple database transactions. Therefore, you may need to optimize your business processes to reduce the number of database transactions.

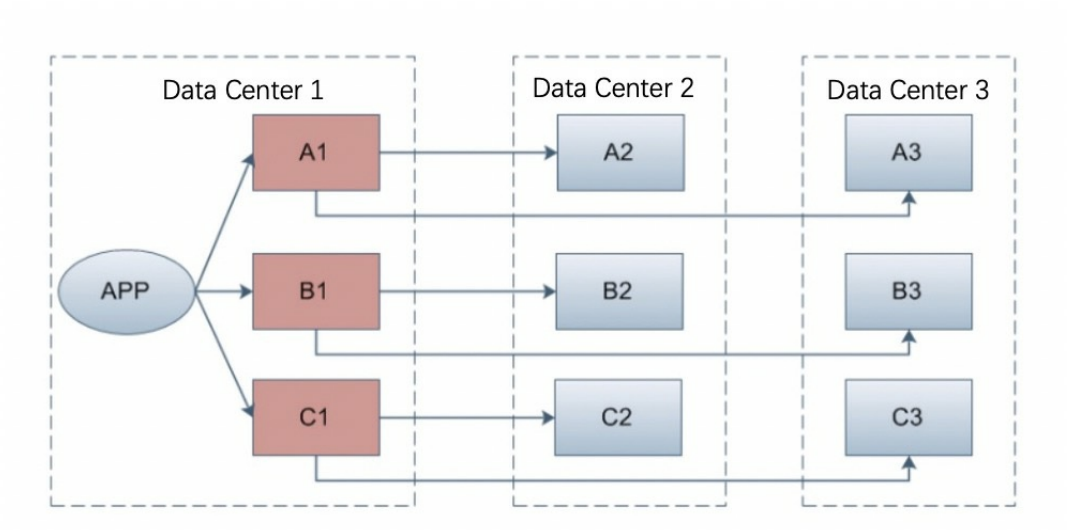
15.5.3.2. Three Data Centers in the Same Region

The Three Data Centers in the Same Region architecture supports only disaster recovery across data centers. This is a compromise solution for disaster recovery.

Based on the principles of the Paxos protocol, at least three data centers must be deployed to support disaster recovery across data centers. Assume that only two data centers are deployed. If one of the data centers is faulty, the other data center cannot form a majority. As a result, the services in the faulty data center cannot be switched to the other data center and data losses occur.

If you deploy three data centers, one data center hosts the primary database and the other two centers host the secondary databases. Three Data Centers in the Same Region shows that the application and the primary database are deployed in Data center 1.

Three Data Centers in the Same Region



In normal cases, the application accesses the leader partitions that are deployed in the same data center as the application. In this example, the application accesses A1, B1, and C1 in Data center 1.

When ApsaraDB for OceanBase in Data center 1 is faulty, the services in the leader partitions of the faulty system are switched to the partitions in Data center 2 or Data center 3. For example, the services that are provided by the A1, B1, and C1 leader partitions can be switched to the A2, B3, and C2 partitions. If this occurs, A2, B3, and C2 serve as the new leader partition. The application that is deployed in Data center 1 accesses the new leader partitions in Data center 1: A2, B3, and C2. After Data center 1 recovers from the failure, ApsaraDB for OceanBase switches the services back to A1, B1, and C1. This prevents the application from accessing services across data centers. In this scenario, A1, B1, and C1 serve as the leader partitions again. Data center 1 is the preferred data center to provide the ApsaraDB for OceanBase services. The system identifies this data center by using the primary zone identifier. If Data center 1 is faulty, the application in this data center is unavailable. The application system provides a disaster recovery solution for applications. This document does not provide details about the disaster recovery solution for applications. The discussions in this document assume that the involved applications ensure service availability and continuity.

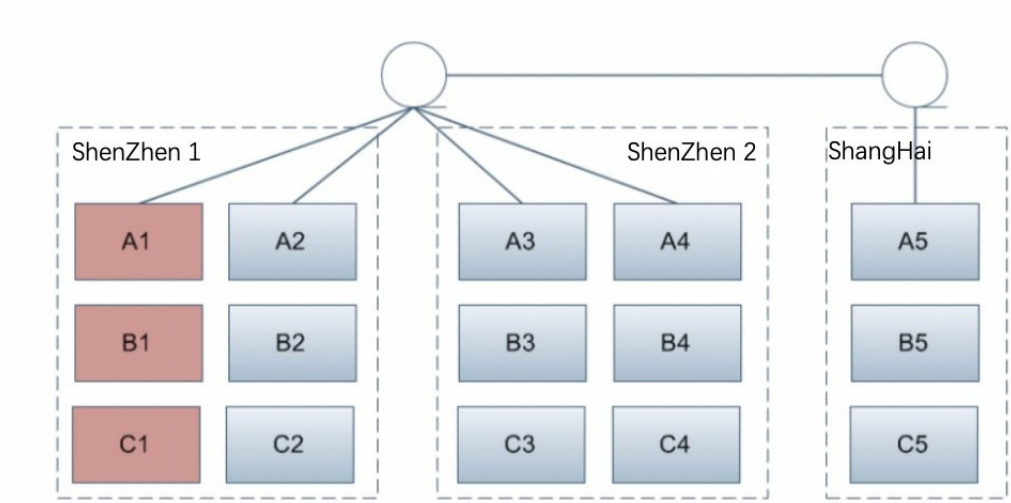
In the Three Data Centers in the Same Region architecture, changes in a leader partition must be synchronously replicated to at least a follower replica in another data center each time you commit a transaction. In most cases, the network latency of the replication process is about 2 milliseconds because the network latency in the same region is 2 milliseconds. For example, in core services of Alipay, a database transaction consists of 6 to 15 Structured Query Language (SQL) statements in most cases. About 15 milliseconds to 50 milliseconds is required to complete each transaction. In this scenario, an additional network latency of 2 milliseconds is acceptable.

15.5.3.3. Three Data Centers Across Two Regions

In some scenarios, only two data centers are available in a region to reduce the deployment costs of data centers and networks. A specific software architecture must be designed for these scenarios. This is because the designs of general software architectures must be applicable to most of the service scenarios if possible.

At least three data centers are required to support non-disruptive disaster recovery across data centers. Therefore, you must use a third data center in another region to deploy the Three Data Centers Across Two Regions solution.

Three Data Centers Across Two Regions



Assume that two data centers are deployed in Shenzhen and one data center is deployed in Shanghai, as shown in Three Data Centers Across Two Regions. Two replicas are deployed for each partition in each data center in Shenzhen, and one replica is deployed in the data center of Shanghai. This deployment architecture is known as the 2+2+1 architecture. In the 2+2+1 architecture, each partition has five replicas. In normal cases, data changes need to be written to only three of the replicas in Shenzhen in synchronous replication mode. In this scenario, the network latency increases by less than 2 milliseconds each time you commit a transaction.

When the data center in Shanghai is faulty, changes in the MemTable of a partition need to be written to only three of the replicas in Shenzhen in synchronous replication mode. This does not affect the system. If a data center, such as Data center 2 in Shenzhen, is faulty, the system has only three replicas. If you do not handle the fault, changes in the MemTable of a partition must be replicated to the data centers from Shenzhen to Shanghai each time you commit a transaction. This replication process causes long network latencies that do not meet your expectations. By using the Paxos protocol, ApsaraDB for OceanBase removes the two replicas of Data center 2 in Shenzhen from the Paxos election group. The total number of replicas changes from five to three. This way, you need to choose only two of the remaining three replicas and replicate data from one replica to the other replica in the same region. The Paxos protocol can be used to implement member changes. A member change takes effect only if the change is accepted by a majority of members before and after the change.

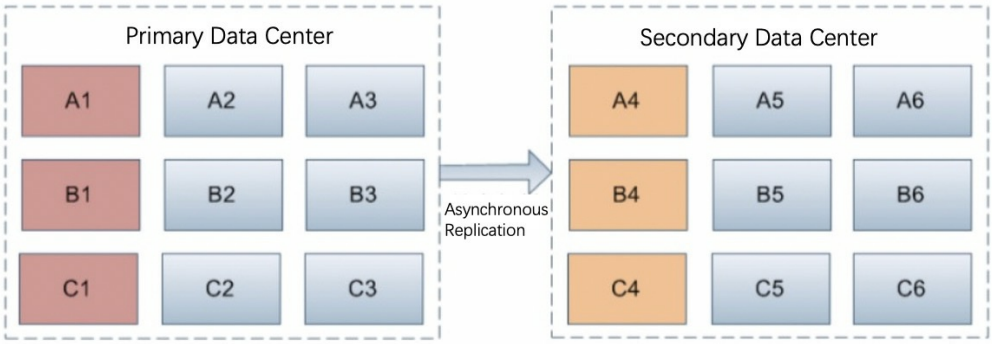
In most cases, traditional relational databases for banks also use the Three Data Centers Across Two Regions architecture. In this architecture, two data centers are deployed in the same region. One of the two data centers serves as the primary database and the other one serves as the secondary database for hot backups. The third data center in the other region serves as the secondary database for cold backups. When the primary database is faulty, the latest changes cannot be synchronized to the two secondary databases. To avoid data losses, you can only use the primary database to provide services after the primary database recovers. If a secondary database is forced to serve as the primary database, your service data is lost. For traditional relational databases, the Three Data Centers Across Two Regions architecture allows you to ensure only one of the following features: high availability and strong consistency. If you need to ensure high availability, you can minimize the recovery time objective (RTO), but the recovery point objective (RPO) is always greater than zero. If you need to ensure strong consistency, the RPO is zero, but you cannot control the RTO. In the 2+2+1 architecture of ApsaraDB for OceanBase, three data centers are also deployed in two regions. The highlight of this architecture is that ApsaraDB for OceanBase uses the Paxos protocol at the underlying layer to ensure non-disruptive disaster recovery across data centers.

15.5.3.4. Hot backups based on two data centers

In some scenarios, only two data centers are deployed for services. In theory, ApsaraDB for OceanBase cannot ensure non-disruptive disaster recovery across data centers in these scenarios.

To resolve this issue, ApsaraDB for OceanBase provides hot backups that are based on dual data centers for disaster recovery.

Hot backups based on two data centers



In Hot backups based on two data centers, one ApsaraDB for OceanBase cluster is deployed in the primary data center and another cluster is deployed in the secondary data center. The cluster in the primary data center is an active cluster and the cluster in the secondary data center is a standby cluster. In the active cluster, A1, B1, and C1 are leader partitions. In the standby cluster, A4, B4, and C4 are leader partitions. The changes in the active cluster are replicated to the standby cluster in an asynchronous way. If a single OBServer is faulty, the ApsaraDB for OceanBase system automatically switches services from the faulty server to another server and the services are not affected. If the entire active cluster is faulty, the standby cluster is forced to serve as the active cluster to ensure service continuity. During the switchover, the services are interrupted for a short period of time. This way, the hot backup solution that is based on two data centers supports non-disruptive disaster recovery across servers. However, the solution does not support non-disruptive disaster recovery across data centers.

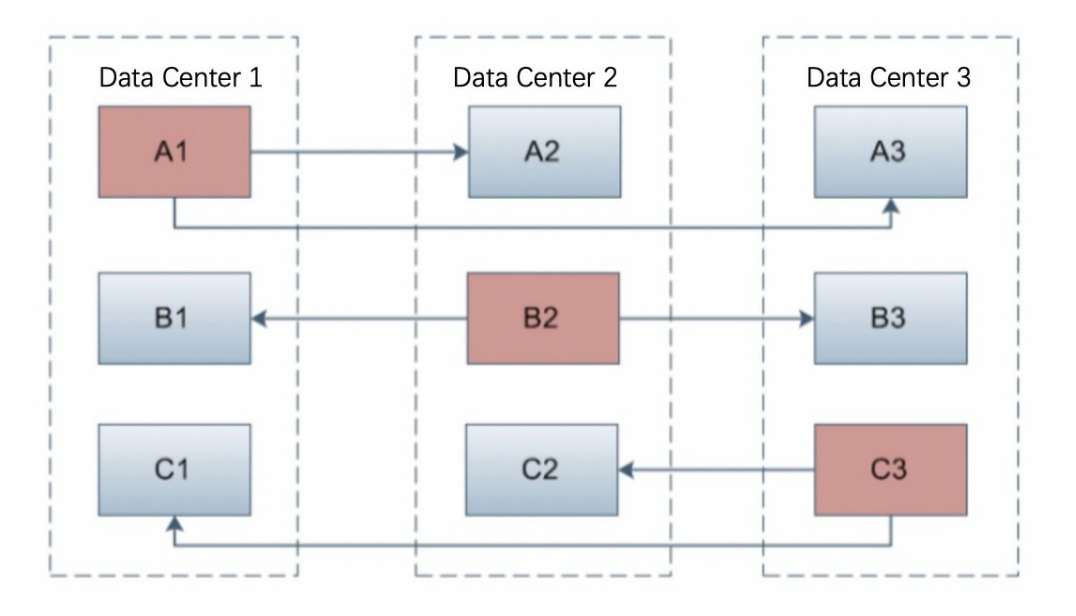
15.5.4. Deployment and costs

15.5.4.1. Mixed deployment

Assume that the ApsaraDB for OceanBase system is deployed in three data centers. Data center 1 is the primary data center. Data center 2 and Data center 3 are secondary data centers. If the secondary data centers are used only for disaster recovery and do not provide services, large amounts of resources are wasted. To avoid the resource waste, you can use the mixed deployment method.

In Mixed deployment, the A1 leader partition is deployed in Data center 1, the B2 leader partition is deployed in Data center 2, and the C3 leader partition is deployed in Data center 3. This maximizes the utilization of ApsaraDB for OceanBase server resources in the three data centers.

Mixed deployment



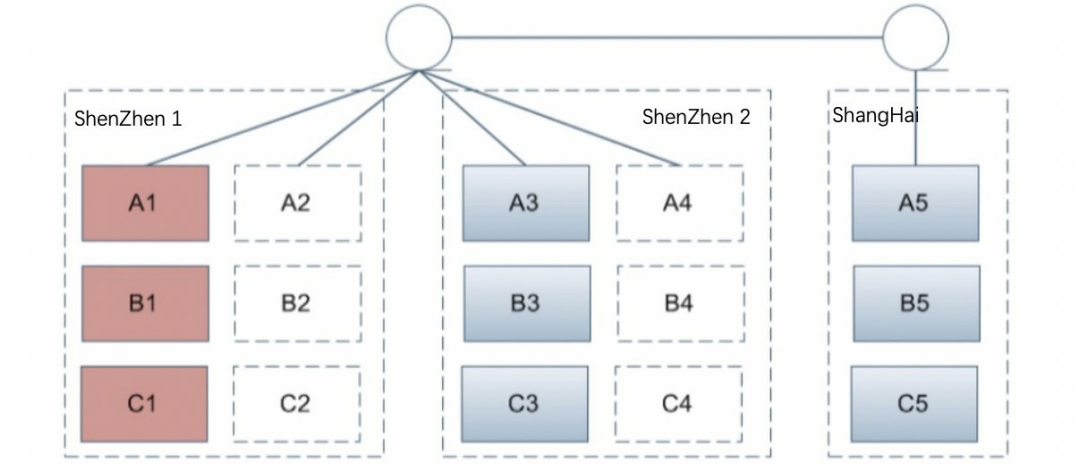
15.5.4.2. Log replicas

Logically, each replica must store complete SSTables and MemTables even if you use the mixed deployment method. In this method, leader partitions are deployed across data centers. Note that the SSTables occupy storage space and the MemTables occupy memory space. However, most of the replicas are used to ensure only high availability. Therefore, you need only to record redo logs for these replicas. In ApsaraDB for OceanBase, these replicas are known as log replicas.

For example, in the 2+2+1 architecture, you can use one replica in Shenzhen data center 1 and one replica in Shenzhen data center 2 as log replicas.

In Log replicas, the A2, A4, B2, B4, C2, and C4 replicas are log replicas. These replicas do not store SSTables and MemTables. Only logs are recorded for these replicas. You can use the mixed deployment method to deploy log replicas and non-log replicas across data centers. This ensures that the log replicas occupy only few additional server resources.

Log replicas



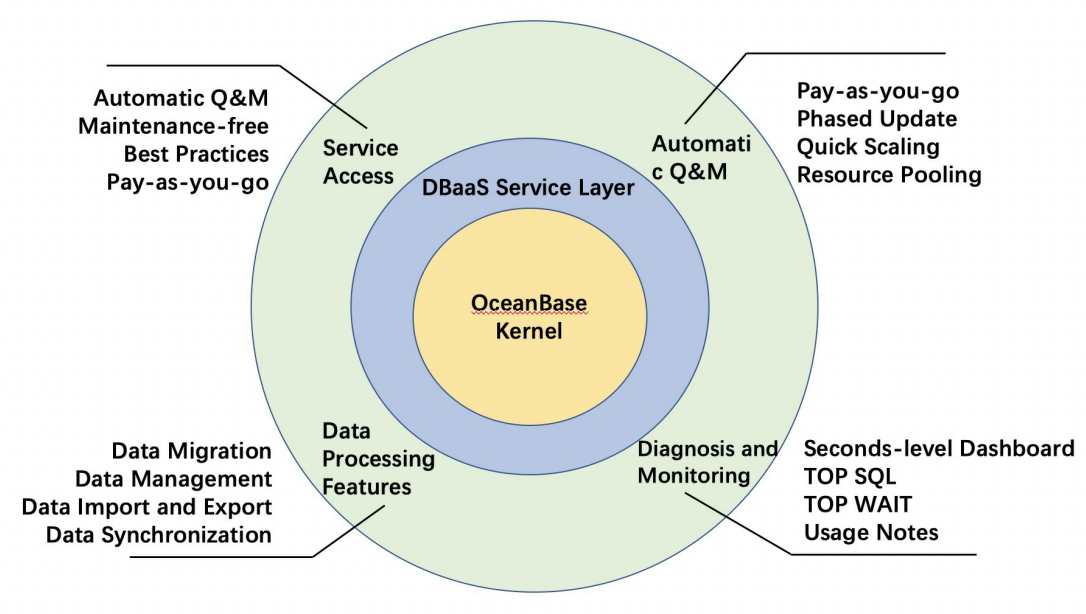
15.6. OCP

The ApsaraDB for OceanBase system serves as only the database kernel. To access and use the services of the system, you must log on to the OceanBase Cloud Platform (OCP), as shown in OCP. ApsaraDB for OceanBase and OCP provide database-as-a-service (DBaaS) capabilities.

In OCP, you can apply for tenant instances and specify limits for the following instance resources: CPU, memory, and input/output operations per second (IOPS). You can also migrate your MySQL business to ApsaraDB for OceanBase. To migrate your business, you can use the component that allows you to migrate data with a few clicks. In OCP, you can dynamically add or reduce tenant resources and back up and restore data.

OCP provides features such as diagnosis and monitoring, reporting of running information, intelligent alerting, and resource scheduling. OCP can also automatically detect SQL statement errors and provide recommendations to revise the SQL statements.

OCP



The service layer in the DBaaS model provides various API operations. This allows you to customize API operations based on your business processes.

16.Data Transmission Service (DTS)

16.1. What is DTS?

Data Transmission Service (DTS) is a data service that is provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases and big data systems. DTS provides the following data transmission methods: data migration, data synchronization, and change tracking. You can use DTS to achieve data migration with minimal downtime, geo-disaster recovery, cache updates, online and offline data synchronization, and asynchronous notifications.

DTS allows you to build a data architecture that features high availability, scalability, and security.

16.2. Benefits

Data Transmission Service (DTS) allows you to transfer data between various data sources, such as relational databases and online analytical processing (OLAP) databases. DTS provides the following data transmission methods: data migration, data synchronization, and change tracking. Compared with other data migration and synchronization tools, DTS provides tasks that have higher compatibility, performance, security, and reliability. DTS also provides a variety of features to help you create and manage tasks.

High compatibility

DTS allows you to migrate or synchronize data between homogeneous and heterogeneous data sources. For migration between heterogeneous data sources, DTS supports schema conversion.

DTS provides the following data transmission methods: data migration, data synchronization, and change tracking. In change tracking and data synchronization, data is transferred in real time.

DTS minimizes the impact of data migration on applications to ensure service continuity. The application downtime during data migration is minimized to several seconds.

High performance

DTS uses high-end servers to ensure the performance of each data synchronization or migration task.

DTS uses a variety of optimization measures for data migration.

Compared with traditional data synchronization, the data synchronization feature of DTS refines the granularity of concurrency to the transaction level. The feature allows you to synchronize incremental data in one table by using multiple concurrent tasks. This improves synchronization performance.

Security and reliability

DTS is implemented based on clusters. If a node in a cluster is unavailable or faulty, the control center switches all tasks on this node to another node in the cluster.

Secure transmission protocols and tokens are used for authentication across DTS modules to ensure reliable data transmission.

Ease of use

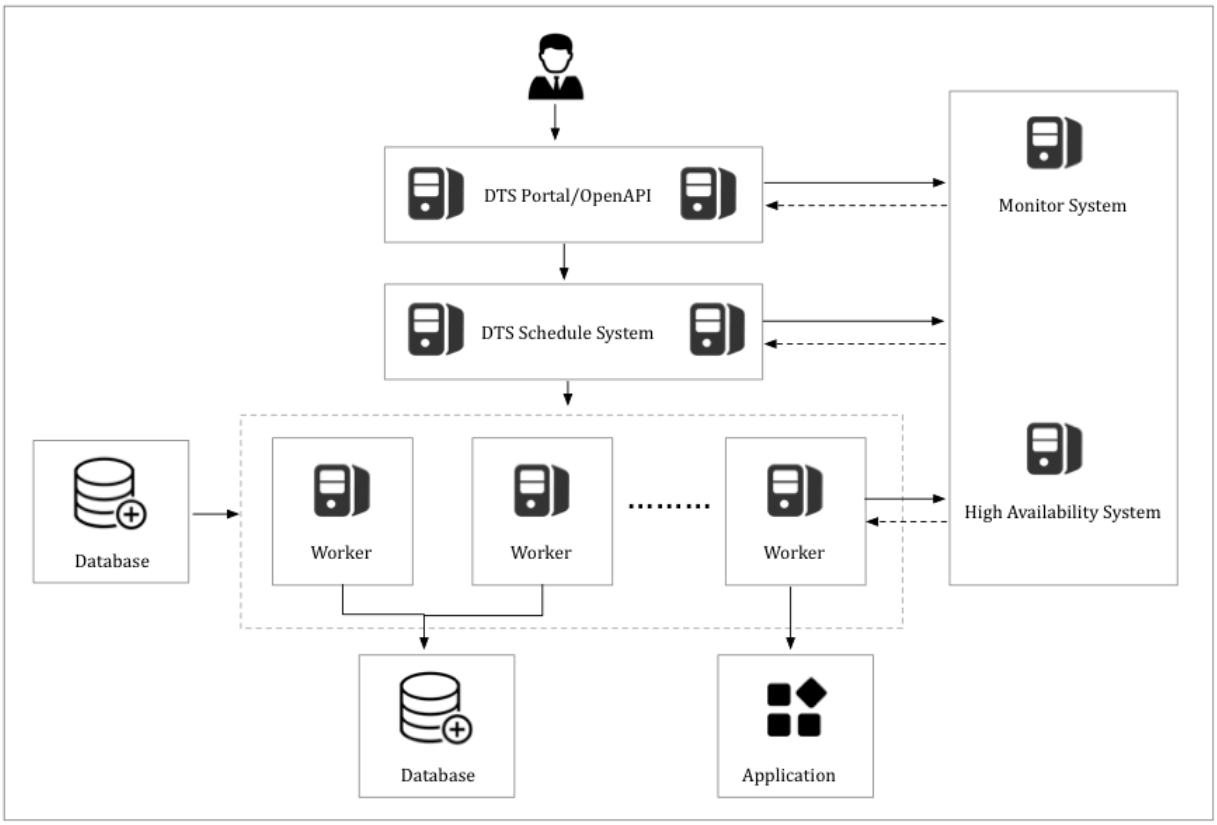
The DTS console provides a codeless wizard for you to create and manage tasks.

To facilitate task management, the DTS console shows task information such as status, progress, and performance.

DTS supports resumable transmission, and monitors task status on a regular basis. If DTS detects a network failure or system error, DTS automatically fixes the failure or error and restarts the task. If the failure or error persists, you must manually repair and restart the task in the DTS console.

16.3. Architecture

The following figure shows the system architecture of DTS.



16.4. Environment requirements

Use Data Transmission Service (DTS) on hosts of the following models:

- PF51.*
- PV52P2M1.*
- DTS_E.*
- PF61.*
- PF61P1.*
- PV62P2M1.*
- PV52P1.*
- Q5F53M1.*
- PF52M2.*
- Q41.*
- Q5N1.22
- Q5N1.2B
- Q46.22

- Q46.2B
- W41.22
- W41.2B
- W1.22
- W1.2B
- W1.2C
- D13.12

Use the following operating system:

AliOS7U2-x86-64

> **Notice**
>
> - Do not use DTS on hosts whose models are excluded from the preceding list.
> - The */apsara* directory used by DTS resides on only one hard disk. Make sure that the space of the hard disk is larger than 2 TB.
>
> If the space of the hard disk where the */apsara* directory resides is smaller than 2 TB, tasks may fail to run and errors may occur. In this case, DTS cannot restore failed tasks or pull data properly.

16.5. Features

16.5.1. Data migration

16.5.1.1. Data migration

Data migration allows you to quickly migrate data between multiple data sources. Typical scenarios include data migration to the cloud, data migration between instances within Alibaba Cloud, and database split and scale-out. DTS supports data migration between homogeneous and heterogeneous data sources. It also supports ETL features such as data mapping at database, table, and column levels and data filtering.

16.5.1.2. Data sources

The following table lists the data sources and data migration types that are supported by DTS.

Source database	Destination database	Migration type
	User-created MySQL database Versions 5.1, 5.5, 5.6, and 5.7	• Schema migration • Full data migration • Incremental data migration

Source database	Destination database	Migration type
• User-created MySQL database Versions 5.1, 5.5, 5.6, and 5.7 • RDS MySQL All versions	RDS MySQL Versions 5.6 and 5.7	• Schema migration • Full data migration • Incremental data migration
	Cloud Native Distributed Database PolarDB-X (formerly known as DRDS) All versions	• Full data migration • Incremental data migration
	User-created Oracle database (RAC or non-RAC architecture) Versions 9i, 10g, 11g, and 12c	• Full data migration • Incremental data migration
User-created SQL Server database Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016 **Note** • DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAG). • If the version of the source database is 2005, incremental data migration is not supported.	• User-created SQL Server database Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016 **Note** DTS does not support SQL Server clusters or SQL Server Always On availability groups (AOAG). • RDS SQL Server Versions 2005, 2008, 2008 R2, 2012, 2014, and 2016	• Schema migration • Full data migration • Incremental data migration
	User-created Oracle database (RAC or non-RAC architecture) Versions 9i, 10g, 11g, and 12c	• Schema migration • Full data migration • Incremental data migration

Source database	Destination database	Migration type
User-created Oracle database (RAC or non-RAC architecture) Versions 9i, 10g, 11g, and 12c	PolarDB (formerly known as ApsaraDB RDS for PPAS) Version 11	• Schema migration • Full data migration • Incremental data migration
	User-created MySQL database Versions 5.1, 5.5, 5.6, and 5.7	• Schema migration • Full data migration • Incremental data migration
	RDS MySQL Versions 5.6 and 5.7	• Schema migration • Full data migration • Incremental data migration
	Cloud Native Distributed Database PolarDB-X All versions	• Full data migration • Incremental data migration
	AnalyticDB for MySQL Versions 2.0 and 3.0	• Schema migration • Full data migration • Incremental data migration
• User-created PostgreSQL database Versions 9.4, 9.5, 9.6, and 10.x • RDS PostgreSQL Versions 9.4 and 10	• User-created PostgreSQL database Versions 9.4, 9.5, 9.6, and 10.x • RDS PostgreSQL Versions 9.4 and 10	• Schema migration • Full data migration • Incremental data migration

16.5.1.3. Online migration

Data migration in DTS refers to online data migration that is an automatic process. You need only to specify the source instance, destination instance, and objects for migration. Online data migration supports migration with zero downtime. You must make sure that the DTS server has connections to the source and destination instances at the same time.

16.5.1.4. Migration modes

Data migration supports schema migration, full migration, and incremental migration. Descriptions of these migration modes are as follows:

- Schema migration: migrates the schema definitions from the source instance to the destination instance.
- Full migration: migrates historical data from the source instance to the destination instance.
- Incremental migration: migrates incremental data generated during migration from the source instance to the destination instance in real time. You can combine these modes to perform business migration with zero downtime.

16.5.1.5. ETL features

Data migration supports the following ETL features:

- Object name mappings at database, table, and column levels. Object name mappings are used for data migration between objects on the source and destination instances. The objects have different database, table, or column names.
- Data filtering is supported for migration. You can configure a standard SQL filtering criteria to filter the table to be migrated. For example, you can specify the time range to migrate the latest data only.

16.5.1.6. Migration task

Migration task is the basic unit of data migration. To migrate data, you must first create a data migration task in the DTS console. To create a data migration task, you must configure information such as the source instance connection type, destination instance connection type, migration type, and objects you want to transfer. You can create, manage, stop, and delete data migration tasks in the DTS console.

16.5.2. Data synchronization

16.5.2.1. Overview

You can use Data Transmission Service (DTS) to synchronize data between two data sources in real time. This feature applies to multiple scenarios, such as zone-disaster recovery, geo-disaster recovery, and data synchronization between OLTP and OLAP databases.

Supported databases

Source database	Destination database	Initial synchronization type	Synchronization topology
• User-created MySQL database 5.1, 5.5, 5.6, and 5.7 • RDS MySQL 5.6 and 5.7	User-created MySQL database 5.1, 5.5, 5.6, and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	RDS MySQL 5.6 and 5.7	Initial schema synchronization Initial full data synchronization	One-way synchronization Two-way synchronization
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
	AnalyticDB for PostgreSQL 4.3 and 6.0	Initial schema synchronization Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization
	MaxCompute	Initial schema synchronization Initial full data synchronization	One-way synchronization
Cloud Native Distributed Database PolarDB-X (formerly	Cloud Native Distributed Database PolarDB-X	Initial full data synchronization	One-way synchronization
	Datahub	Initial schema synchronization	One-way synchronization

known as DRDS) Source database	Destination database	Initial synchronization type	Synchronization topology
	AnalyticDB for MySQL 2.0 and 3.0	Initial schema synchronization Initial full data synchronization	One-way synchronization

16.5.2.2. Synchronization tasks

Synchronization tasks are the basic units for real-time data synchronization. To synchronize data between two instances, you must create a synchronization task in the DTS console.

Synchronization task statuses and descriptions shows the statuses of a synchronization task during creation and running.

Synchronization task statuses and descriptions

Status	Description	Available operation
Pre-checking	The synchronization task is performing a pre-check before the task is started.	• View synchronization configurations. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.
Pre-check failed	The synchronization task has failed the pre-check.	• Perform the pre-check. • View synchronization configurations. • Modify synchronization objects. • Modify synchronization speed. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.

Status	Description	Available operation
Not started	The synchronization task that has passed the pre-check is not started.	• Perform the pre-check. • Start the synchronization task. • Modify synchronization objects. • Modify synchronization speed. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.
Initializing	The synchronization task is being initialized.	• View synchronization configurations. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.
Initialization failed	Data migration has failed during the synchronization task initialization.	• View synchronization configurations. • Modify synchronization objects. • Modify synchronization speed. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.

Status	Description	Available operation
Synchronizing	The task is synchronizing data.	• View synchronization configurations. • Modify synchronization objects. • Modify synchronization speed. • Pause the synchronization task. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.
Synchronization failed	A synchronization exception occurred.	• View synchronization configurations. • Modify synchronization objects. • Modify synchronization speed. • Start the synchronization task. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.
Paused	The synchronization task is paused.	• View synchronization configurations. • Modify synchronization objects. • Modify synchronization speed. • Start the synchronization task. • Delete the synchronization task. • Replicate synchronization configurations. • Configure monitors and alarms.

16.5.2.3. Synchronization objects

- Data synchronization objects include databases, tables, and columns. You can specify the tables that you want to synchronize.
- Data synchronization supports the mapping of database, table, and column names. In other words, objects can have different databases, tables, and column names during data synchronization.
- You can also synchronize specified columns of data in a table.

16.5.2.4. Advanced features

The following advanced features are used to facilitate data synchronization:

- Dynamically add and remove synchronization objects

  You can add and remove synchronization objects during data synchronization.

- Improve the performance query system

  Data synchronization provides the synchronization latency and performance trend chart (RPS and traffic). You can use this to easily view the performance of synchronization links.

16.5.3. Data subscription

16.5.3.1. Overview

You can use Data Transmission Service (DTS) to track data changes from RDS or DRDS instances in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, synchronization of heterogeneous data, and synchronization of extract, transform, and load (ETL) operations.

Supported databases

- User-created MySQL database or ApsaraDB RDS for MySQL instance
- Cloud Native Distributed Database PolarDB-X (formerly known as DRDS)
- User-created Oracle database

16.5.3.2. Subscription channels and objects

Subscription channels

Subscription channels are the basic units of incremental data subscription and consumption. To subscribe to RDS incremental data, you must create a subscription channel in the DTS console for the relevant RDS instance. The subscription channel reads RDS incremental data in real time and stores the most recent increments. You can use the SDK provided by DTS to subscribe to and consume the incremental data in the channel. You can create, manage, and delete subscription channels in the DTS console.

A subscription channel can only be subscribed and consumed by one downstream SDK. To subscribe to an RDS instance for multiple downstream SDKs, you must create an equivalent number of subscription channels. RDS instances subscribed to with these subscription channels share the same instance ID.

Subscription channel statuses and descriptions shows the statuses of a subscription channel during creation and running.

Subscription channel statuses and descriptions

Status	Description	Available operation
Pre-checking	The subscription channel has completed task configurations and is performing a pre-check.	Delete the subscription channel.
Not started	The migration task has passed the pre-check, but is not started.	• Start subscription. • Delete the subscription channel.
Initializing	The subscription channel is being initialized. This process takes about one minute.	Delete the subscription channel.
Normal	The subscription channel is reading incremental data from an RDS instance.	• View sample code. • View the subscribed data. • Delete the subscription channel.
Abnormal	An exception occurs when the subscription channel reads incremental data from an RDS instance.	• View sample code. • Delete the subscription channel.

Subscription objects

Subscription objects contain databases and tables. You can specify the tables that you want to subscribe to.

Incremental data is divided into data update and schema update in data subscription. You can select the specific data type when you configure data subscription.

16.5.3.3. Advanced features

The following advanced features are used to facilitate data subscription:

- Dynamically add and remove subscription objects

  You can add and remove subscription objects during data subscription.

- View the subscribed data online

  You can view the incremental data that has been subscribed to in the DTS console.

- Modify data consumption time

You can modify the time for data consumption at any time.

17.Data Management (DMS)

17.1. What is Data Management?

Data Management (DMS) centrally manages relational databases and online analytical processing (OLAP) databases. It is built on the iDB database service platform of Alibaba and has been providing database R&D support for tens of thousands of R&D engineers since it was launched. You can use DMS to build your own database DevOps, which improves database R&D efficiency by providing better self-service and ensures secure database access and high database performance.

DMS is used to manage relational databases such as MySQL, SQL Server, Cloud Native Distributed Database PolarDB-X, and PostgreSQL. It integrates data management with schema management.

17.1.1. Product value

DMS provides you with a convenient and secure database access and management platform. Visualized data services enable you to use databases on browsers, eliminating the need to install various database clients. When you edit data online, you can easily perform operations on table data and change table structures, without having to write complex SQL statements. DMS provides advanced functions that common clients do not offer, such as table structure synchronization, database clone, chart-based presentation of result sets, and real-time monitoring.

To use DMS, you must first log on to the Apsara Stack console, and then use your database account and password to log on to the DMS console. This feature adds an extra layer of security to your database account. DMS supports HTTPS and SSL for data transmission, and prevents data from being intercepted or tampered with during transmission.

DMS also supports RAM and STS for permission verification to prevent unauthorized actions.

DMS supports VPC instance access and provides data access interfaces for users while ensuring security of the database instance network, which is beyond the capability of common clients.

DMS provides the following benefits:

- Simple data operations
  - Pain point: You need a convenient and all-in-one product to complete SQL operations, save common operations, and apply common operations to specific services.
  - Solution: You can create a table in DMS and perform operations on table data just as you would in an Excel worksheet. You can add, delete, change, query, and make statistical analysis of table data without understanding SQL. You can customize SQL operations and save common business-related SQL operations in DMS. Then you can apply these operations directly when managing other databases or instances.
- Visualization of database table structures
  - Pain point: When you design a new business table or perform operations on an existing business table, you often need to understand the structures of all tables in a database. You can execute SQL commands one by one to display the table structures, but this method is neither intuitive nor convenient.
  - Solution: Through the document generation function of DMS, you can generate the table structures of an entire database with a single click. Then you can browse these structures online or export them to other formats such as Word, Excel, and PDF.
- Real-time optimization of database performance

  - Pain point: Detailed monitoring logs over a long period of time are required for database performance optimization. You need to make a detailed analysis of the logs and locate exceptions to better improve the database performance.
  - Solution: DMS provides second-level monitoring of database performance metrics, such as SELECT, INSERT, UPDATE, and DELETE operations, the number of active connections, and network traffic volume, and helps keep you informed of any performance variations. DMS allows you to view and terminate database sessions.

- Chart-based presentation of SQL result sets
  - Pain point: Users used to use SQL statements to find data, and import the data into Excel to create static charts such as line charts and pie charts. This process takes a lot of time.
  - Solution: With DMS, you can directly create charts from SQL result sets. You can also create many advanced charts, such as dynamic charts, period-over-period comparison charts, and personalized tooltips. This helps you produce high-quality work.

- SQL statement reuse
  - Pain point: When you access a database, there is always a need to execute SQL statements. Simple queries are easy to master, while complex analytical queries or queries with certain business logics are not. The cost of rewriting SQL statements each time is too high, and even if the statements are saved to text files, they require constant maintenance and cannot be used flexibly.
  - Solution: You can use the My SQL function provided by DMS to save frequently used SQL statements. As the SQL statements are not saved locally, they can be reused in any databases or instances.

- Monitoring of changes to the table data volume

  Big data is the latest trend in data analysis and is widely discussed. However, taking full advantage of the values provided by big data analysis is not an easy task. The core idea of DMS is to start analyzing data when data is available.

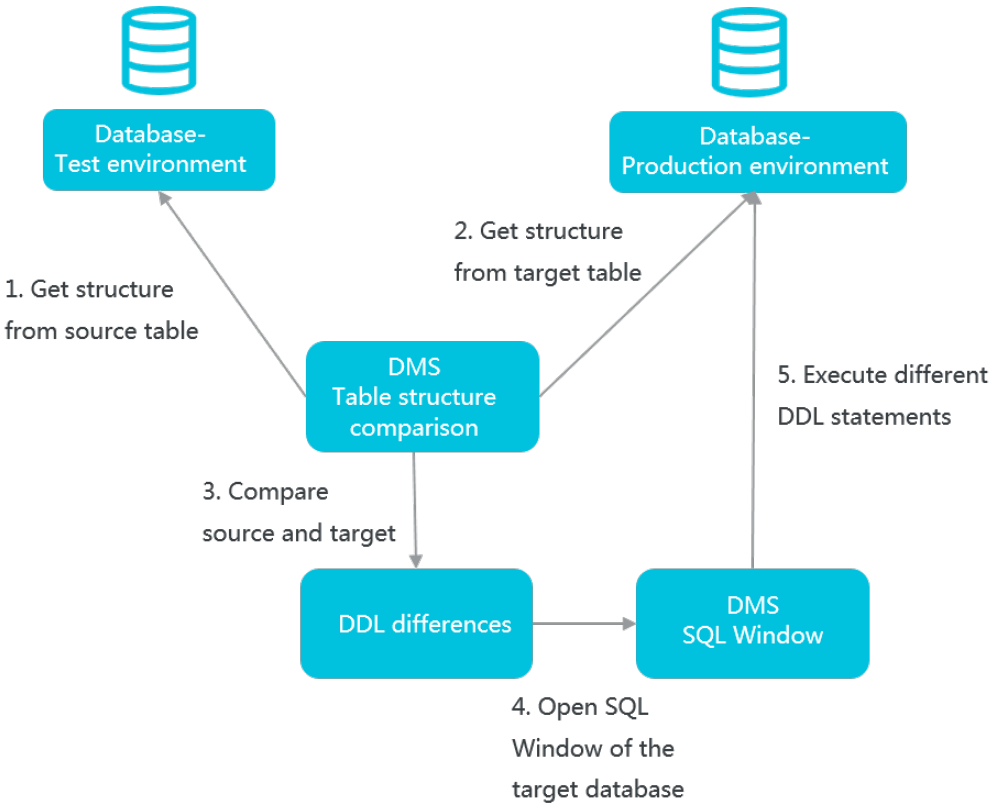
  DMS monitors changes to table data volume through a custom RDS kernel, which allows it to quickly collect row count changes of each instance, database, and table. DMS provides real-time monitoring, data trends, and detailed data through professional data analysis and interaction.

- Table structure synchronization
  - Pain point: Within enterprises, database environments are divided into production environment and test environment. A database will be released in the production environment after it is verified in the test environment. If table structures in the test environment are not completely synchronized to the production environment, major faults can occur during the release.

- Solution: You can use the table structural comparison function of DMS to detect inconsistencies in database table structures between the production and test environments. You can also obtain a DDL statement for table structure correction to ensure table structure consistency between the production and test environments.

  DMS - Table structure comparison - Table structure synchronization shows how to use the table structure comparison function to synchronize table structures.

  DMS - Table structure comparison - Table structure synchronization



17.2. Benefits

Improved R&D efficiency

- Schema comparison
- Smart SQL completion
- Convenient reuse of custom SQL statements and SQL templates
- Automatic restoration of work environments
- Export of dictionary files

Real-time optimization of database performance

- Effective session management
- Monitoring of core metrics in seconds
- Graphical lock management
- Real-time SQL index recommendations
- Reports on overall performance

Comprehensive access security protection

- Four-layer authentication system
- Fine-grained authorization
- Logon and operation audit

Extensive options for data sources

- ApsaraDB RDS for MySQL
- ApsaraDB RDS for PostgreSQL and Apsara PolarDB

17.3. Architecture

DMS consists of the business layer, scheduling layer, and connection layer. DMS processes real-time data access and schedules data-related background tasks for relational databases.

Business layer

- The business layer supports online GUI-based database operations and can be scaled to improve the general service capabilities of DMS.
- DMS supports stateless failover to ensure 24/7 availability.

Scheduling layer

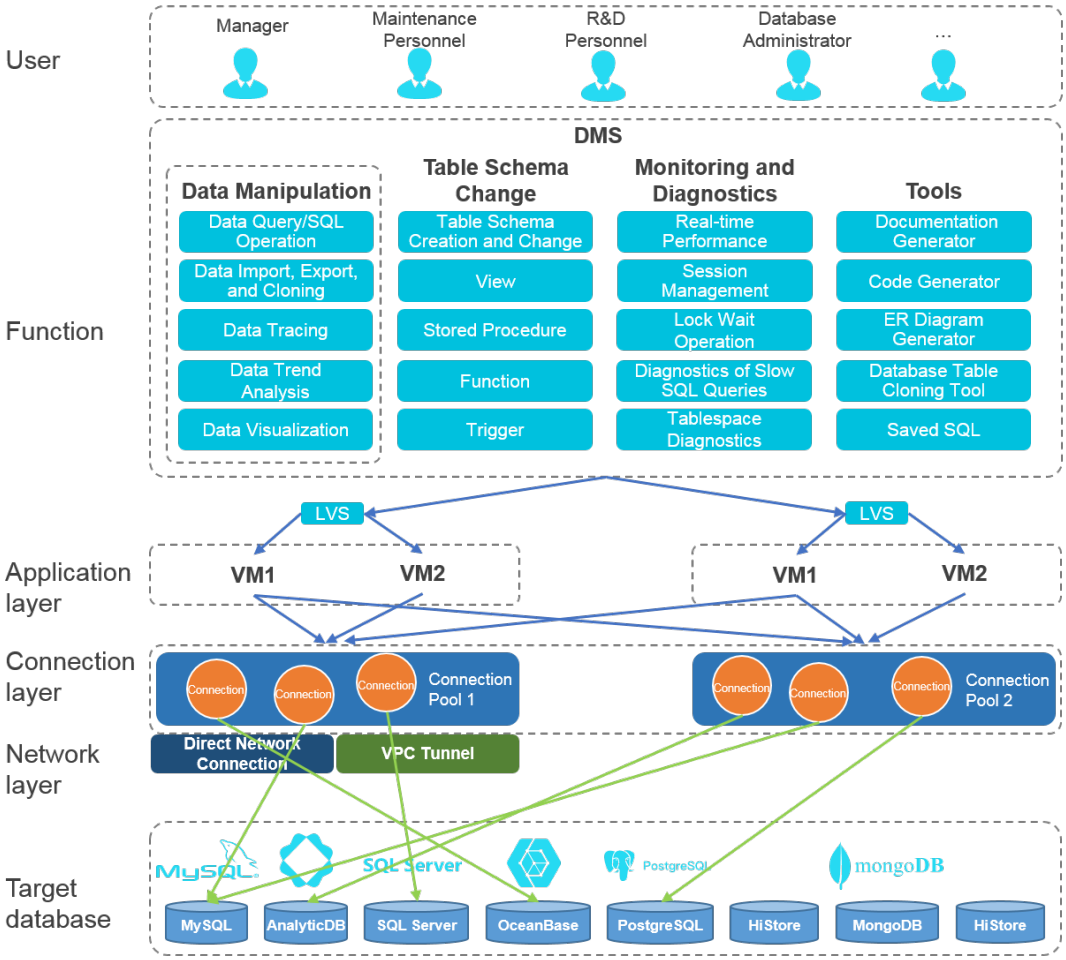
- The scheduling layer allows you to import and export tables and compare schemas. This layer schedules tasks by using the thread pool in the real-time scheduling or background periodic scheduling mode.
- Real-time scheduling allows you to schedule and run tasks on the frontend. After you submit a task, DMS automatically runs the task in the background. After the task is completed, you can download or view the execution result.
- Background periodic scheduling allows you to periodically obtain specified data such as data trends. DMS collects business data in the background for your reference and analysis based on scheduled tasks.

Connection layer

The connection layer is the core component for accessing data in DMS. It has the following characteristics:

- Processes requests from MySQL, SQL Server, and PostgreSQL databases.
- Supports session isolation and persistence. SQL windows opened in DMS are isolated from each other and the sessions in each SQL window are persistent to simulate the client experience.
- Controls the number of instance sessions to prevent a large number of connections from being established to a single instance.
- Provides different connection release policies for different features. This improves user experience and reduces the number of connections to the databases.

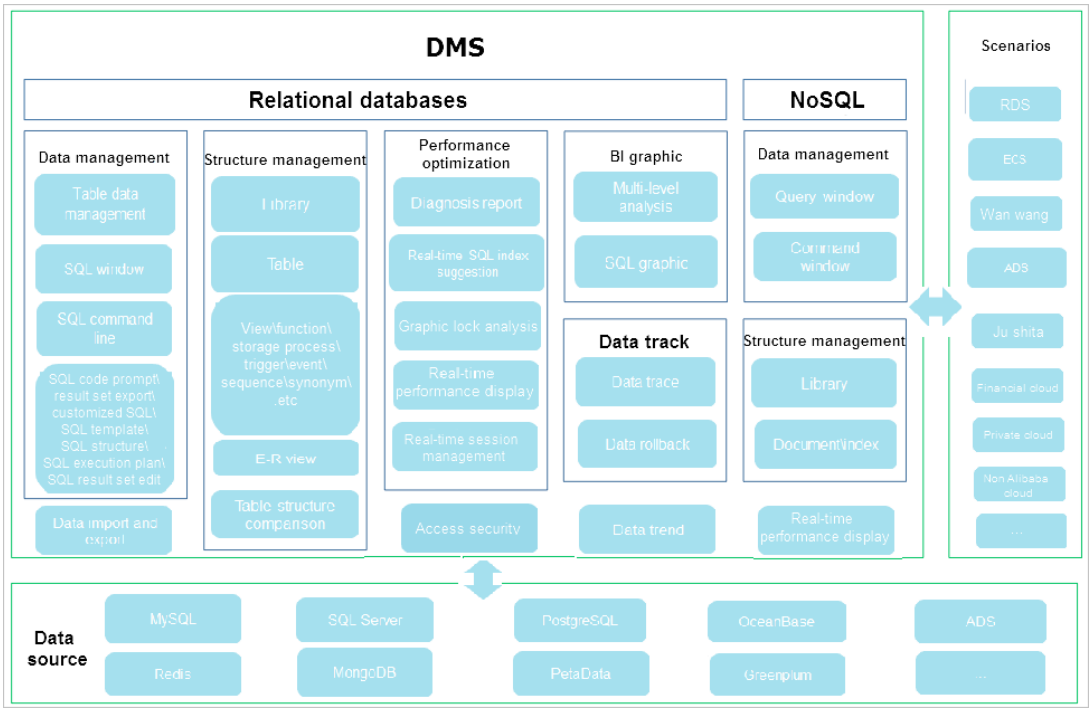
DMS system architecture



17.4. Features

The following figure shows the features of DMS.

DMS features



- **Features for relational databases**
  - Data management: SQL editor, SQL command-line interface, table management, smart SQL completion, SQL formatting, custom SQL queries, SQL templates, SQL execution plans, and import and export.
  - Schema management: schema comparison and management for objects such as databases, tables, views, functions, storage procedures, triggers, events, series, and synonyms.
  - Performance optimization: real-time performance monitoring, real-time SQL index recommendation, graphical interface for lock management, session management, and diagnostic reporting.
  - Access control: four-layer authentication, logon and operation auditing, and fine-grained authorization at the Apsara Stack tenant account, access address, and feature levels.
- **Features for NoSQL databases**
  - Data management: query editor and command-line interface.
  - Schema management: management of objects such as databases, documents, and indexes.
  - Real-time performance monitoring: real-time display of key performance indicators.

18.Server Load Balancer (SLB)

18.1. What is SLB?

This topic provides an overview of Server Load Balancer (SLB). SLB distributes inbound network traffic across multiple Elastic Compute Service (ECS) instances that act as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

SLB consists of three components:

- SLB instances

  An SLB instance is a key load-balancing component in SLB. It receives traffic and distributes traffic to backend servers. To get started with SLB, you must create an SLB instance and add at least one listener and two ECS instances to the SLB instance.

- Listeners

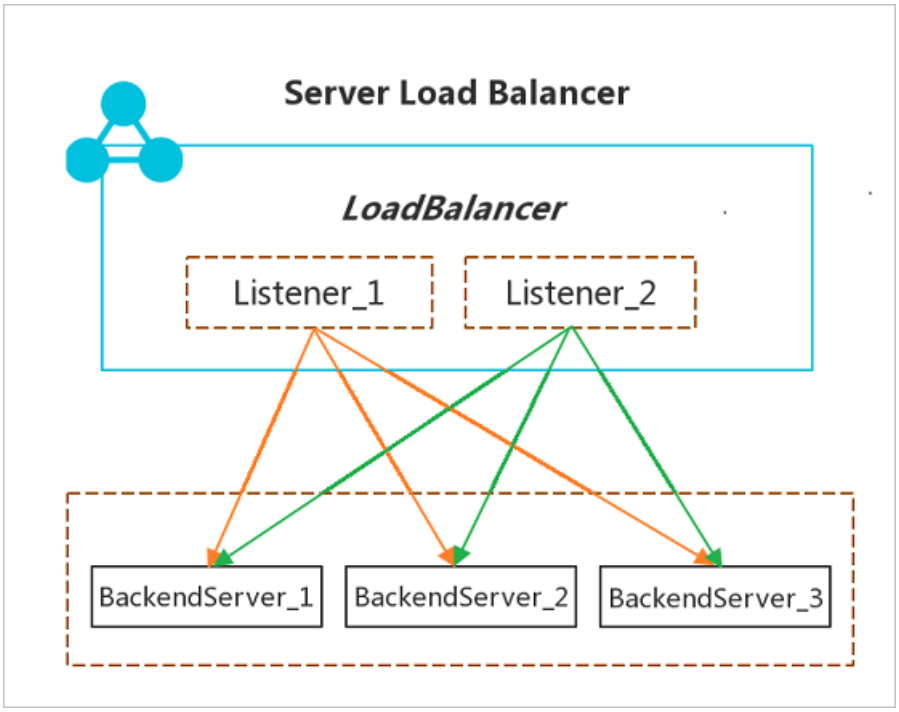
  A listener checks for connection requests from clients, forwards requests to backend servers, and performs health checks on backend servers.

  You can create listeners for Layer-4 (TCP and UDP) or Layer-7 (HTTP and HTTPS) load balancing. For Layer-7 listeners, you can create domain- and URL- based forwarding rules.

- Backend servers

  ECS instances are used as backend servers in SLB to receive and process distributed requests. You can create server groups to categorize your ECS instances in different ways, for example, by use case or by application.

  After an SLB instance receives client requests, the listeners of the SLB instance forward the requests to corresponding backend ECS instances based on the configured forwarding rules, as shown in the following figure.



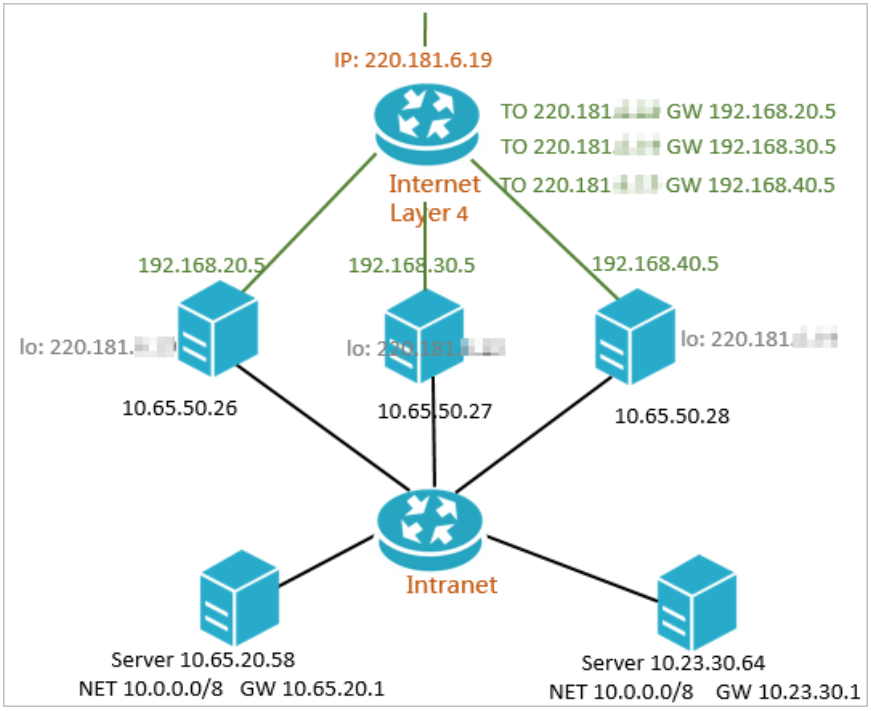
18.2. Architecture

This topic describes the SLB architecture. SLB instances are deployed in clusters to synchronize sessions and protect backend servers from single points of failures (SPOFs), improving redundancy and ensuring service stability.

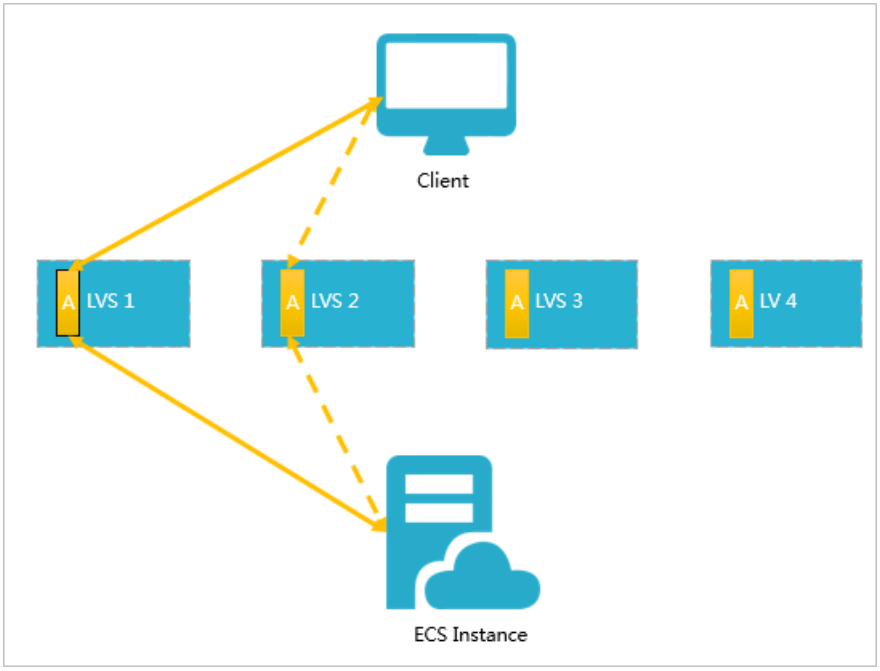
Apsara Stack provides Layer-4 (TCP and UDP) and Layer-7 (HTTP and HTTPS) load-balancing services.

- Layer-4 SLB combines the open-source Linux Virtual Server (LVS) with Keepalived to balance loads, and implements customized optimizations to meet cloud computing requirements.
- Layer-7 SLB uses Tengine to balance loads. Tengine is a web server project launched by Taobao. Based on NGINX, Tengine has a wide range of advanced features optimized for high-traffic websites.

Layer-4 SLB runs in a cluster of LVS machines, as shown in the following figure. This cluster deployment model strengthens the availability, stability, and scalability of the load balancing service in abnormal cases.



In an LVS cluster, each machine uses multicast packets to synchronize sessions with the other machines. Session A established on LVS1 is synchronized to other LVS machines after the client transfers three data packets to the server, as shown in the following figure. Solid lines indicate the current active connections, while dotted lines indicate that the session requests will be sent to other normally working machines if LVS1 fails or is being maintained. In this way, you can perform hot updates, machine maintenance, and cluster maintenance without affecting business applications.



18.3. Function principles

This topic describes the working principles of SLB. SLB distributes inbound network traffic across multiple ECS instances that act as backend servers based on forwarding rules. You can use SLB to improve the responsiveness and availability of your applications.

After you add ECS instances to an SLB instance, SLB uses virtual IP addresses (VIPs) to virtualize the ECS instances into backend servers in a high-performance server pool that ensures high availability. Client requests are distributed to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically removes unhealthy ones from the server pool to eliminate SPOFs. This enhances the resilience of your applications. You can also use SLB to defend your applications against distributed denial of service (DDoS) attacks

18.4. Benefits

18.4.1. LVS in Layer-4 SLB

This topic describes the customized technical improvements on LVS.

Drawbacks of LVS

LVS is an open-source project established by Dr. Zhang Wensong in May 1998. It is now the world's most popular Layer-4 load-balancing software for Linux kernel-based operating systems. LVS is implemented as a kernel module named IP Virtual Server (IPVS) in the netfilter framework, which is similar to iptables. LVS is hooked into LOCAL_IN and FORWARD.

In a large-scale cloud computing network, LVS has the following drawbacks:

- Drawback 1: LVS supports three packet forwarding modes: NAT, DR, and TUNNEL. When these forwarding modes are deployed in a network with multiple VLANs, the network topology becomes complex and incurs high O&M costs.
- Drawback 2: Compared with commercial load-balancing devices such as F5, LVS lacks defense against

DDoS attacks.

- Drawback 3: LVS uses PC servers and the Virtual Router Redundancy Protocol (VRRP) of Keepalived to deploy primary and secondary nodes for high availability. Therefore, its performance cannot be extended.
- Drawback 4: The configurations and health check performance of the Keepalived program are insufficient.

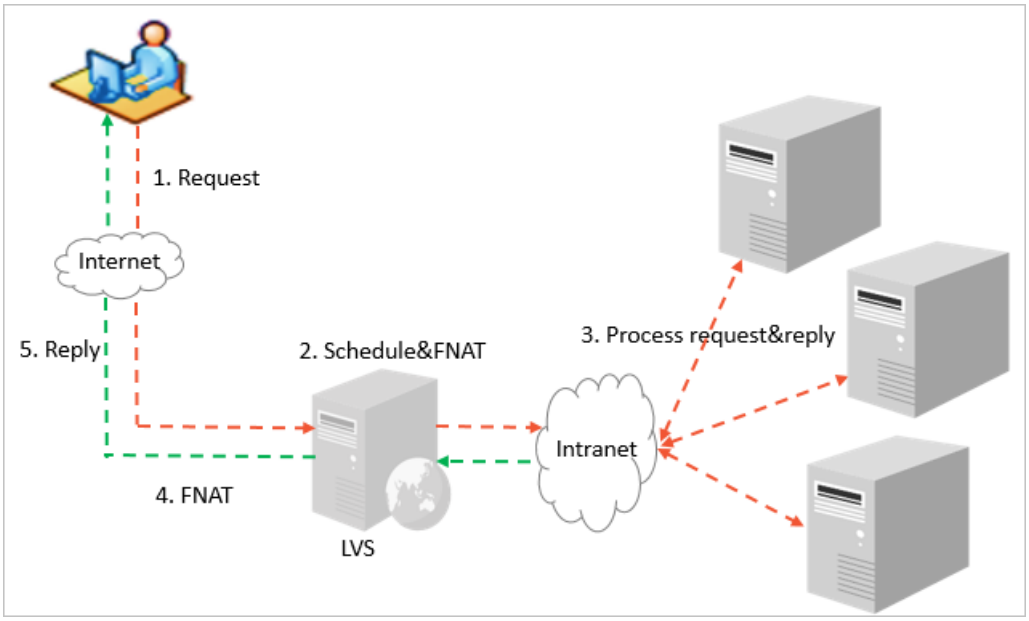
LVS customized features

To solve these problems, Alibaba Cloud added the following customized features to LVS. For more information about Ali-LVS, visit https://github.com/alibaba/LVS.

- Customization 1: FULLNAT, a new forwarding mode that enables inter-VLAN communication between LVS load balancers and backend servers.
- Customization 2: Defense modules such as SYNPROXY against TCP flag-targeted DDoS attacks.
- Customization 3: Support for LVS cluster deployment.
- Customization 4: Improved Keepalived performance.

FULLNAT technology

- Principles: The module introduces local IP addresses (internal IP addresses). IPVS translates CIP (client IP address)-VIP to LIP (local IP address)-RIP (real IP address), in which both LIP and RIP are internal IP addresses. This means that the load balancers and backend servers can communicate across VLANs.
- All inbound and outbound data flows traverse LVS. 10-GE Network Interface Cards (NICs) are used to ensure adequate bandwidth.
- FULLNAT supports only TCP.

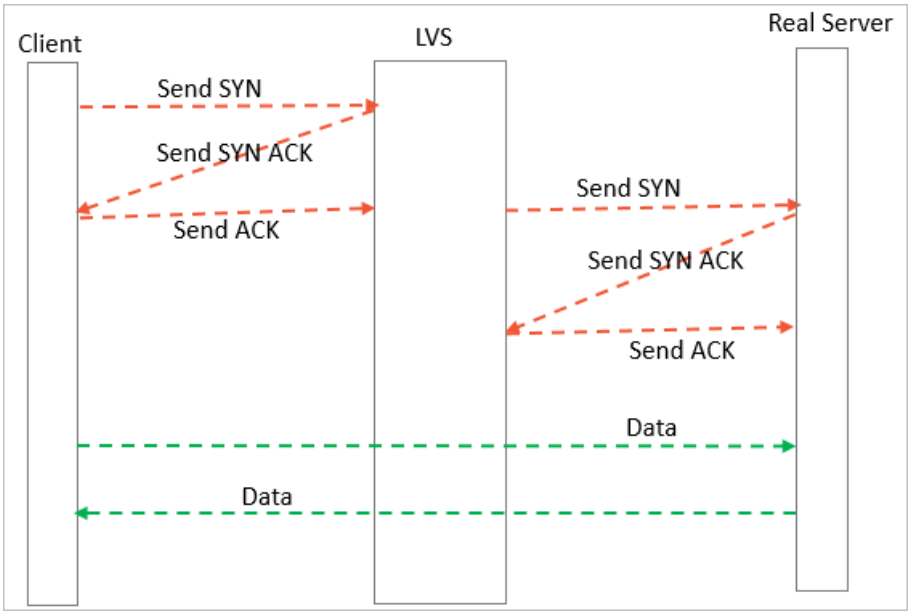


SYNPROXY technology

LVS uses the SYNPROXY module to defend against TCP flag-targeted attacks and SYN flood attacks. Based on the principle of SYN cookies in the Linux TCP protocol stack, LVS acts as a proxy for TCP three-way handshakes.

The process consists of the following steps:

1. A client sends an SYN packet to LVS.
2. LVS constructs an SYN-ACK packet with a unique sequence number and sends this packet to the client. The client returns an ACK response to LVS.
3. LVS verifies the validity of the sequence number in the ACK packet. If the sequence number is valid, LVS establishes a three-way handshake with the backend server.



To defend against ACK, FIN, and RST flood attacks, LVS checks the connection table and discards all requests for connections that are not defined in the table.

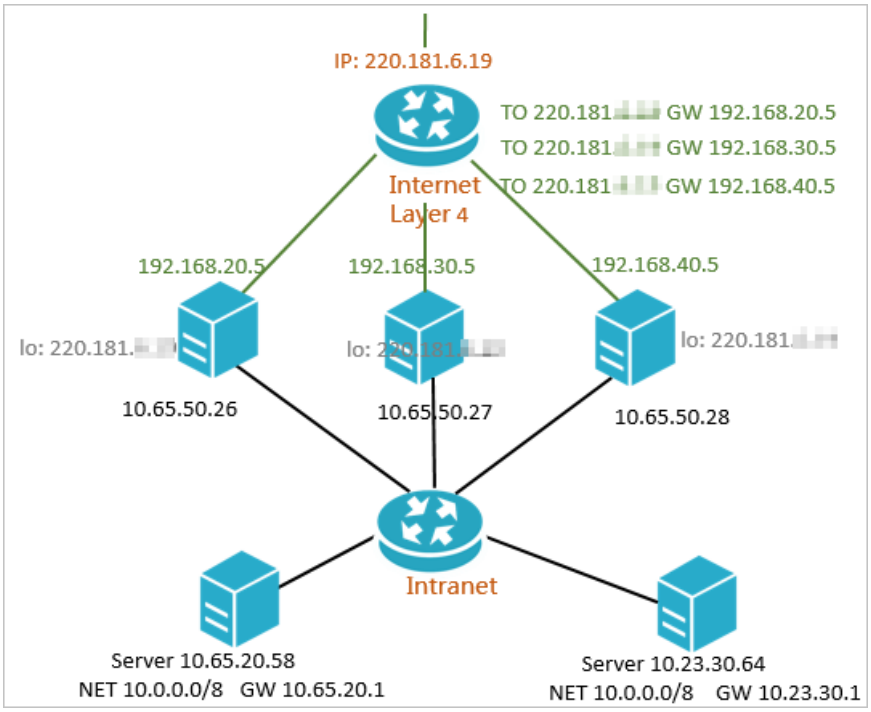
Cluster deployment

An LVS cluster communicates with uplink switches over Open Shortest Path First (OSPF). The uplink switches use equal-cost multi-path (ECMP) routes to distribute traffic to the LVS cluster. Then, the LVS cluster forwards the traffic to your servers.

The cluster deployment model ensures the stability of Layer-4 SLB with the following features:

- Robustness: LVS and uplink switches use OSPF as the heartbeat protocol. A VIP is configured on all LVS nodes in the cluster. The switches can locate the failure of any LVS node and remove it from the ECMP route list.
- Scalability: You can scale out an LVS cluster if traffic from a VIP exceeds the cluster capacity.

Cluster deployment



Keepalived optimization

Improvements made to Keepalived include:

- Change the asynchronous network model from select to epoll.
- Optimize the reloading process.

Features of Layer-4 SLB

In conclusion, Layer-4 SLB has the following features:

- High availability: The LVS cluster ensures redundancy and prevents SPOFs.
- Security: Together with Apsara Stack Security, LVS provides quasi-real-time defense.
- Health check: Health checks are performed on backend ECS instances to automatically remove unhealthy ones from the server pool until they restore.

18.4.2. Tengine in Layer-7 SLB

Tengine is a Web server project launched by Alibaba. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites. NGINX is one of the most popular open-source Layer-7 load-balancing software.

For more information about Tengine, visit http://tengine.taobao.org/.

Customized features

Tengine is customized for cloud computing scenarios:

- Inherits all features of NGINX 1.4.6 and is fully compatible with NGINX configurations.
- Supports the dynamic shared object (DSO) module. This means you do not need to recompile Tengine to add a module.
- Provides enhanced load balancing capabilities, including a consistent hash module and a session persistence module. It can also actively perform health checks on back-end servers and automatically

enable or disable servers based on their status.

- Monitors system loads and resource usage to protect the system.
- Provides error messages to help locate abnormal servers.
- Provides an enhanced protection module (by limiting the access speed).

Features of Layer-7 SLB combined with Tengine

Layer-7 Server Load Balancer (SLB) is based on Tengine, and has the following features:

- High availability: The Tengine cluster ensures redundancy and prevents single points of failure (SPOFs).
- Security: Tengine provides multi-dimensional protection against CC attacks.
- Health check: Tengine performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.
- Supports Layer-7 session persistence.
- Supports consistent hash scheduling.

19.Virtual Private Cloud (VPC)

19.1. What is a VPC?

A virtual private cloud (VPC) is a logically isolated virtual network.

Background information

The continuous development of cloud computing technologies leads to increasing virtual network requirements such as scalability, security, reliability, privacy, and performance. This scenario has hastened the birth of a variety of network virtualization technologies.

Earlier solutions combined virtual and physical networks to form a flat network architecture, such as large layer-2 networks. As the scale of virtual networks grew, earlier solutions faced more serious problems. A few notable problems include ARP spoofing, broadcast storms, and host scanning. Various network isolation technologies emerged to resolve these problems by completely isolating the physical networks from the virtual networks. One of the technologies utilized VLAN to isolate users, but due to VLAN limitations, it could only support up to 4096 nodes. It is insufficient to support the huge amount of users in the cloud.

Benefits

A VPC has the following benefits:

- **High security**

  Each VPC has an exclusive and unique tunnel ID, and a tunnel ID corresponds to only one VPC. VPCs are isolated by tunnel IDs.

- **Ease of use**

  You can quickly and easily create and manage a VPC in the VPC console. When you create a VPC, the system automatically provisions a VRouter and a route table for your VPC.

- **High scalability**

  A VPC can be partitioned into multiple subnets to deploy different services. Additionally, you can connect a VPC to an on-premises data center or another VPC to extend the network architecture.

Scenarios

VPCs allow you to flexibly customize the network configuration in the following scenarios:

- **Host Internet-facing applications**

  You can host Internet-facing applications in VPCs and enforce access limits with security group rules and whitelists. VPCs enable you to launch web servers in a public subnet but run your databases in private subnets for isolation and security purposes.

- **Host applications that require access to the Internet**

  By hosting an application in a subnet of a VPC, you can allow this application to receive Internet traffic by using a NAT gateway that provides source network address translation (SNAT). An SNAT rule allows outbound connectivity from the subnet to the Internet without exposing the private IP address of your instance. Furthermore, you can change the public IP address used in an SNAT mapping as needed to prevent targeted attacks.

- **Implement zone-disaster recovery**

Multiple VSwitches can be created in a VPC as subnets. Since VSwitches within a VPC can communicate with each other, they can be used to host your resources in different zones to implement zone-disaster recovery.

- **Isolate business units**

  You can utilize the logical boundaries between VPCs to isolate business units, such as production and test environments. When these business units need to communicate with each other, you can create a peering connection between the VPCs they reside to route traffic between them.

- **Extend your on-premises IT infrastructure**

  To expand the capacity of the existing infrastructure, you can establish a connection between your on-premises data center and a VPC. Moreover, your IT resources can be seamlessly migrated to the cloud without changing how users access these applications.

19.2. Benefits

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs:

- Similar to traditional networks, VPCs can also be divided into subnets. ECS instances in the same subnet use the same VSwitch to communicate with each other, whereas ECS instances in different subnets use VRouters to communicate with each other.
- VPCs are completely isolated from each other and can only be interconnected by mapping an external IP address (EIP or NAT IP address).
- The IP packets of an ECS instance are encapsulated by using the tunneling technology. Therefore, information about the data link layer (the MAC address) of the ECS instance is not transferred to the physical network. This way, ECS instances in different VPCs are isolated at Layer 2.
- ECS instances in VPCs use security groups as firewalls to control the traffic to and from ECS instances. This way, ECS instances in different VPCs are isolated at Layer 3.

19.3. Architecture

A VPC is a private network logically isolated from other virtual networks.

Network architecture

Each VPC consists of a private Classless Inter-Domain Routing (CIDR) block, a VRouter, and at least a VSwitch.

- **CIDR blocks**

  A CIDR block is a private IP address range in a VPC. The IP addresses of all cloud resources deployed in the VPC are within the specified CIDR block. When creating a VPC or a VSwitch, you must specify the private IP address range in the form of a CIDR block.

  You can use any of the following standard CIDR blocks and their subnets as the IP address range of the VPC.

CIDR block	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532

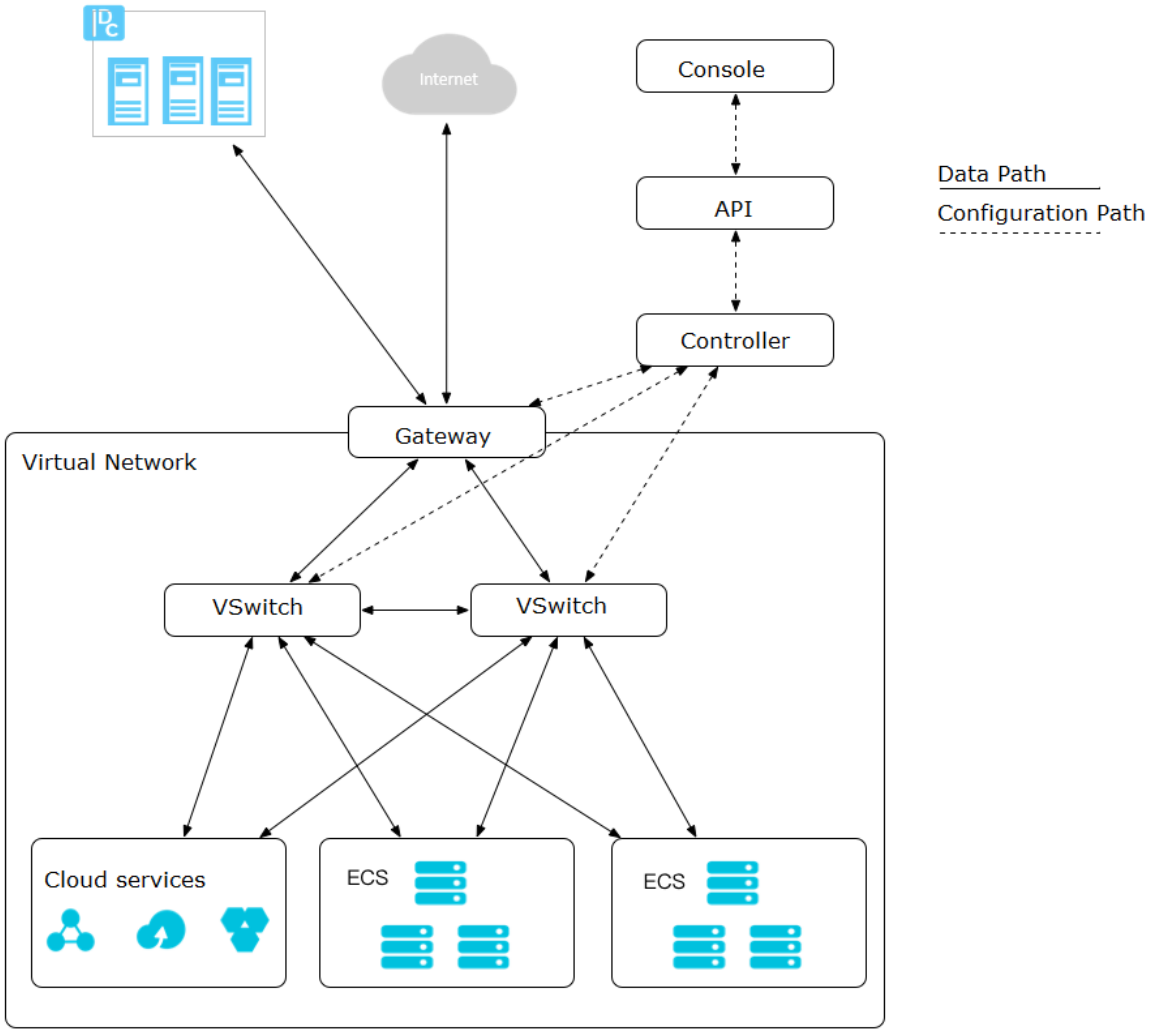
CIDR block	Number of available private IP addresses (system reserved ones excluded)
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- **VRouters**

  A VRouter is the hub of a VPC. A VRouter is also an important component of a VPC. The VRouter connects the VSwitches in a VPC and serves as the gateway connecting the VPC with other networks. After you create a VPC, the system automatically creates a VRouter, which is associated with a routing table.

- **Switches**

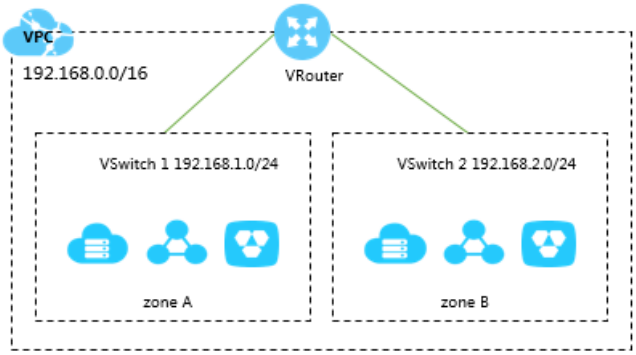
  A VSwitch is a basic network device in a VPC and is used to connect different cloud product instances. After creating a VPC, you can further divide the VPC into one or more subnets by creating VSwitches. The VSwitches within a VPC are interconnected. You can deploy applications in VSwitches of different zones to improve the service availability.



System architecture

The VPC architecture contains the VSwitches, gateway, and controller. The VSwitches and gateway form the key data path. Controllers use the protocol developed by Alibaba Cloud to forward the forwarding table to the gateway and VSwitches, completing the key configuration path. In the overall architecture, the configuration path and data path are separated from each other. VSwitches are distributed nodes. The gateway and controller are deployed in clusters. Multiple data centers are built for backup and disaster recovery. Redundant links are provided for disaster recovery. This deployment mode improves the overall availability of the VPC.

VPC architecture



19.4. Features

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. A unique tunnel ID is generated when tunnel encapsulation is performed on each data packet transmitted between the ECS instances within a VPC. Then, the data packet is transmitted over the physical network. ECS instances in different VPCs cannot communicate with each other. They have different tunnel IDs and therefore are on different routing planes.

Alibaba Cloud developed technologies such as the VSwitch, Software Defined Network (SDN), and hardware gateway based on the tunneling technology. These technologies serve as the basis for VPCs.

20.Apsara Stack Security

20.1. What is Apsara Stack Security?

Apsara Stack Security is a solution that provides Apsara Stack assets with a full suite of security features, such as network, server, application, data, and security management.

Background information

Traditional security solutions for IT services detect attacks on network perimeters. These solutions use hardware products such as firewalls and intrusion prevention systems (IPSs) to protect networks against attacks.

With the development of cloud computing, an increasing number of enterprises and organizations use cloud computing services instead of traditional IT services. Cloud computing features low costs, on-demand flexible configuration, and high resource utilization. Cloud computing environments do not have definite network perimeters. As a result, traditional security solutions cannot effectively safeguard cloud assets.

With the powerful data analysis capabilities and professional security operations team of Alibaba Cloud, Apsara Stack Security provides integrated security protection services for networks, applications, and servers.

Complete security solution

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services to provide a comprehensive security solution.

Security domain	Service name	Description
Security management	Threat Detection Service (TDS)	Monitors traffic and overall security status to audit and centrally manage security.
Server security	Server Guard	Protects ECS instances against intrusions and malicious code.
Application security	Web Application Firewall (WAF)	Protects web applications against attacks and ensures that mobile and PC users can securely access web applications over the Internet.
Network security	Anti-DDoS	Ensures the availability of network links and improves business continuity.
	Cloud Firewall	Allows you to centrally manage access control policies for traffic transferred within your business system (east-west) and between the Internet and your business system (north-south).
Data security	Sensitive Data Discovery and Protection (SDDP)	Prevents data leaks and helps your business system meet compliance requirements.
Security O&M service	On-premises security service	Helps you establish and optimize your cloud security system to protect your business system against attacks by using security features of Apsara Stack Security and other Apsara Stack services.

Security domain	Service name	Description

20.2. Advantages

Since the enforcement of China Internet Security Law, Regulations on Critical Information Infrastructure Security Protection and Cloud Security Classified Protection Standard 2.0 have been published. As a result, private cloud platforms must pass the classified protection evaluation to ensure the security of cloud systems. Increasing security threats such as attacker intrusions and ransomware have led to the rising needs for security issue detection and prevention.

At the network perimeter of Apsara Stack, Apsara Stack Security uses a traffic security monitoring system to detect and block network-layer attacks in real time. It detects and removes Trojans and malicious files on servers to prevent attackers from exploiting the servers. In addition, Apsara Stack Security can block brute-force attacks and send alerts on unusual logons. This prevents attackers from stealing or destroying business data after logging on the system with weak passwords.

In-depth defense system

Apsara Stack Security comprises multiple functional modules. These modules work together to provide in-depth defense on the Apsara Stack network perimeter, within the Apsara Stack network, and on the Elastic Compute Service (ECS) instances in Apsara Stack. To help you manage security risks of Apsara Stack in a centralized manner and in real time, Apsara Stack Security provides a unified security management system. This system allows you to manage the security policies in all security protection modules and perform association analysis on the logs.

The security protection modules provided by Apsara Stack Security cover network security, server security, application security, and threat analysis. Based on a management center that can integrate the security information from all modules, Apsara Stack Security can accurately detect and block attacks. In this way, Apsara Stack Security protects your business systems in the cloud against intrusions.

Security solutions completely integrated with the cloud platform

Apsara Stack Security is a product born from ten years of protection experience. After a decade of experience in providing security operations services for the internal businesses of Alibaba Group and six years of safeguarding the Alibaba Cloud security operations, Alibaba has obtained considerable security research achievements, security data, and security operations methods, and has built a professional cloud security team. Apsara Stack Security brings together the rich experience of these experts to develop the sophisticated systems that provide enhanced security for cloud computing platforms. This product can protect the cloud platform, cloud network environments, and cloud business systems of Apsara Stack users.

The components of Apsara Stack Security are software-defined, with a full hardware compatibility. With these components, you can implement elastic cloud computing services based on quick deployment, expansion, and implementation. The protection modules on the cloud network perimeter or in the cloud network adopt the bypass architecture, which completely fits the cloud businesses and has the minimal adverse impacts on the cloud businesses. The protection modules running on the ECS instances are all virtualized to fit the flexibility of the ECS instances.

User security situation awareness

The cloud platform provides services for users. In Apsara Stack Security console, a user can view the security protection data, generate security reports, and enable SMS and email alerts by configuring external resources.

Security capability output

Apsara Stack Security has accumulated a large number of protection policies over the last several years. The service has protected millions of users from hundreds of thousands of attacks every day. This has generated a large amount of security protection data. Apsara Stack Security analyzes over 10 TB of this data every day. The analysis results are used to enhance the fundamental security capabilities, such as the malicious IP library, malicious activity library, malicious sample library, and vulnerability library. These capabilities are applied in the protection modules of Apsara Stack Security to enhance your business security.

20.3. Architecture

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services.

Apsara Stack Security Standard Edition

- Threat Detection Service

  This module collects network traffic and server information, and detects possible vulnerability exploits, intrusions, and virus attacks through machine learning and data modeling. It also provides you with up-to-date information about ongoing attacks to help you monitor the security status of your businesses.

- Network Traffic Monitoring System

  This module is deployed on the network perimeter of Apsara Stack. It allows you to inspect and analyze each inbound or outbound packet of an Apsara Stack network through traffic mirroring. The analysis results are used by other Apsara Stack Security modules.

- Server Guard

  This module safeguards ECS instances by providing security features such as vulnerability management, baseline check, intrusion detection, and asset management. To do this, the module performs operations such as log monitoring, file analysis, and signature scanning.

- Web Application Firewall

  This module protects web applications against common web attacks reported by Open Web Application Security Project (OWASP), such as Structured Query Language (SQL) injections, cross-site scripting (XSS), exploitation of vulnerabilities in web server plug-ins, Trojan uploads, and unauthorized access. It also blocks a large number of malicious visits to avoid data leaks and ensure both the security and availability of your websites.

Apsara Stack Security Standard Edition also provides on-premises security services. These services help you better use the features of Apsara Stack products such as Apsara Stack Security to secure your applications.

On-premises security services include pre-release security assessment, access control policy management, Apsara Stack Security configuration, periodic security check, routine security inspection, and urgent event handling. These services cover the entire lifecycle of your businesses in Apsara Stack and help you create a security operations system. This system enhances the security of your application systems and ensures both the security and stability of your businesses.

Optional security services

You can also choose the following service modules to enhance your system security.

- DDoS Traffic Scrubbing

  This module detects and blocks distributed denial of service (DDoS) attacks.

- Cloud Firewall

  This module sorts and isolates businesses based on visualized business data to implement access control over east-west traffic in Apsara Stack.

- Sensitive Data Discovery and Protection

  This module uses Alibaba Cloud's big data analytics capabilities and artificial intelligence (AI) technologies to detect and classify sensitive data based on your business requirements. It masks sensitive data both in transit and at rest, monitors dataflows, and detects abnormal activities. This module provides visible, controllable, and industry-compliant security protection for your sensitive data by means of precise detection and analysis.

20.4. Features

20.4.1. Apsara Stack Security Standard Edition

20.4.1.1. Threat Detection Service

Threat Detection Service (TDS) is a system developed by the Alibaba Cloud security team for analyzing big data security.

This system analyzes server and network traffic to detect possible threats or attacks by using machine learning and data modeling. It identifies vulnerable exploits and potential virus attacks, and provides you with up-to-date information about ongoing attacks to help you monitor the security status of your businesses.

Features

The following table describes features of TDS.

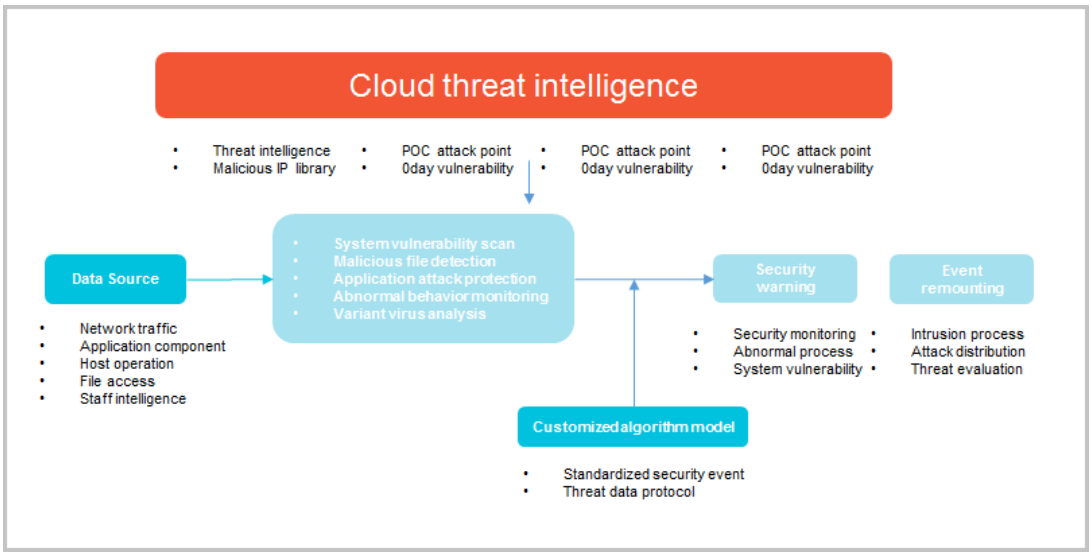
Feature	Description
Overview	Provides a comprehensive security overview with statistics on security score, asset status, unhandled alerts, and handled alerts.
Security Dashboard	Displays the security data on dashboards, including assets, vulnerabilities, baselines, attack sources, and attack distribution.
Security Alerts	Allows you to view and handle security events, including suspicious process, webshells, unusual logons, sensitive file tampering, malicious processes, suspicious network connections, and web application threat detection.

Feature	Description
Attack Analysis	Displays the attack trends and attack type distribution in the last 7 days and 30 days. Displays the attack information such as the attack time, attack source, attacked assets, number of attacks, risk level, and attack type.
Cloud Service Check	Checks the security configurations of cloud services from the aspects of network access control and data security. It supports periodic checks that run automatically and manual checks. You can verify the check results or configure whitelist policies for the check results.
Application Whitelists	Allows you to add servers to a whitelist based on intelligent learning and identifies programs as trusted, suspicious, or malicious based on the whitelist. Unauthorized processes will be terminated.
Assets	• Server: displays the security statuses for servers. You can view the numbers of all servers, risky servers, unprotected servers, inactive servers, and new servers. • Cloud Product: provides security status information for cloud services and supports SLB and NAT.
Security Reports	Allows you to query reports. For example, you can retrieve historical reports by report name.

How it works

The following figure shows how TDS works.

TDS working principle



- Big data security analysis platform
  - Network: TDS uses HTTP requests and responses collected by the traffic security monitoring module to create HTTP logs. It uses big data models to analyze the logs and discover security events and threats.

  - Server: TDS uses the rule engine to analyze server process data collected by Server Guard and discover security events and threats.

- Security event display
  - Security events reported by Server Guard
  - Server security events discovered in server process analysis conducted by the rule engine
  - Network security events discovered in HTTP log analysis conducted by big data models

Benefits

TDS provides the following benefits:

- Big data-based threat analysis

  TDS analyzes and computes petabyte-level big data. It collects all security data and threat information from the entire network. It also uses the machine learning technology to create comprehensive, intelligent security threat models that can be used in business scenarios with millions of users.

  This service focuses on the security trends and new threats that are faced by users of cloud computing services in data centers, such as targeted web application attacks and system brute-force attacks. It defends your systems against diverse threats.

- Dashboard

  To facilitate security decision making on Apsara Stack, TDS displays the results of big data threat analysis in graphs by using Internet visualization technologies.

20.4.1.2. Traffic Security Monitoring

The Traffic Security Monitoring module is an Apsara Stack Security service that can detect attacks within milliseconds.

By performing in-depth analysis on the traffic packets mirrored from the Apsara Stack network ingress, this module can detect various attacks and unusual activities in real time and coordinate with other protection modules to implement defenses. The Traffic Security Monitoring module provides a wealth of information and basic data support for the entire Apsara Stack Security defense system.

Features

The following table describes the features that the Traffic Security Monitoring module provides.

Feature	Description
Traffic data collection and analysis	Uses a bypass in traffic mirroring mode to collect inbound and outbound traffic that passes through the interconnection switch (ISW) and generates a traffic diagram.
Unusual traffic detection	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.
Malicious server identification	Detects attacks launched by internal servers to identify controlled malicious servers.

Feature	Description
Web application protection	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks, and exploitation of upload vulnerabilities and common content management system (CMS) vulnerabilities.
Suspicious TCP connection blocking	Uses a bypass to send TCP RST packets to the server and the client to block layer-4 TCP connections.
Network log recording	Records UDP and TCP traffic logs and the Request and Response logs of HTTP queries. Threat Detection Service (TDS) uses these logs for big data analysis.

How it works

The Traffic Security Monitoring module collects data, processes the data, and then generates data processing results. It uses sockets to exchange data.

- Collection: The module collects traffic data through multiple high-performance PCs with dual-port 10GE network interface controllers (NICs).
- Processing: Traffic from an IP address may pass through multiple collectors. Traffic data must be consolidated to generate usable information.
- Output: The module stores and provides the consolidated traffic data.

20.4.1.3. Server Guard

Server Guard provides security protection measures such as vulnerability management, baseline check, intrusion detection, and asset management for Elastic Compute Service (ECS) instances by means of log monitoring, file analysis, and feature scanning.

Server Guard uses the client-server model. To protect the security of ECS instances in real time, Server Guard clients work with the Server Guard server to monitor attacks and vulnerabilities at the system layer and the application layer on the ECS instances.

Features

Category	Feature	Description
Overview	Overview	Displays assets, vulnerabilities, exceptions, configuration defects, and events that require attention.

Category	Feature	Description
Servers	Server Fingerprints	Provides the following modules: • Port: checks and displays the listening port information, including the listening port, protocol, process, IP address, and update time. • Software: checks and displays the software installation information on servers, including the software name, software version, software installation directory, and update time. • Process: checks and displays the process information, including the process name, process path, startup parameter, startup time, user, permission, process ID (PID), parent process, and update time. • Account: checks and displays the host account information, including the account name, logon permission, root permission, user group, expiration time, last logon time, and update time. • Scheduled Tasks: checks and displays the scheduled tasks of the host, including the task path, execution command, task cycle, account name, and update time.
Threat Prevention	Baseline Check	Automatically detects configuration risks related to the system, account, database, weak password, and security compliance on your servers, and provides security hardening suggestions. This feature also checks database, system, and middleware assets.
	Vulnerabilities	Detects four types of vulnerabilities: Linux, Windows, Web CMS, and emergency vulnerabilities and provides vulnerability fix solutions. You can verify vulnerability fixes, view vulnerability details, and identify all vulnerabilities at one click.
Intrusion Prevention	Intrusions	Displays the alert information of affected host assets, including the number of alerting servers, the total number of unhandled alerts, and the number of urgent alerts.
	File Tamper Protection	Supports web page tamper-proofing and provides the blacklist and whitelist prevention modes.
	Virus Removal	Detects and removes virus and webshell. The system automatically detects and removes common trojan viruses, ransomware, mining viruses, and DDoS trojans.
Log Retrieval	Log Retrieval	Allows you to query logs for logon, brute-force attack, process snapshot, network connection, listening port snapshot, account snapshot, and process startup.
Server Settings	Client Installation	Allows you to view offline servers. You can install clients for the servers again based on the Client Installation Guide. You can uninstall the Server Guard client from the specified server.
	Protection Mode	Provides business first and protection first modes for different scenarios.

How it works

Server Guard uses the client-server model. The client is installed on ECS instances. The client communicates with the server through a TCP persistent connection and uses HTTP to obtain scripts, rules, and installer packages from the server.

The client can be used in Windows or Linux. It can automatically connect to the server for online updates.

Server Guard supports the following key features:

- **Vulnerability management**: The client collects the ECS instance information, including component information, software versions, file information, and registry information. Then, the client checks whether the information matches the vulnerability detection rules provided by the server. The information that matches the rules will be sent to the server for further analysis. The detected vulnerabilities will be displayed in the Server Guard console. You can fix vulnerabilities in the console or by calling API operations. After receiving the vulnerability patches from the server, the client on the vulnerable ECS instance automatically fixes the vulnerabilities and synchronizes the vulnerability status to the server.
- **Baseline check**: When you manually start a check or a periodic check is triggered, the Server Guard server sends a baseline check request to the client. The client then collects the server information according to the check policy and compares the information with the security baseline. Check items that do not comply with the baseline are labeled as at-risk items and reported to the server.
- **Unusual logon detection**: The client monitors the logon logs of the server system in real time. In a Linux system, the */var/log/secure* and */var/log/auth.log* files are also monitored. All failed and successful logons are recorded. Unusual logons or brute-force attacks will be reported to the server.
- **Webshell detection**: The client uses an Alibaba-developed dynamic webshell detection engine to detect complex webshells. It then restores these webshells to an identifiable status to analyze the hidden webshell activities. This prevents webshells from bypassing the detection due to the use of static detection rules.
- **Suspicious process detection**: The Server Guard server uses a data analysis rules engine to analyze the server process data collected by the client. By doing so, the server can detect suspicious processes such as reverse shells, mining processes, DDoS trojans, worms, viruses, and hacking tools.
- **Log collection**: The client collects logs such as processes logs and network logs.

Scenarios

Server Guard is applicable to server security protection in the following scenarios:

- **Use common software for website building**

  In this scenario, attackers may intrude servers by exploiting vulnerabilities in common software. You can use Server Guard to detect and fix vulnerabilities.

- **Use Web application services**

  Attackers may steal website data through both internal and external web services. You can use Server Guard to prevent attackers from launching attacks or controlling your servers.

20.4.1.4. WAF

Web Application Firewall (WAF) protects the web applications of cloud users against common web attacks.

Different from traditional web application firewalls, Apsara Stack WAF uses intelligent semantic analysis algorithms to identify web attacks. WAF also integrates a learning model to enhance its analysis capability so that it can meet your daily security protection requirements without relying on traditional rule libraries.

WAF protects the traffic of businesses on HTTP and HTTPS websites. In the WAF console, you can import certificates and private keys to enable end-to-end encryption. This prevents the interception of business data on the links.

WAF not only prevents common web application attacks defined by Open Web Application Security Project (OWASP) but also mitigates HTTP flood attacks. In addition, WAF allows you to customize protection policies based on the businesses of your website to block malicious web requests.

Features

The following table describes the features provided by WAF.

Category	Feature	Description
Detection Overview	Detection Overview	Provides statistics on protection for the last 24 hours and the last 30 days.
	Access Status Monitor	Displays the top 100 access requests in real time.
	Export Detection Report	Allows you to export daily reports, weekly reports, and scheduled task reports.
	Attack Detection Statistics	Provides statistics on attack detection.
Detection Logs	Attack Detection Logs	Provides attack detection logs. The log list displays the processing results, attacked addresses, attack types, attacker IP addresses, and attack time. You can view log details for each attack.
	HTTP Flood Detection Logs	Provides HTTP flood protection logs. The log list displays logs for matched HTTP flood protection rules, including the request URL, the name of the matched rule, and the match time. You can filter logs based on the event generation time and the name of the HTTP flood protection rule.
	System operation log	Provides system operations logs, including usernames, operations, and IP addresses.
	Access Log	Provides access logs, including the access address, destination IP address, source IP address, request method, and response code.
	Protection site management	Allows you to create, delete, modify, enable, and disable function forwarding proxies of a protected site.
	Customized Rules	Allows you to create, delete, enable, and disable custom rules. This implements fine-grained HTTP access control for websites.

Category	Feature	Description
Protection Configuration	Website Protection Policies	• Supports decoding methods, such as URL decoding, JSON parsing, Base64 decoding, hexadecimal conversion, backslash unescape, XML parsing, PHP deserialization, and UTF-7 decoding. • Detects SQL injections, cross-site scripting (XSS), intelligence, cross-site request forgery (CSRF), server-side request forgery (SSRF), Hypertext Preprocessor (PHP) deserialization, Java deserialization, Active Server Pages (ASP) code injections, file inclusion attacks, file upload attacks, PHP code injections, command injections, crawlers, and server responses. • Provides five built-in protection templates, including the template with default protection policies, monitoring mode template, anti-DDoS template, template for financial customers, and template for Internet customers. WAF allows you to customize the decoding algorithms in the templates, enable or disable each attack detection module separately, and configure the detection granularity. WAF also allows you to specify the Block Status Code parameter. • Allows you to enable HTTP response detection and configure the length of the response body in detection rules. • Allows you to configure the length of the request body in detection rules. • Allows you to enable or disable detection timeout settings.
	HTTP Flood Protection	Allows you to configure access frequency control rules for domain names and URLs. This restricts the access frequency of IP addresses or sessions that meet the criteria, or blocks these IP addresses or sessions. Restricts the access frequency of known IP addresses or sessions or blocks these IP addresses or sessions. Supports the HTTP flood protection whitelist function. HTTP flood protection rules are not applicable to IP addresses or sessions in a whitelist.
	SSL Certificate Management	Allows you to upload certificate files and SSL private keys to manage SSL certificates.
System Management	Node status	• Payload Status: displays the CPU utilization and memory usage. • Node Network Status: displays the read throughput and write throughput. • Detection Status: displays the queries per second (QPS) and the average detection time consumed by WAF nodes. • Forward Status: displays the number of new connections per second and the average latency. • Disk Status: displays the disk usage and total disk size.
	Syslog Configuration	Configures syslog to send logs and also configures the service- and system-related alert thresholds.

How it works

WAF performs protocol parsing and in-depth decoding on the web access traffic. It then calls the access control, rule detection, and semantic analysis engines to analyze the traffic and determines whether to allow or block the traffic based on the preset policies. Besides, WAF provides a good human-machine interaction interface for administrators to adjust protected websites and security policies.

Scenarios

WAF can be used for web application protection in fields such as government, finance, insurance, e-commerce, online to offline (O2O), Internet Plus, and games. It provides the following features:

- Prevents website data leaks caused by SQL injections.
- Mitigates HTTP flood attacks by blocking a large number of malicious requests. This ensures the availability of your website.
- Prevents website defacement arising from trojans to ensure the credibility of your website.
- Provides virtual patches that enable quick fixes for newly discovered vulnerabilities.

20.4.1.5. Security Operations Center (SOC)

Security Operations Center (SOC) provides security administrators with centralized management of all users and the platform and analysis functions of Apsara Stack logs.

Features

SOC provides the following features:

Feature	Description
Dashboard	Allows you to view the overall security statistics and perform operations.
Security monitoring	Allows you to view the security events of all users and the platform.
Asset management	Allows you to view the security status of user assets and platform assets.
Log analysis	Analyzes logs from multiple data sources, detects unexpected alerts, and improves alert detection of Apsara Stack.
Report management	Allows you to quickly export reports for various purposes.
System configurations	Allows you to configure system features such as alerts, updates, global policies, and account management.

Scenarios

- Scenario 1: routine monitoring

  SOC regularly inspects system security. Currently, SOC focuses on security issues on the users. The following features are provided:

  - Urgent risk detection: checks for urgent security risks on a daily basis. Security risks include user security alerts, vulnerabilities, and server configuration risks.

- Risk management: identifies and handles high-risk security alerts, vulnerabilities, and server configuration risks.
- Attack data collection: shows the number of attacks and attack protection information.
- Security reports: sends daily, weekly, or monthly security reports to users.

- Scenario 2: security evaluation for new assets

  Monitors asset changes, detects new assets, and evaluates asset security. Generates security evaluation reports on new assets to help you determine whether to add these assets to your network. The following features are provided:

  - Scans vulnerabilities on servers and web applications.
  - Verifies server configurations.
  - Performs baseline check on cloud services.

- Scenario 3: urgent event handling and cause tracking

  After an urgent event is detected, SOC handles the event and tracks the event cause.

20.4.1.6. On-premises security operations services

To ensure the stability, reliability, security, and regulatory compliance of the cloud platform, Apsara Stack Security Standard Edition provides multiple security products and on-premises security operations services to ensure the availability, confidentiality, and integrity of the systems and data of users. Security operations services are indispensable in the security system. The combination of security products and security operations services gives full play to the security features of both Apsara Stack products and Apsara Stack Security products, and enhances the security of the Apsara Stack network environment from both technology and management aspects.

On-premises security operations services aim to help users use the security features of both Apsara Stack products and Apsara Stack Security products to protect the user applications. Security operations services include services that cover the entire security lifecycle of Apsara Stack user businesses, such as pre-release security assessment, access control policy optimization, periodic security assessment, routine security inspection, and emergency response. These services help users create a cloud security operations system to enhance the application system security and ensure secure and stable businesses.

Services

On-premises security operations services are as follows:

On-premises security operations services

Category	Service	Description
	User asset research	With the authorization of a user, this service periodically researches the cloud businesses of the user and develops a business list containing information such as the business system name, ECS, RDS, IP address, domain name, and owner.

Category	Service	Description
User business security operations	New business security assessment	• Before a user migrates a new business system to the cloud, this service detects system vulnerabilities and application vulnerabilities in the new business system using both automation tools and manual operations. • Provides advice and verification on vulnerability fixes.
	Periodic business security assessment	• Periodically uses automation tools to detect system vulnerabilities, application vulnerabilities, and security risks in running businesses. • Provides advice on handling detected risks, including but not limited to security policy settings, patch updates, and application vulnerability handling.
	Access control management	Provides inspection and guidance on applying access control policies when a new business is migrated to the cloud.
	Access control routine inspection	Periodically checks for access control risks of user businesses.
	Security risk routine inspection	Monitors and inspects security events in Apsara Stack Security. Informs the user of verified events and provides advice on event handling.
Apsara Stack Security operations	Rule update	Periodically updates the rule libraries of Apsara Stack Security products.
	Product integration	• Provides support for integrating Apsara Stack Security products with the application systems of users. • Helps users customize and optimize security policies.
Security event response	Event alerts	Synchronizes recent security events information from Alibaba Cloud, and helps users remove the risks.
	Event handling	Handles urgent events such as attacker intrusions.

Service output

On-premises security operations services output the following documents:

- Weekly, monthly, and yearly service reports
- Asset lists
- System security check reports

SLA

The SLA terms of on-premises security operations services are as follows:

- **Asset management**: Update the asset list once a month.
- **Security event response**: Respond within 30 minutes during work hours.
- **Security check**:
  - Complete a pre-release security check within two workdays.
  - Perform a periodic security check once a quarter.

Duties

Partners authorized by Alibaba Cloud provide on-premises security operations services, and Alibaba Cloud provides service quality management and technical support.

Owner	Duties
Alibaba Cloud	• Assign and manage tasks of service providers and on-premises engineers. • Assess the services provided by service providers and on-premises engineers. • Train service providers and on-premises engineers and provide technical support. • Provide project coordination and process and quality management.
Service provider	• Perform security check and routine inspection on the system of the user. • Provide advice on fixing vulnerabilities. • Maintain the access control policies of the user resources. • Update and maintain the security rules and policies of Apsara Stack Security. • Respond to security events. • Provide security technical support for users.
User	• Authorize service providers to perform security operations. • Follow the security advice to carry out the security plans on businesses. • Improve the security system.

Risk control

The following measures are taken to control risks in on-premises security operations services:

Category	Risk Item	Measure
Engineer and organization qualification	Organization	Only Alibaba Cloud and authorized enterprises can provide security services.
	Engineers	All engineers must be assessed and trained by the Alibaba Cloud security team.
Confidentiality	Confidentiality agreements	All enterprise and individual service providers must sign a confidentiality agreement.

Category	Risk Item	Measure
Service tool security	Tool selection	Only security tools specified by Alibaba Cloud are allowed.
	Tool use	Apply standard configurations to avoid risks in using the tools.
Operation security	Operation procedure	Perform at-risk operations, such as scanning, in batches.
	Risk notification	Inform the users of risks in the operations, and provide risk avoidance and control methods. Perform operations only with the consent of the users.

20.4.2. Optional security services

In addition to the security services provided by Apsara Stack Security Standard Edition, multiple optional security services are also provided to meet various security needs. We recommend that you choose optional security services based on your business needs.

20.4.2.1. DDoS Traffic Scrubbing

Backed by its large-scale and distributed operating system and more than a decade of experience in defending against security attacks, Alibaba Cloud has designed and developed the DDoS Traffic Scrubbing module based on the cloud computing architecture to protect the Apsara Stack platform against large amounts of distributed denial of service (DDoS) attacks.

Features

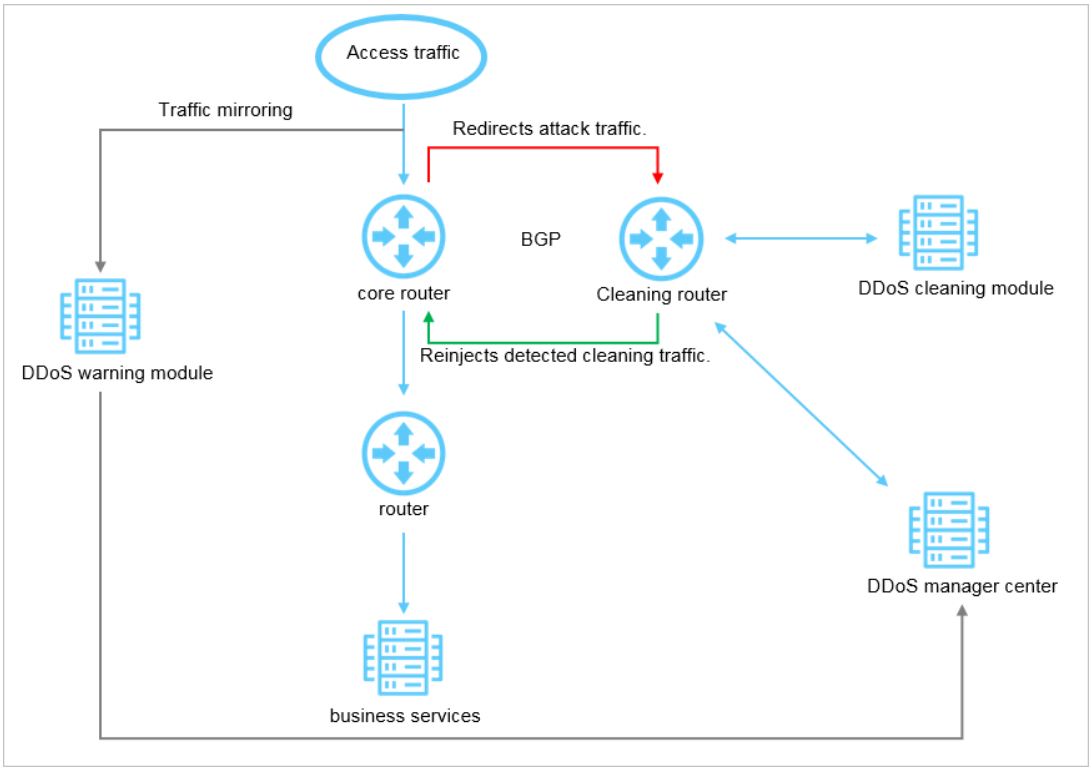
The following table describes the features provided by the DDoS Traffic Scrubbing module.

Feature	Description
Traffic scrubbing against DDoS attacks	Detects and prevents attacks such as SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, and HTTP flood.
DDoS attack display	Allows you to view DDoS attacks in the console and search for DDoS attacks by IP address, status, and event information.
DDoS traffic analysis	Allows you to monitor and analyze the traffic of a DDoS attack, and view the attack traffic protocol and the top 10 IP addresses that have launched most attacks.

How it works

After the Traffic Security Monitoring module detects unusual traffic, the DDoS Traffic Scrubbing module reroutes, scrubs, and reinjects the traffic, as shown in Traffic scrubbing. This mitigates DDoS attacks and ensures normal running of businesses.

Traffic scrubbing



The Traffic Security Monitoring module sends information about the detected DDoS attacks to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module is connected to the border gateway device. When a DDoS attack is detected, this module configures a Border Gateway Protocol (BGP) path for the border gateway to reroute the attack traffic to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module then scrubs the traffic based on the configured scrubbing policies, filters out unusual traffic, and reinjects the normal traffic to the border gateway.

> **Note** Apsara Stack Security cannot scrub the traffic between internal networks.

Advantages

The DDoS Traffic Scrubbing module has the following feature advantages:

- **Detection of all common DDoS attacks**

  This module protects you from various DDoS attacks, such as HTTP flood, SYN flood, UDP flood, UDP DNS query flood, stream flood, ICMP flood, and HTTP GET flood, at the network layer, transport layer, and application layer. This module also informs you of the website defense status through real-time SMS messages.

- **Automatic response to attacks within one second**

  This module uses the world leading attack detection and prevention technologies. It can complete the protection process within one second, covering attack discovery, traffic rerouting, and traffic scrubbing. This module triggers traffic scrubbing when the traffic scrubbing thresholds are violated or when DDoS attacks are detected during network behavior analysis. This reduces network jitter and ensures the availability of your businesses in the case of DDoS attacks.

- **High scalability and high redundancy of anti-DDoS capabilities**

  With high scalability and high redundancy of the cloud computing architecture, this module can be easily scaled up to realize high scalability of anti-DDoS capabilities.

- **Bidirectional protection to avoid the abuse of cloud resources**

  This module not only protects your system against external DDoS attacks but also detects resource abuse in your cloud environment. If any of your cloud resources in Apsara Stack is used to launch DDoS attacks, the Traffic Security Monitoring module will cooperate with Server Guard to restrict the network access of the hijacked resource and generate an alert.

20.4.2.2. Cloud Firewall

Cloud Firewall manages north-south traffic in a centralized manner and provides access control and traffic analysis features to better protect your network.

Features

The following table describes the features provided by Cloud Firewall.

Category	Feature	Description
Internet Firewall	Access Control	Supports Internet firewalls. You can configure outbound and inbound policies, including the access source, destination type, destination, protocol, port type, port, application, and policy action.
	Firewall Switch Policy	Allows you to search for target assets by asset type, region, instance ID, and IP address. You can enable firewall policies, including Internet Firewall, VPC-VPC, and IDC-VPC policies.
	Intrusion Prevention Policies	Allows you to set the threat engine mode to the monitoring mode or traffic control mode, to configure an IP address whitelist for outbound and inbound policies, and to customize basic policies for basic protection. You can use the Virtual Patches function and turn on the function at one click. This feature protects your network against abnormal connections, command execution, brute-force cracking, scanning, information leakage, distributed denial of service (DDoS) attacks, overflow attacks, web attacks, backdoors, trojans, worms, mining, and reverse shells.
	Event Log	Allows you to search for event logs by source IP address, destination IP address, event type, action, and time range.
	Traffic Log	Allows you to search for traffic logs by different conditions.
	VPC Firewall	Detects and controls communication traffic between two VPCs. You can configure VPC firewall policies, including the access source, destination type, protocol type, port type, application, and policy action.
	Internal Firewall	Controls inbound and outbound traffic between ECS instances. You can configure internal firewall policies, including the access source, destination, protocol type, port range, and action.

Category	Feature	Description
VPC Firewall	IDC-VPC Firewall	You can configure IDC-VPC firewall policies, including the access source, destination type, protocol type, port type, application, and action.
	Firewall Switch Policy	Allows you to search for target assets by asset type, region, instance ID, and IP address. You can enable the firewall policies, including Internet Firewall, VPC-VPC, and IDC-VPC policies.
	Intrusion Prevention Policies	Allows you to set the threat engine mode to the monitoring mode or traffic control mode, to configure an IP address whitelist for outbound and inbound policies, and to customize basic policies for basic protection. You can use the Virtual Patches function and turn on the function at one click. This feature protects your network against abnormal connections, command execution, brute force cracking, scanning, information leakage, distributed denial of service (DDoS) attacks, overflow attacks, web attacks, backdoors, trojans, worms, mining, and reverse shells.
	Event Log	Allows you to search for event logs by source IP address, destination IP address, event type, action, and time range.
	Traffic Log	Allows you to search for traffic logs by different conditions.

Scenarios

Cloud Firewall is applicable to the following scenarios:

- Control the access traffic from the Internet to ECS instances: For example, a financial company on Alibaba Cloud uses IPS to protect their HTTP and other businesses exposed on the Internet.
- Prevent command-and-control activities: For example, a governmental organization on Alibaba Cloud analyzes not only the access traffic from the Internet to ECS instances but also the command-and-control traffic from ECS instances to the Internet. Based on the analysis, the organization can determine which ECS instances are at risk and block anomalous access in real time to avoid potential risks.

Benefits

- Firewall as a service (FWaaS), which is easy to use

  Cloud Firewall uses the SDN technology. You can use Cloud Firewall after a simple policy configuration. Cloud Firewall helps you get rid of the basic but complex system and network configurations such as image installation and routing setup that are required by traditional firewalls. In addition, you do not need to be concerned about issues such as disaster recovery, capacity expansion, and deployment.

- Stability and reliability

  Cloud Firewall is deployed in dual available zone (AZ) mode. The failure of any server or AZ does not cause Cloud Firewall to break down.

- Centralized policy management

Cloud Firewall provides complete north-south traffic control for your assets. You can fully control access to your ECS instances and isolate ECS instances for security.

With Cloud Firewall, you can control access to common cloud assets such as ECS, RDS, and SLB instances at the network level and resolve anomalous access issues.

- Real-time intrusion prevention

  With the built-in IPS, Cloud Firewall can receive simultaneous updates of network-wide threat intelligence, and detect and block threats from the Internet in real time.

- Business relationship visibility

  Cloud Firewall shows assets and their access relationships in topology views. After you subscribe to the Cloud Firewall service, you can gain instant visibility of your business regions, groups, assets, access relationships between assets, and clustering analysis of user traffic without any configurations. Cloud Firewall supports visual analysis of traffic to maximize the correctness of policies.

20.4.2.3. Sensitive Data Discovery and Protection

Sensitive Data Discovery and Protection (SDDP) is a data security service used to detect and protect sensitive data in Apsara Stack big data services.

SDDP uses Alibaba Cloud's big data analytics capabilities and artificial intelligence (AI) technologies to detect and classify sensitive data based on your business requirements. It can also both dynamically and statically mask sensitive data, monitor dataflows, and detect abnormal activities. It provides visible, controllable, and industry-compliant security protection for your sensitive data by using precise detection and analysis. SDDP can detect and protect sensitive data in a variety of Apsara Stack big data services, such as MaxCompute, Object Storage Service (OSS), AnalyticDB, Tablestore, and ApsaraDB for RDS.

Features

The following table describes features of SDDP.

Feature		Description
Classification and detection of sensitive data	Detection of new data	A department administrator can authorize SDDP to scan and protect data assets based on business requirements. SDDP only scans and monitors authorized data assets.
	Sensitive data classification	SDDP can classify sensitive data in big data services such as MaxCompute, OSS, AnalyticDB, Tablestore, and ApsaraDB for RDS. You can define classification rules for sensitive data by using methods such as keywords and regular expressions.
	Sensitive data detection	SDDP has built-in algorithms for detecting sensitive data, and uses file clustering, deep neural networks, and machine learning to detect sensitive images, text, and fields.

Feature		Description
Management of sensitive data permissions	Asset permissions detection	SDDP can redirect you to pages that display the permissions of data assets and allows you to view the accounts that have permissions to access those assets. The data assets include MaxCompute projects, MaxCompute tables, MaxCompute columns, MaxCompute packages, AnalyticDB databases, AnalyticDB tables, OSS buckets, Tablestore instances, and Tablestore tables
	Account permissions detection	SDDP allows you to view all accounts in a department and search for departments or accounts in fuzzy search mode. SDDP displays relationships between departments and accounts in a hierarchical and visible layout.
	Abnormal permissions usage detection	SDDP automatically detects abnormal permissions usage in big data services, such as MaxCompute, OSS, AnalyticDB, and Tablestore.
Monitoring of dataflows and operations	Dataflow monitoring	SDDP monitors dataflows among entities, including data storage services (such as MaxCompute, OSS, AnalyticDB, and Tablestore), data transmission services (such as DataHub and CDP), the data stream processing service Blink, external databases, and external files. It displays dataflows and abnormal activities on dynamic graphs. This way, you can click an abnormal activity on a graph to redirect to the page for handling the abnormal activity.
	Abnormal data operation detection	SDDP detects abnormal operations in big data services, such as MaxCompute, OSS, AnalyticDB, and Tablestore.
	Abnormal dataflow detection	SDDP detects abnormal dataflows (including abnormal downloads) in big data services, such as MaxCompute, OSS, AnalyticDB, and Tablestore.
	Detection rule customization	SDDP allows you to customize rules for detecting abnormal dataflows and operations based on algorithms.
Abnormal activity processing	Configuration for abnormal activity detection	SDDP allows you to configure thresholds and rules for detecting abnormal activities, such as abnormal dataflows, permissions usage, and data operations.
	Abnormal activity processing	SDDP processes abnormal activities with a built-in console. You can search for abnormal activities by department, event type, account, processing status, or time of occurrence.
	Abnormal activity statistics	SDDP collects statistics on the processing status of abnormal activities, including abnormal dataflows, permissions usage, and data operations, and then dynamically displays these statistics.

Feature		Description
Static data masking	Static data masking	SDDP statically masks sensitive data in big data services, such as MaxCompute, OSS, AnalyticDB, Tablestore, and ApsaraDB for RDS. It supports the following masking algorithms: hash masking, shield masking, substitution masking, conversion masking, encryption masking, and shuffle masking.
Intelligent audit	Intelligent audit	SDDP collects and audits the operation logs of big data services, such as MaxCompute, OSS, and ApsaraDB for RDS.

Scenarios

- Complies with laws and regulations on personal information protection.

  SDDP detects personal information in large amounts of data, automatically marks risk levels for personal information, and effectively detects data leaks. By using SDDP, enterprises can ensure that their systems comply with laws and regulations on personal information protection.

- Classifies and protects sensitive data of enterprises.

  SDDP classifies and detects sensitive data, manages data permissions, and identifies abnormal activities (such as abnormal dataflows, permission usage, and data operations) based on specified rules. This way, enterprises can properly protect their sensitive data of diverse classifications.

- Handles data leaks.

  SDDP detects abnormal activities based on specific rules and allows you to centrally summarize and handle these activities. This helps enterprises process data leaks online and provides effective support for security O&M.

Benefits

As a data security module of Alibaba Cloud Security, SDDP can detect and protect sensitive data in real-time computing services (such as Blink, DataHub, AnalyticDB, and Tablestore) and offline computing services (such as MaxCompute and OSS). It can detect structured, semi-structured, and unstructured sensitive data based on the same standards. SDDP provides the following benefits:

- Precise detection

  SDDP uses a built-in rule engine, a natural language processing model, and a neural network model to precisely detect sensitive personal information, sensitive system configurations, and confidential documents in a large amount of data.

- Closed-loop management

  SDDP implements closed-loop management that covers detection, protection, and handling to help enterprises effectively avoid risks.

- Intelligent detection

  SDDP provides an intelligent and multi-level filtering model to effectively detect abnormal activities and meet operational requirements.

- Flexible definition

SDDP allows you to customize a variety of data based on your business requirements, such as rules for detecting sensitive data, definitions of sensitive data, and thresholds and rules for detecting abnormal activities.

21.Key Management Service (KMS)

21.1. What is KMS?

Key Management Service (KMS) is a one-stop service platform for key management and data encryption. KMS provides simple, reliable, secure, and standard-compliant capabilities to encrypt and protect data. KMS greatly reduces your costs of purchase, operations and maintenance (O&M), and research and development (R&D) on cryptographic infrastructure and data encryption services. This helps you focus on the business development.

KMS provides the following features:

- Encryption key hosting

  KMS supports encryption key hosting. An encryption key hosted on KMS is called a customer master key (CMK). You can manage the lifecycle of a CMK by enabling or disabling the CMK.

- BYOK

  KMS supports Bring Your Own Key (BYOK). You can import your own keys to KMS to encrypt data on the cloud. This facilitates key management. You can import the following types of keys to KMS:

  - Keys in your on-premises key management infrastructure (KMI)
  - Keys in user-managed hardware security modules (HSMs) of Data Encryption Service

  > **Note** Keys imported to managed HSMs in KMS cannot be exported by using any method because secure key exchange algorithms are used in KMS. Operators or third parties are not allowed to check the plaintext of keys.

- Automatic rotation of encryption keys

  A CMK in KMS can have multiple key versions. Each version represents an independently generated key and does not have any relation with other versions. KMS automatically rotates encryption keys. This helps you implement the best security practices and comply with audit requirements. For more information, see the Overview and Automatic key rotation topics of *Key rotation* in *User Guide*.

- Fully managed HSMs

  KMS provides fully managed HSMs. You can host keys in HSMs. Cryptographic operations are implemented in HSMs to protect key security.

  > **Note** To use this feature, you must purchase an HSM and the KMS license of the Advanced edition.

- Simple cryptographic API operations
  - KMS provides cryptographic API operations that are simpler than those for traditional cryptographic modules or cryptographic software libraries.
  - Encryption keys in KMS support authenticated encryption with associated data (AEAD) and deliver additional authenticated data (AAD) to protect data integrity. For more information, see the EncryptionContext topic of *Use symmetric keys* in *User Guide*.

- CMK aliases

KMS allows you to create CMK aliases, which facilitate CMK usage. For more information, see the Use aliases topic in *User Guide*. For example, you can use CMK aliases to manually rotate CMKs in specific scenarios.

- Resource tags

  KMS supports resource tags, which facilitate key resource management.

21.2. Features

21.2.1. Convenient key management

You can call KMS API operations or perform operations in the KMS console to manage CMKs.

- You can disable or enable CMKs at any time. After a CMK is disabled, the data encrypted by using this CMK cannot be decrypted.
- You can schedule the deletion of a CMK by specifying a waiting period. You can cancel the scheduled deletion of a CMK at any time before the waiting period ends. This prevents CMKs from being accidentally deleted.
- You can use RAM to manage permissions on CMKs and separate encryption and decryption permissions.
- You can use EncryptionContext to enhance control over keys and ciphertext.

21.2.2. Envelope encryption

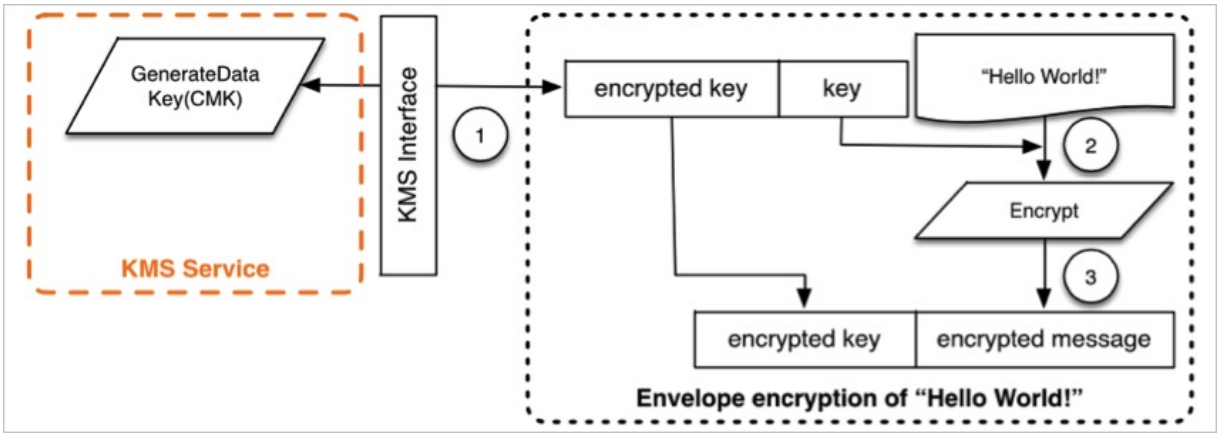
Envelope encryption is an encryption mechanism similar to the digital envelope technology. Envelope encryption allows you to encrypt data by using data keys (DKs) and encapsulate DKs in an envelope to ensure the security of their storage, transfer, and use. CMKs are not used to encrypt or decrypt data directly.

Although KMS provides the Encrypt API operation, KMS does not directly encrypt data. KMS manages CMKs and uses CMKs to encrypt and decrypt DKs. DKs are used to encrypt data.

You can use your own DK to encrypt data and then call the Encrypt API operation to encrypt the DK. You can also call the GenerateDataKey API operation to generate a DK.

Encryption process

The following figure shows the encryption process.



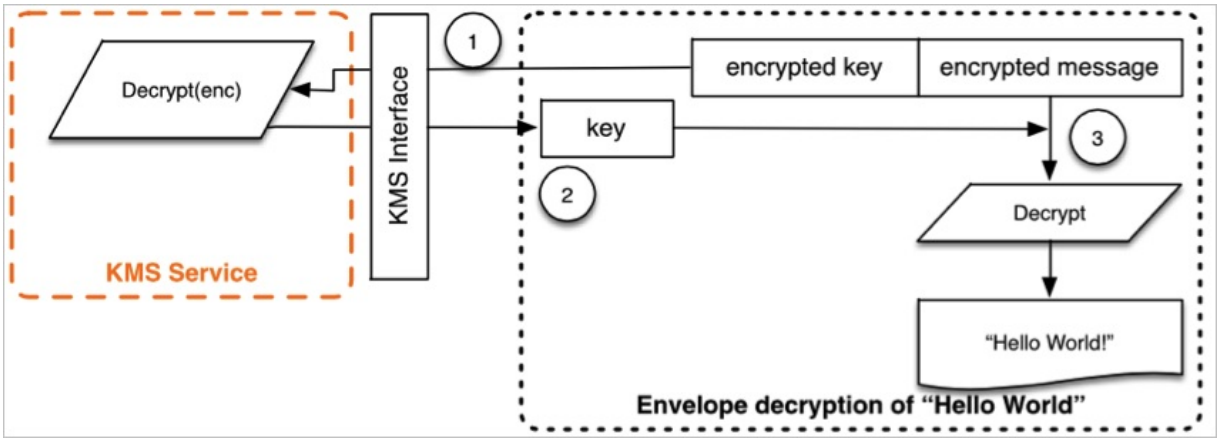
Encryption procedure:

1. Use a specific CMK to generate a DK. KMS returns the plaintext and ciphertext of a DK.

   Alternatively, you can call the Encrypt operation to encrypt your own DK. KMS returns the encrypted DK.
2. Use the DK to encrypt your data. KMS returns the ciphertext of the data.
3. Store the encrypted DK and the ciphertext of the data in your storage device.

Decryption process

The following figure shows the decryption process.



Decryption procedure:

1. Use KMS to decrypt the encrypted DK. The plaintext DK is returned.
2. KMS returns the plaintext DK.
3. Use the plaintext DK to decrypt the ciphertext of your data. The plaintext data is returned.

21.2.3. Secure key storage

KMS uses the following methods to ensure key security:

- The plaintext of CMKs is stored only in the memory of hardened security appliance (HSA) modules, whereas the ciphertext of CMKs is stored only in the storage modules of KMS.
- CMKs are encrypted by using DomainKeys managed by HSA modules. DomainKeys are rotated on a daily basis.
- DomainKeys are encrypted by using a trusted computing technology and stored based on a distributed storage protocol. This ensures the high reliability of DomainKeys.

22.Apsara Stack DNS

22.1. What is Apsara Stack DNS?

Apsara Stack DNS is a service that runs on Apsara Stack to resolve domain names over internal networks, such as VPCs, data centers, and the classic network. You can configure rules to map domain names to IP addresses. Apsara Stack DNS then distributes domain name requests from clients to cloud resources, user-created business applications, business systems on your internal networks, or the business resources of Internet service providers (ISPs).

Apsara Stack DNS provides the DNS resolution and Global Server Load Balancer (GSLB) services in VPCs, data centers, and the classic network. You can perform the following operations by using Apsara Stack DNS in these internal networks:

- Access other ECS instances deployed in the same VPC.
- Access other cloud service instances on Apsara Stack.
- Access enterprise business systems.
- Access services over the Internet.
- Use the GSLB service to implement multiple-active solutions and disaster recovery, such as local active-active, local multi-active, remote active-active, active geo-redundancy, and geo-disaster recovery.
- Connect to Apsara Stack DNS with your own DNS servers over a leased line to achieve hybrid cloud integration for your business.

22.2. Benefits

As a key network service, Apsara Stack DNS controls data flows that go through Apsara Stack, resolves domain names, balances server loads, and connects Apsara Stack to data centers. Apsara Stack DNS provides you with multiple solutions for cloud environment deployment, zone high availability, server load balancing, and disaster recovery to support your IT operations.

Enterprise domain name management

Apsara Stack DNS provides management and resolution services for your domain names. It supports the following features:

- Performs forward and reverse DNS resolutions for domain names of cloud service instances, such as ECS instances.
- Performs forward and reverse DNS resolutions for your internal domain names.
- Allows you to add, modify, and delete DNS records of the following types: A, AAAA, CNAME, NS, MX, TXT, SRV, and PTR.
- Allows you to add multiple A, AAAA, or PTR records at a time. DNS servers randomly respond to all DNS queries through round robin to achieve load balancing.
- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve global traffic scheduling.

Flexible integration with data centers

Apsara Stack DNS can forward enterprise domain names and provide the following services for you to flexibly build your network and cascade DNS servers with user-created DNS servers:

- Global default forwarding
- Forwarding queries for specific domain names

Internet access from enterprise servers

Apsara Stack DNS supports recursive resolution for Internet domain names, which allows your servers to access the Internet.

Tenant isolation is supported.

Apsara Stack DNS allows you to manage private zones in VPCs, resolve internal domain names, and isolate DNS records by tenant.

You can use Apsara Stack DNS to isolate data by VPC without the need to build your own DNS system. This helps reduce server and O&M costs.

GSLB

Global Server Load Balancer (GSLB) provides the following features on internal networks:

- Allows you to add multiple A, AAAA, or CNAME records at a time. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.
- Synchronizes configuration data for resolution among multiple clusters for which GSLB is activated. This feature is supported in multi-cloud scenarios.
- Supports address pool management to centrally manage enterprise applications by application service cluster.
- Supports custom global scheduling domains. You can centrally manage and code global scheduling instances based on your naming conventions.

Centralized management console

You can access DNS and other cloud services in the Apsara Stack Cloud Management (ASCM) console with one account. This provides the following benefits:

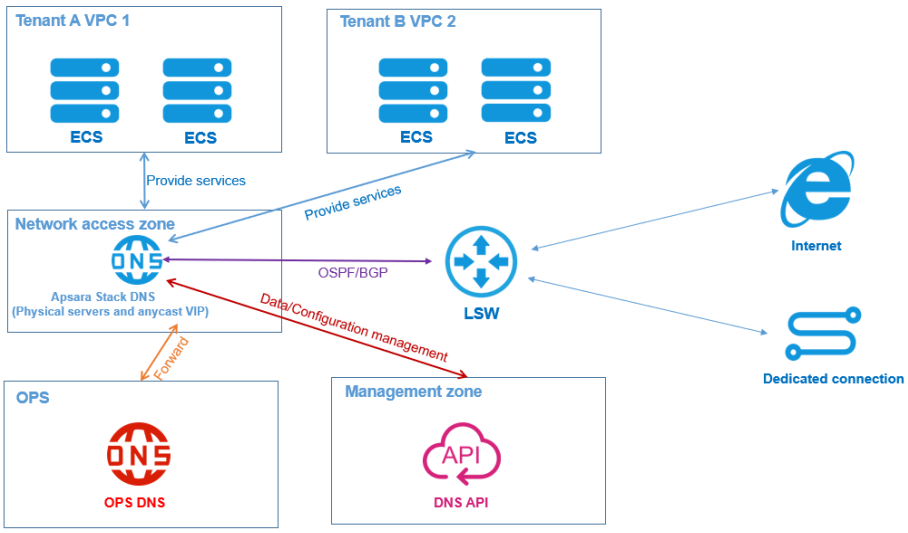
- Apsara Stack DNS supports web operations for data and service management, which facilitates your use of the DNS service.
- Apsara Stack DNS is deployed on clusters. You can add more clusters based on your needs.
- You can deploy Apsara Stack DNS in multiple zones. Apsara Stack DNS supports local active-active and zone-disaster recovery.
- Apsara Stack DNS is deployed in anycast mode, which delivers high availability and disaster recovery.

API operations

Apsara Stack DNS provides API operations so that you can integrate it with other systems.

22.3. Architecture

Architecture of Apsara Stack DNS



Architecture of Apsara Stack DNS (DNS Basic Edition and DNS Standard Edition)

- Uses two independent physical machines that are deployed in the network access zone to improve service availability. Apsara Stack DNS in this architecture can be scaled in or out.
- Issues anycast virtual IP address (VIP) routing requests over the LAN switch (LSW) by using Open Shortest Path First (OSPF) or Border Gateway Protocol (BGP). Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, performs recursive DNS queries for Internet domain names, allows you to add, modify, delete, and query authoritative domain names and forwarding domain names of private zones, and binds and unbinds a private zone to and from a VPC.

Architecture of Apsara Stack DNS (DNS Lightweight Edition)

- Supports the deployment with two physical machines on the OPS3 or OPS4 base, which eliminates the need to apply for an independent physical machine. The two physical machines achieve high availability. Apsara Stack DNS in this architecture cannot be scaled in or out.
- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants. The outbound IP address configured on the DNS servers can be used to forward requests to the OPS DNS server, Internet, or a dedicated enterprise network based on forwarding and recursive rules.
- Manages data and configurations by using APIs in the management zone.
- Allows you to create and query domain names on a web UI, forwards requests for cloud service domain names to the OPS DNS server, and performs recursive DNS queries for Internet domain names.

Architecture of Apsara Stack DNS (internal GTM Standard Edition)

- Depends on the deployment of DNS Basic Edition or DNS Standard Edition. Apsara Stack DNS of the internal GTM Standard Edition is deployed on the two physical machines of DNS Basic Edition or DNS

Standard Edition in the network access zone. Apsara Stack DNS in this architecture can be scaled in or out.

- Issues anycast VIP routing requests over the LSW by using OSPF or BGP. Anycast VIPs provide DNS services for VPCs and the classic network of tenants.
- Manages data and configurations by using APIs in the management zone.
- Allows you to manage domain names on a web UI, allows you to add, modify, delete, and query address pools, access policies, and scheduling instances. You can also create and delete Global Traffic Manager (GTM) synchronization clusters.

22.4. Features

1. Internal DNS resolution management

Internal DNS resolution management allows you to manage global internal domain names, global forwarding configurations, and global recursive resolution configurations that you have created in Apsara Stack. Changes to these configurations take effect on all VPCs and the classic network.

This feature provides the same global DNS resolution service to all servers in VPCs. DNS servers use anycast IP addresses within a region. This way, seamless service failover and failback can be achieved in a specific region where data centers support disaster recovery. Note: If you do not need to upgrade Apsara Stack DNS to the Standard Edition, you can configure DNS server addresses as global anycast IP addresses to implement seamless service failover and failback over the entire network if data centers support disaster recovery.

(1) Global internal domain names

Allows you to register, search, and delete global internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple DNS records of the A, AAAA, and PTR types on one host. By default, the resolution results include all the matching records. Records can be randomly rotated for load balancing.

Allows you to add multiple DNS records of the A, AAAA, and CNAME types on one host. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

(2) Global forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS forwards requests with or without recursion.

- Forward All Requests (without Recursion): Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.
- Forward All Requests (with Recursion): The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

(3) Global recursive configurations

Supports recursive resolution for Internet domain names, which enables your servers to access the Internet.

Allows you to enable, disable, or modify the global default forwarding configurations.

2 PrivateZone (DNS Standard Edition only)

The PrivateZone feature allows you to create tenant-specific domain names in VPCs. You can bind and unbind the domain names to and from VPCs as needed to isolate tenants. Changes to these configurations take effect only in the VPCs to which the domain names are bound.

This feature provides personalized DNS resolution service to servers in the VPCs to which the domain names are bound. DNS servers use anycast IP addresses within a region. This way, seamless service failover and failback can be achieved in a specific region where data centers support disaster recovery.

(1) Tenant internal domain names

Allows you to register, search, and delete tenant internal domain names and add descriptions for these domain names. You can also add, delete, and modify DNS records. The following DNS record types are supported: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.

Allows you to add multiple DNS records of the A, AAAA, and PTR types on one host. By default, the resolution results include all the matching records. Records can be randomly rotated for load balancing.

Allows you to add multiple DNS records of the A, AAAA, and CNAME types on one host. DNS servers respond to DNS queries based on the weight of each record type to achieve load balancing.

Allows you to bind and unbind a domain name to and from a VPC.

(2) Tenant forwarding configurations

Forwards domain name requests to another DNS server for resolution.

Supports global default forwarding, which forwards requests of domain names that do not have forwarding configurations to another DNS server for resolution.

Apsara Stack DNS can forward requests with or without recursion.

- Forward All Requests (without Recursion): Only the specified DNS server is used to resolve domain names. If the resolution fails or the request times out, a message is returned to the DNS client to indicate that the query failed.
- Forward All Requests (with Recursion): The specified DNS server is preferentially used to resolve domain names. If the resolution fails, the local DNS server is used.

Allows you to bind and unbind a domain name to and from a VPC.

3 Internal Global Traffic Manager (internal GTM Standard Edition only)

Internal Global Traffic Manager (GTM) provides multi-cloud disaster recovery for your domain names. You can connect your domain names to an internal GTM instance to manage traffic loads between Apsara Stack systems.

Internal GTM supports internal Global Server Load Balancer (GSLB). This feature intelligently allocates IP addresses for DNS queries from request sources based on configured scheduling policies. It also supports multi-cloud, hybrid deployment and configuration data synchronization between cloud networks.

(1) Scheduling instance management

Allows you to manage scheduling instances. Each scheduling instance corresponds to an application instance.

Allows you to manage address pools. Each address pool corresponds to a service cluster of an application instance.

Allows you to manage scheduling domains and set the scheduling domains to which scheduling instances belong. You can centrally manage and code global scheduling instances based on your own naming conventions.

(2) Data synchronization management

Allows you to manage global data synchronization links. You can create data synchronization links, manage data synchronization configurations, and view data synchronization information of multiple internal GTM services. The information includes local system information, information of cluster nodes on which data synchronization relationship has been established, and primary and secondary relationships.

Allows you to manage the messages for changes to data synchronization links, which helps you confirm request messages for primary nodes to actively add secondary nodes.

23.Log Service

23.1. What is Log Service?

23.1.1. Overview

Log Service is a unified solution for high volumes of log data, and provides log data collection, subscription, query, and transfer functions.

- Real-time collection and consumption: Log Service collects log data in real time from multiple channels through the client, APIs, tracking.js, and libraries. Data can be immediately subscribed and read after it is written. Interfaces such as Spark Streaming, Storm, and Consumer Library can be used to process data in real time.
- LogSearch: LogSearch creates indexes for log data in real time and provides real-time and powerful storage and query engines. LogSearch allows you to retrieve logs by various dimensions such as time, keyword, and context.

Log Service can automatically scale based on processing requirements. It can scale out to handle large volumes (PBs) of data.

23.1.2. Values

Log Service helps you build solutions for large volumes of log data.

Log Service is applicable to the following scenarios: data collection, real-time computing, data warehousing and offline analysis, product operation and analysis, operations and maintenance, and management.

- Data collection and consumption
- ETL and stream processing
- Event sourcing and tracing
- Log management

23.2. Benefits

23.2.1. Features

This topic describes the features of Log Service.

Real-time log collection

Log Service supports real-time log collection. You can collect data in real time by using the following methods.

- Log collection by using Logtail: stable, reliable, and secure. This method provides high performance at low resource consumption. You can use Logtail to collect logs from servers that run Linux or Windows and from Docker containers.
- Log collection by using APIs or SDKs: flexible, convenient, and scalable. You can use an API or SDK developed in multiple programming languages to collect logs from mobile devices.
- Log collection by integrating Log Service with cloud services: convenient and efficient. You can

integrate Log Service with cloud services such as Elastic Compute Service (ECS) and then collect logs from the cloud services.

- Other log collection methods: Syslog and Logstash.

Real-time log consumption

Log Service allows you to use stream computing systems to consume data. Log Service provides consumer libraries that are developed in multiple programming languages for data consumption.

- Comprehensive features: Log Service provides all of the features of Kafka. Log Service records consumption checkpoints and scales computing resources based on the volume of data reads. Log Service also allows you to specify a time range to consume data.
- Stability and reliability: After data is written to Log Service, it can be consumed in real time. Data is stored in duplicates in Log Service. Computing resources are scalable based on the volume of data reads.
- Easy to use: You can use Spark Streaming, Storm and SDKs to consume data. Log Service provides consumer libraries that allow you to consume data in load balancing mode.

Log query

Log Service allows you to create indexes for log data and query the data in real time. You can specify a time range and query log data by keyword.

- Large scale: Log Service supports real-time indexing for petabytes of data.
- Flexible queries: Log Service supports keyword-based queries, fuzzy matching, cross-topic queries, and contextual queries.

23.2.2. Benefits

This topic describes the benefits of Log Service.

Fully managed service

- Log Service is easy to access and use.
- LogHub provides all the features of Kafka. You can store and query functional data such as monitoring and alerting data in LogHub. The data that you store or query per day can amount to petabytes.
- LogSearch/Analytics allows you to query log data, view log data on dashboards, and configure alerts.
- Log Service allows you to import data from more than 30 data sources such as the open-source software applications Storm and Spark Streaming.

Inclusive ecosystem

- The 30-odd data sources include embedded devices, web pages, servers, and programs. LogHub can also be interconnected with consumption systems such as , Storm, and Spark Streaming.
- LogSearch/Analytics provides complete query syntax and supports connection with Grafana based on the SQL-92 and JDBC protocols.

Real-time response

- LogHub: Data can be consumed immediately after being written to Log Service. Logtail acts as an agent to collect and send data to Log Service in real time.

- LogSearch/Analytics: Data can be queried immediately after being written to Log Service. If you specify multiple query conditions, data can be returned within seconds.

Complete operations of the API and SDKs

- Log Service supports custom management and secondary development.
- All Log Service features can be implemented by using SDKs or the API. SDKs in multiple programming languages are provided. You can use an SDK to manage services and millions of devices.
- The query syntax is simple and compatible with the SQL-92 standard. User-friendly interfaces are integrated into the software environment.

23.3. Architecture

23.3.1. Components

This topic describes the components of Log Service.

Logtail

Logtail is an agent that collects logs. It has the following characteristics:

- Non-intrusive file-based log collection
  - Logtail only reads log files.
  - Log collection is not intrusive.
- High security and reliability
  - Logtail can rotate log files without data loss.
  - Data that is collected by using Logtail can be locally cached.
  - Logtail retries log collection if network exceptions occur.
- Ease of use and management
  - You can configure log collection by using Logtail on the web.
  - You can configure log collection by using Logtail in the Log Service console.
- Complete self-protection mechanism
  - Logtail monitors the CPU and memory resources that it consumes in real time.
  - Logtail allows you to set an upper limit on the resources that it consumes.

Frontend servers

Frontend servers are built on the Linux Virtual Server (LVS) and NGINX servers. They have the following features:

- Support for HTTP and REST
- Scale-out

  More frontend servers can be deployed to accommodate traffic surges.
- High throughput and low latency
  - Requests are asynchronously processed. If an exception occurs when a single request is sent, other requests are not affected.

- Log data is compressed by using the LZ4 algorithm: This reduces required network bandwidth resources and increases the processing capabilities of individual servers.

Backend servers

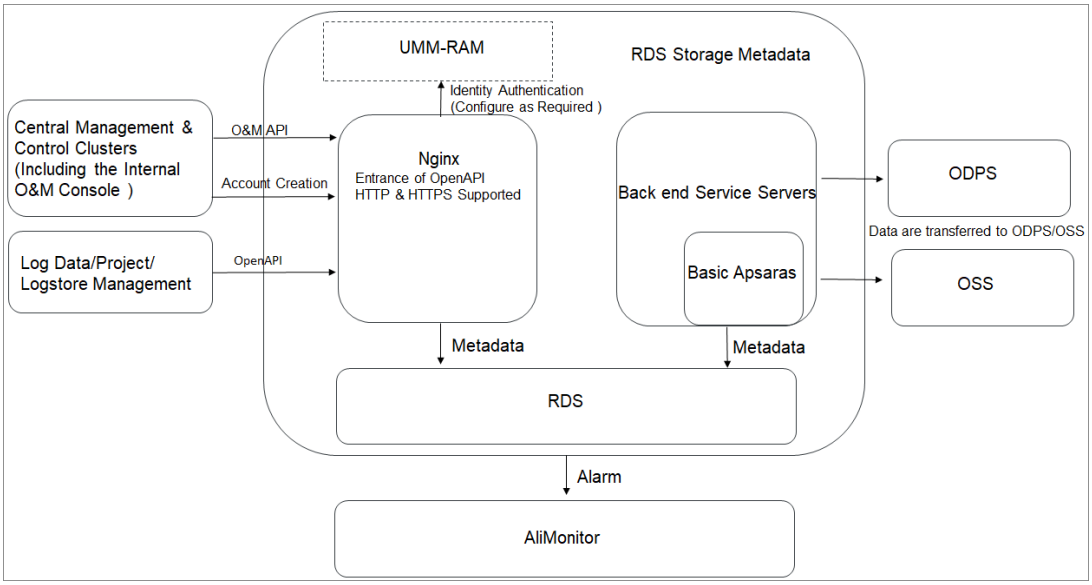
Backend processes are deployed across multiple servers. Backend servers store, index, and query data in real time. The following list describes the features of backend servers.

- High data security
  - Each log entry is stored in three copies.
  - Data is automatically recovered in the cases of disk damage or server downtime.
- Stable service
  - Logstores are automatically migrated in case of process crashes or server downtime.
  - Automatic load balancing ensures that traffic is distributed evenly among servers.
  - Strict quotas prevent some unexpected or incorrect operations of a single user from affecting other users.
- Scale-out
  - A shard is the basic unit for scale-out.
  - You can add shards to increase throughput based on your requirements.

23.3.2. System architecture

The following figure shows the architecture of Log Service.

Architecture



- The console and open APIs are located on the left side. They are used to interact with external modules.
- The core modules in the middle include:
  - UMM and RAM: account management
  - RDS: metadata storage
  - NGINX: front-end servers

- Log Service background: back-end business servers

24.API Gateway

24.1. What is API Gateway?

API Gateway provides a comprehensive suite of API hosting services that help you share capabilities, services, and data with partners in the form of APIs.

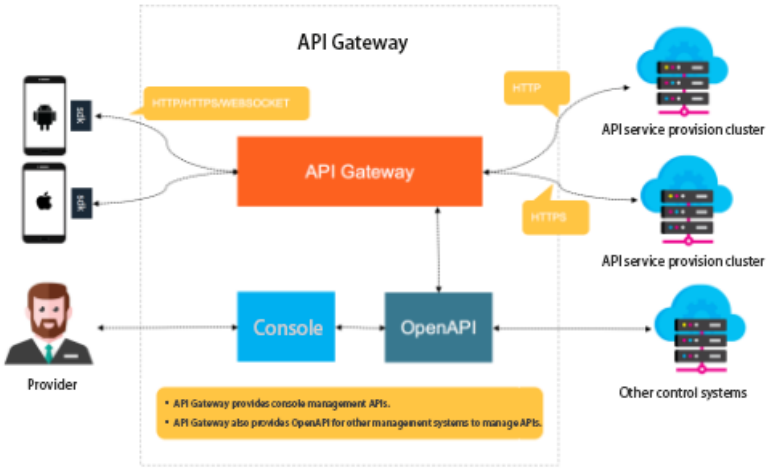
- API Gateway provides multiple security mechanisms to secure APIs and reduce the risks introduced by open APIs. These mechanisms include protection against replay attacks, request encryption, identity authentication, permission management, and throttling.
- API Gateway provides API lifecycle management that allows you to create, publish, and unpublish APIs, and improve API management and iteration efficiency.

API Gateway allows enterprises to reuse and share their capabilities with each other so that they can focus on their core business.

API Gateway

24.2. System architecture

System architecture of API Gateway



API Gateway consists of the following components:

- Gateway

  The gateway component is the core system that implements all business logic.

The gateway component supports multi-protocol access for all clients, including HTTP, HTTPS, and WebSocket. The gateway component manages client connections, throttles API requests, and implements IP address-based access control.

The gateway component loads user-defined APIs into the memory, processes requests from clients based on API definitions, calls back-end APIs, and returns back-end responses to clients.

- OpenAPI

  The OpenAPI component consists of a group of standard management APIs provided by API Gateway to manage API definitions. You can use the OpenAPI component to manage groups, metadata, and authorization for APIs. When the OpenAPI component receives an API change request, it synchronizes the change to all gateway services. System administrators can use the management APIs to manage the APIs that are running in the gateway in real time.

  System administrators can manage their own APIs in the API Gateway console in real time. They can also call the management APIs in their own management systems to manage their own APIs.

- Console

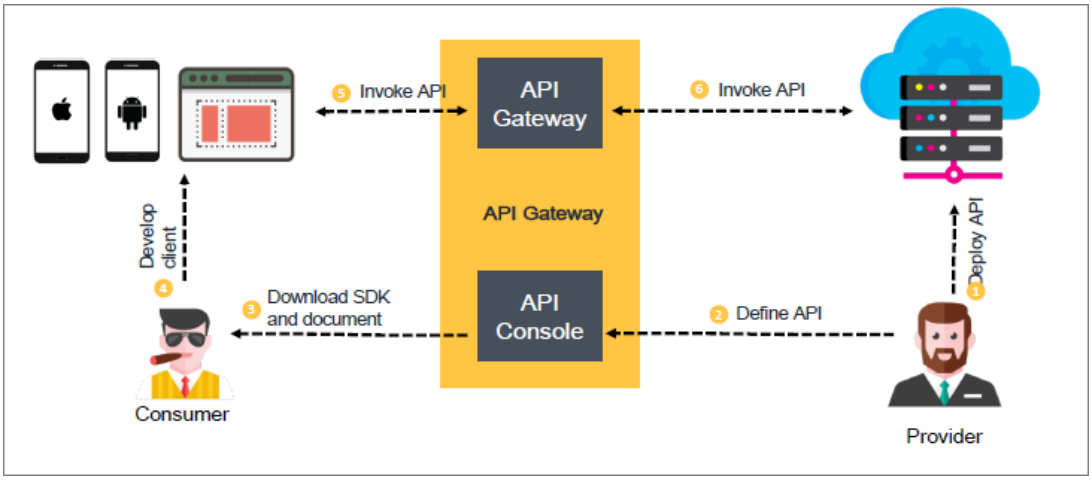
  The API Gateway console implements all features of API Gateway. System administrators can manage their own APIs in the console in real time.

  The API Gateway console provides you with a graphical user interface to call APIs through the OpenAPI component.

24.3. Features

24.3.1. API lifecycle management

API lifecycle management

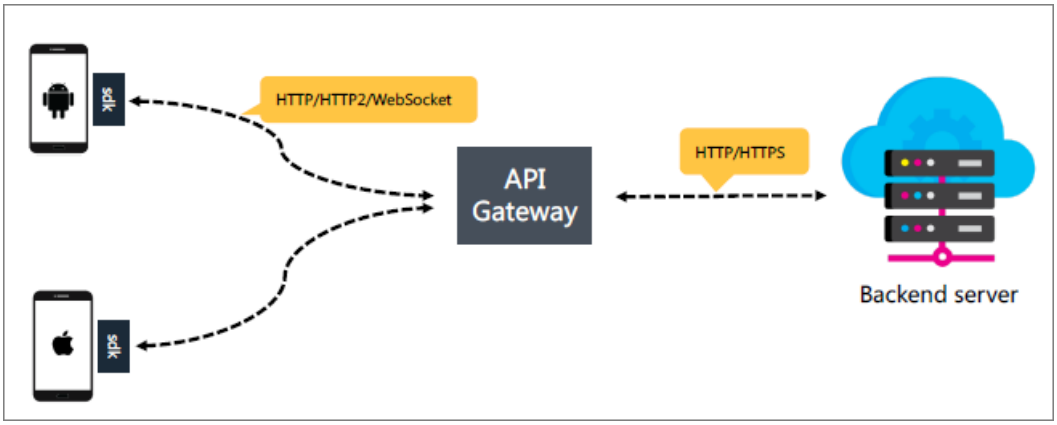


It provides a range of lifecycle management functions to publish, test, and unpublish APIs.

It provides maintenance functions such as routine API management, API version management, and quick API rollback.

24.3.2. Multi-protocol access

Multi-protocol access



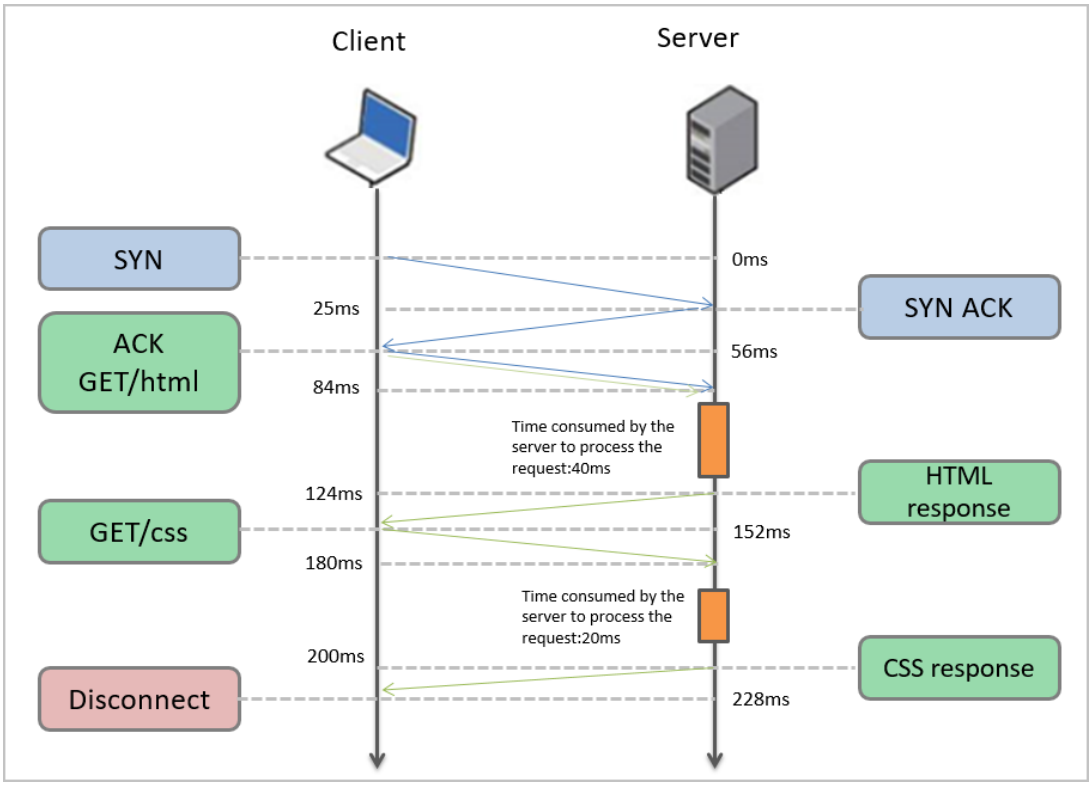
The following protocols can be selected to establish bidirectional communication between clients and API Gateway:

- HTTP: is the most popular Internet text protocol.
- HTTP/2: supports multiplexing and header compression for high efficiency.
- WebSocket: supports persistent connections for binary communications.

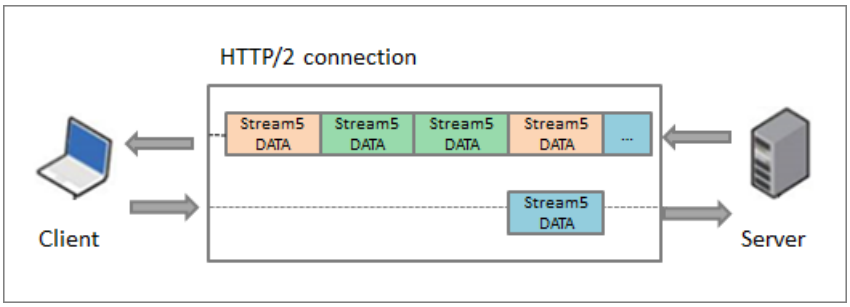
Unlike HTTP/1.x, all HTTP/2 communication is split into smaller messages and frames, each of which is encapsulated with binary encoding. In HTTP/1.x, the header information is encapsulated in the Headers frame, and the request body is encapsulated in the Data frame. A single TCP connection can be used to send multiple requests. This reduces connections to the server and improves throughput. HTTP/2 uses header compression to enable faster data transmission and deliver more benefits to the mobile network environment so that network congestion is reduced.

Browsers limit the number of HTTP/1.x connections with the same domain name at the same time. If the limit is exceeded, further requests will be blocked. The binary framing layer in HTTP/2 enables full request and response multiplexing within a shared TCP connection. An HTTP message is divided into independent frames that are interleaved and reassembled on the other end based on stream identifiers and headers. The following figures compare data transmission between HTTP/1.x and HTTP/2.

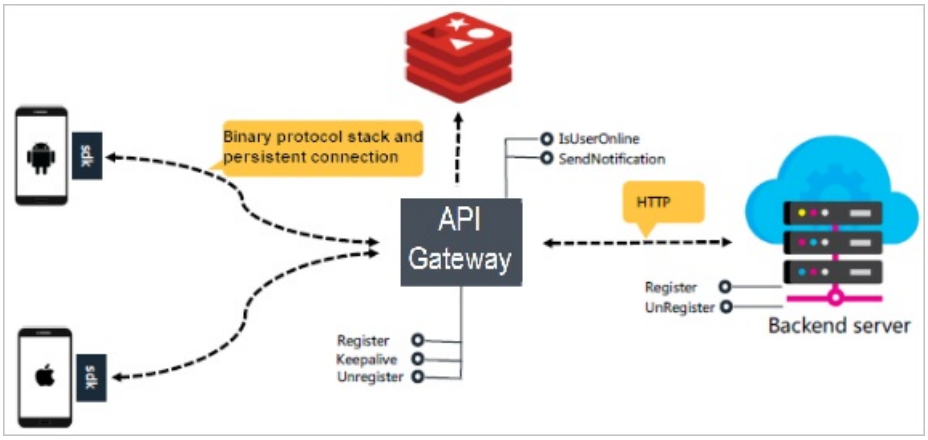
HTTP/1.x



HTTP/2



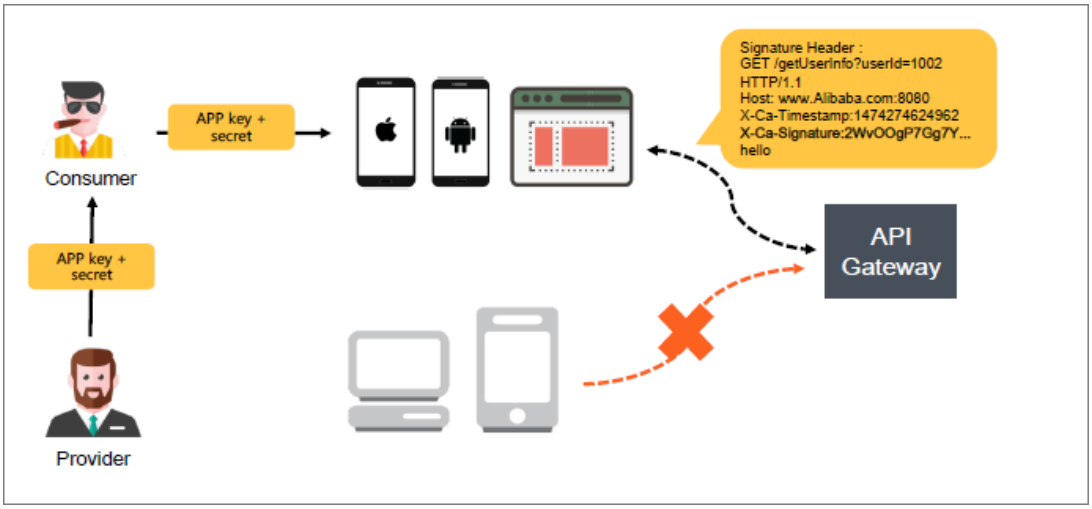
WebSocket is a protocol that enables full-duplex persistent communication. Both clients and servers can send and receive data to and from each other. HTTP is used during the handshake. After a successful handshake, clients and servers can directly communicate with each other without HTTP. The data format is lightweight, the performance overheads are low, and the communication efficiency is high. Clients can communicate with any servers without the same-origin policy.



API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications. API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

24.3.3. Application access control

Application access control



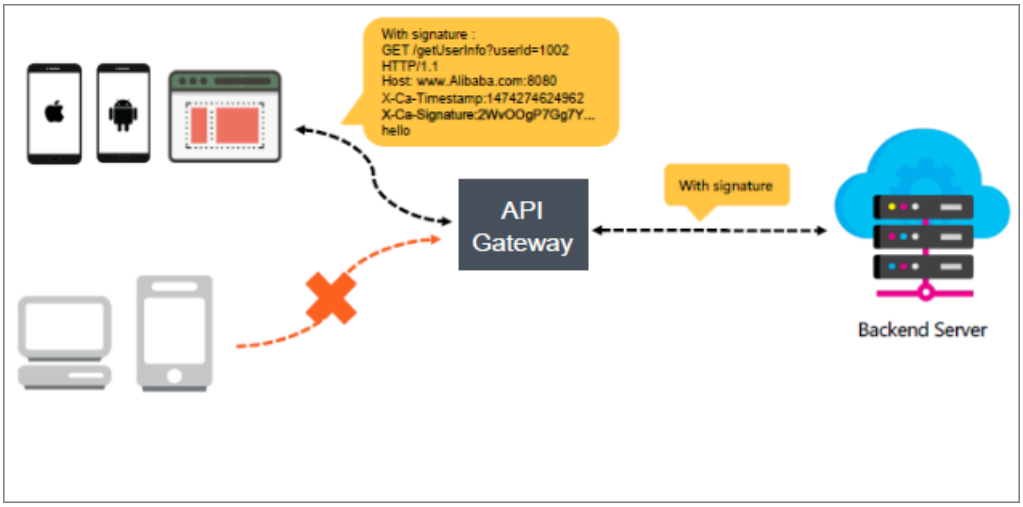
API Gateway provides an application-based authentication mechanism. This mechanism ensures that only authorized clients can send requests to the back-end services. Applications are the identities that you use to call APIs. Each application has a key pair that consists of an AppKey and an AppSecret. The AppKey parameter is added to the request header when a client sends a request, while the AppSecret is used to calculate the signature. Signature verification can protect user data from being tampered. If the verification fails, an error is reported immediately.

Symmetric encryption is used to verify the signature. The construction rules of the signature string are described in public documentation. HMAC-SHA256 is used as the signature algorithm. AppSecret is used as the encryption key that is only available to the clients and API Gateway. The client constructs and encrypts StringToSign based on specific rules. For more information about the construction methods, see Request signatures. After receiving the request, API Gateway constructs the StringToSign based on request parameters. Then, API Gateway finds the corresponding AppSecret through the AppKey. API Gateway calculates the signature string and compares it with the signature string that is sent by the user. If both signature strings are same, the signature verification is passed. Otherwise, it fails.

The principle of application-based authentication is as follows: API Gateway obtains the unique AppID based on the AppKey, and checks whether the application is authorized to access the API based on the AppID and API. If yes, access to the API is allowed. Otherwise, an unauthorized access error is reported.

24.3.4. Full-link signature verification mechanism

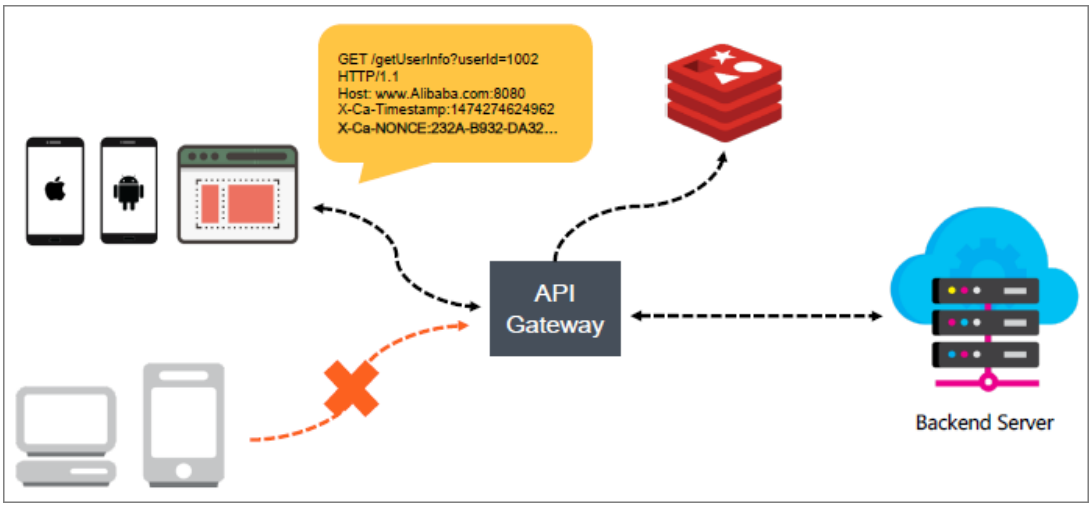
Full-link signature verification mechanism



API Gateway provides a full-link signature verification mechanism for communication between the client and API Gateway or between API Gateway and the backend service. This mechanism prevents data tampering during request transmission. When a client calls an API, the client must convert the key request data into a signature string based on API Gateway signature algorithms. The client must attach the signature string to the request header. API Gateway performs symmetric calculation to parse the signature and verify the identity of the request sender. HTTP, HTTPS, and WebSocket requests must have a signature in their header.

24.3.5. Anti-replay mechanism

Anti-replay mechanism



API Gateway provides an anti-replay mechanism to protect against data tampering used in replay attacks.

- When a client sends a request to API Gateway, the X-Ca-Nonce header is added. The value of the X-Ca-Nonce header can be any string. API Gateway verifies whether the same X-Ca-Nonce header has been passed in within 15 minutes. If yes, the request is considered a replay, and API Gateway reports an error immediately.

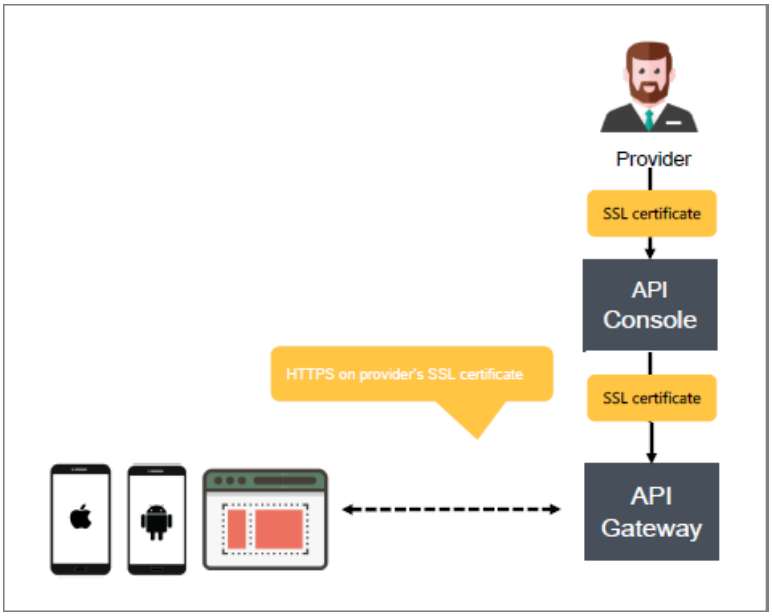
  A distributed cache is used. API Gateway verifies whether the same X-Ca-Nonce header exists for each request.

- The value of the Nonce parameter is included in the signature string. Therefore, it cannot be

tampered.

24.3.6. HTTPS communication based on the SSL certificate of the user

HTTPS communication based on the SSL certificate of the user

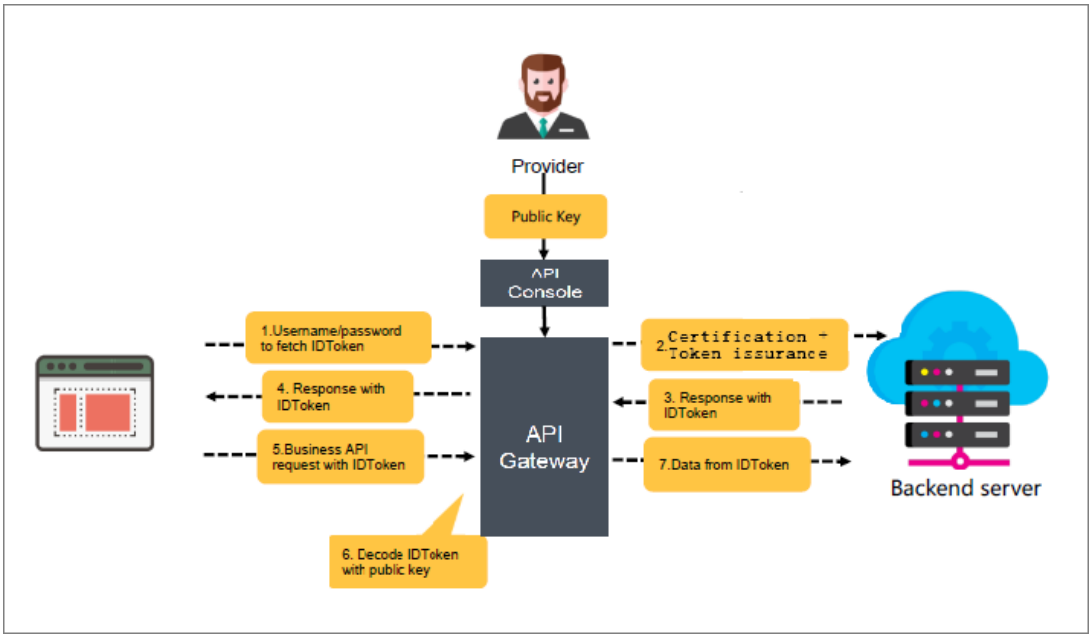


A system administrator can upload an SSL certificate corresponding to the domain name in the API Gateway console. Data transmitted between clients and API Gateway will then be encrypted based on the certificate. This prevents data tampering during transmission.

System administrators can update SSL certificates in real time in the API Gateway console.

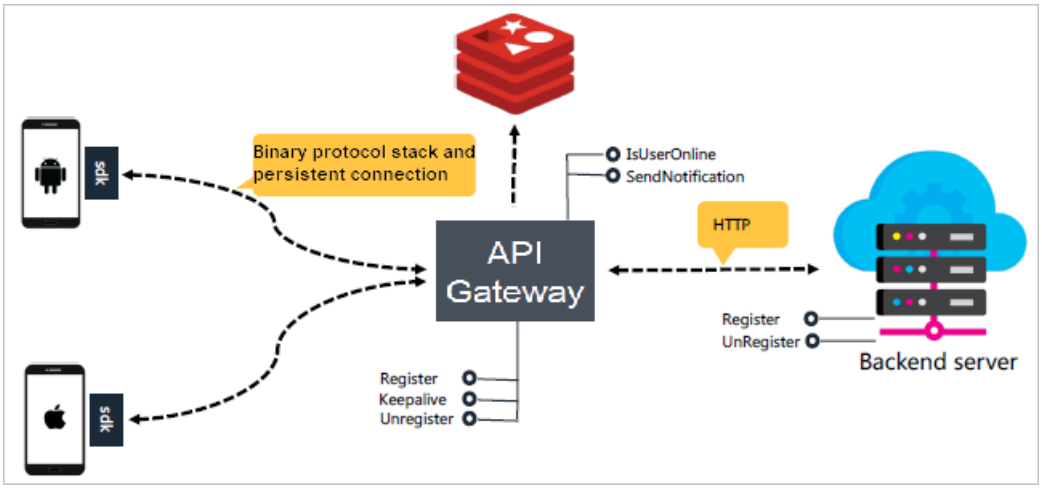
24.3.7. Support for OpenID Connect

Support for OpenID Connect



API Gateway supports OpenID Connect authentication, allowing API providers to verify requests based on their own user systems. OpenID Connect is a lightweight authentication standard based on OAuth 2.0. It provides a framework for identity interaction through APIs. Compared with OAuth, OpenID Connect not only authenticates a request, but also specifies the identity of the requester.

24.3.8. Bidirectional communication

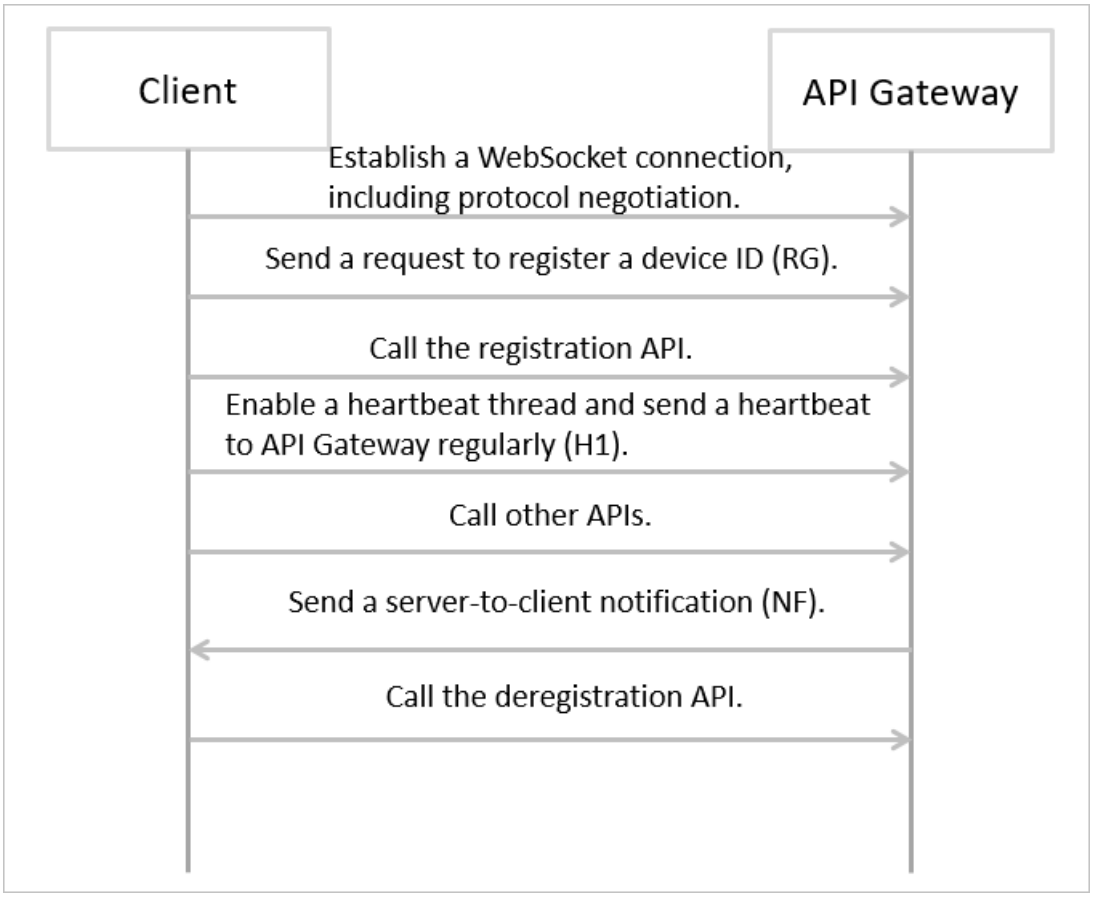


API Gateway supports bidirectional communication and maintains persistent connections between clients and itself. API Gateway can update the online status of clients after receiving heartbeat requests. Back-end services can access the API Gateway interface to query the online status of clients and push in-application notifications.

API Gateway implements bidirectional communication based on the WebSocket protocol. Bidirectional communication is supported by Android SDKs, Objective-C SDKs, and Java SDKs.

API Gateway provides built-in APIs to WebSocket users, including the APIs that clients use to register and deregister device IDs with API Gateway, and the APIs that are called to detect heartbeats.

Before establishing a WebSocket connection between clients and API Gateway, you must call the registration API to register device IDs. The following figure shows the interaction between clients and API Gateway.

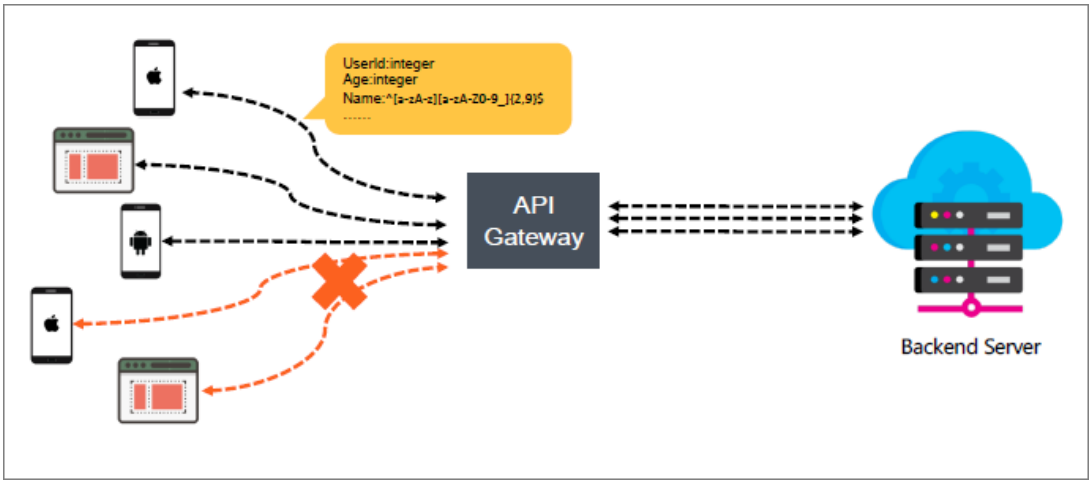


Clients need to complete the following operations:

1. Establish a WebSocket connection, including protocol negotiation.
2. Send a request to register a device ID (RG).
3. Call the registration API.
4. Enable a heartbeat thread and send a heartbeat to API Gateway regularly (H1).
5. Call other APIs.
6. Receive notifications sent by API Gateway.
7. Call the deregistration API.

24.3.9. Parameter cleaning

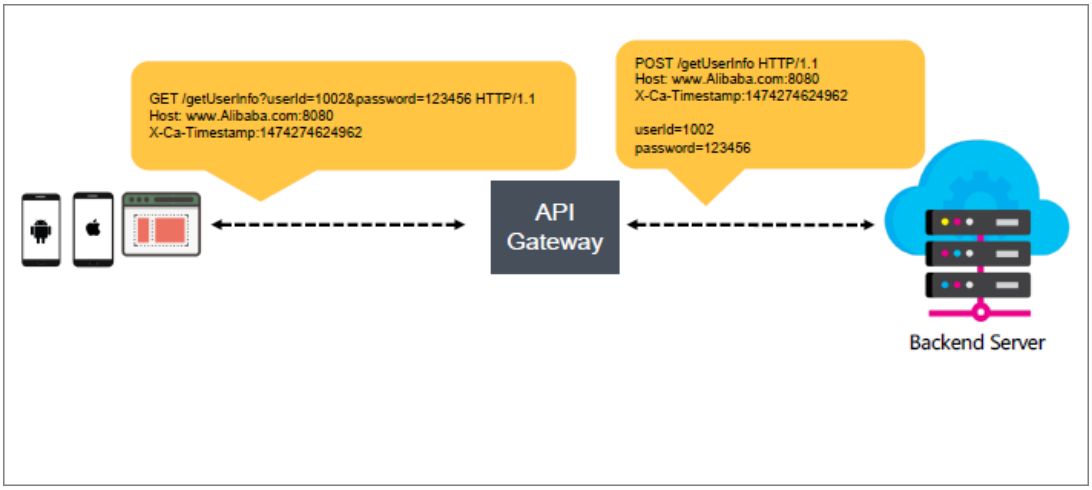
Parameter cleaning



System administrators can define the data type, regular expression, and enumeration of all API parameters. API Gateway forwards API requests that match the API definition to the backend service, while rejecting the requests that do not match the definition. This ensures that the backend service only receives standard requests that match API definitions.

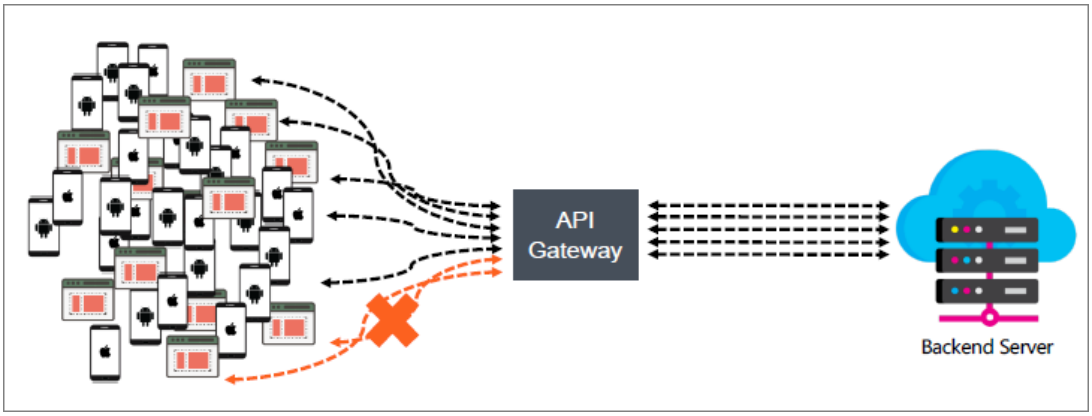
24.3.10. Mappings between frontend and backend parameters

Mappings between frontend and backend parameters



API Gateway provides parameter mapping capabilities to relocate parameters within a request before sending the request to the backend service. For example, a parameter in a request sent to API Gateway is defined in Query. API Gateway can map the parameter to Form and then send the request to the backend service. This function ensures that users can access complicated backend functions by calling well-organized APIs.

24.3.11. Throttling

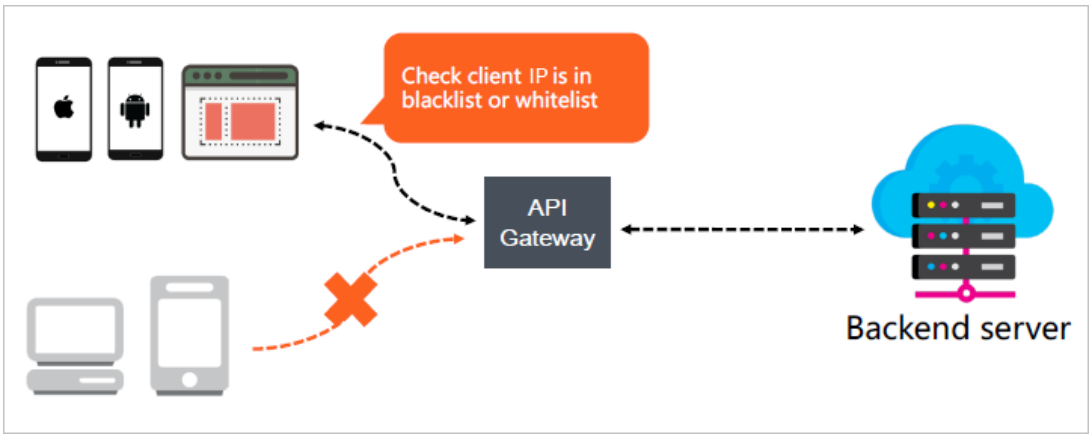


System administrators can set a request threshold based on the maximum processing capabilities of backend services. If the total number of requests exceeds the threshold, API Gateway will reject the excess requests to protect backend services from being overloaded. Supported dimensions include API, user, and application. Supported time granularity includes second, minute, hour, and day.

API Gateway uses a distributed cache, calculates the number of API requests from clients, and customizes keys in different formats based on the unit of time that you set. For example, if you set the unit of time to minute, the key is displayed in the yyyyMMddHHmm format. If the current time is 20:00 on May 7, 2019, the key is displayed as 201905072000. All requests in the current period are accumulated for this key. Requests will be recounted automatically during the next period. When another request is received, counting will be restarted.

24.3.12. IP address-based access control

IP address-based access control

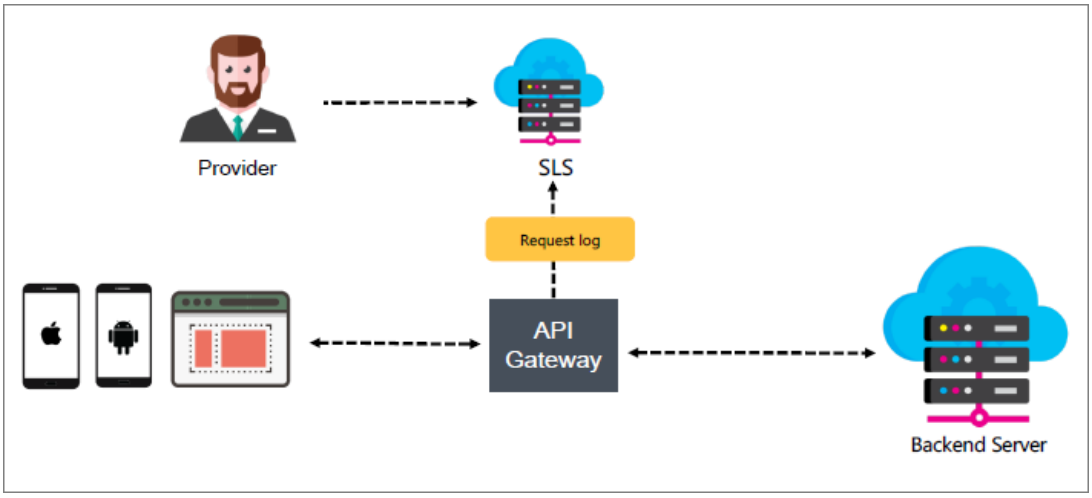


IP address-based access control is one of the API security protection measures provided by API Gateway. This measure controls the source IP addresses or IP address segments for API requests. System administrators can configure an IP whitelist or blacklist for an API to allow or deny an API request from an IP address.

The IP addresses obtained by API Gateway are egress IP addresses of clients. You cannot use the X-Forward-For header because its value can be randomly set by clients. API Gateway compares these client IP addresses with user-defined rules. It allows access to APIs from IP addresses in the whitelist and denies access to APIs from IP addresses in the blacklist.

24.3.13. Log analysis

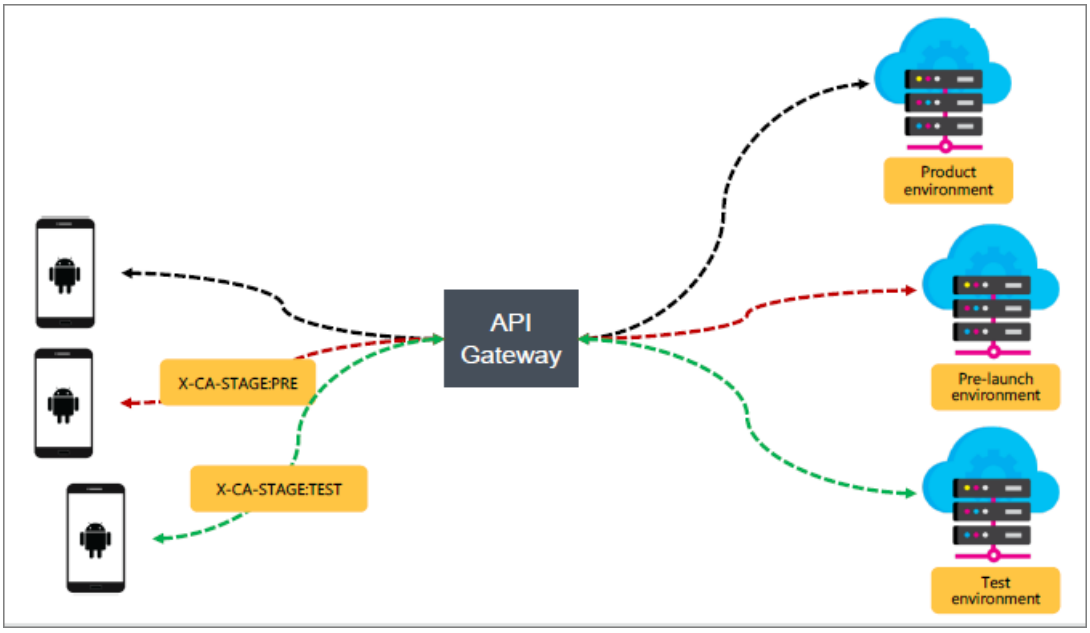
Log analysis



API Gateway sends called logs to Log Service. System administrators can use Log Service to query or download logs, or perform statistical analysis in real time. Logs can also be sent to OSS or MaxCompute.

24.3.14. Publish an API in multiple environments

Publish an API to multiple environments

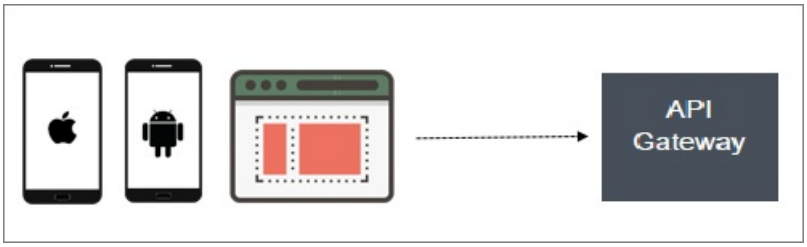


API Gateway allows you to publish an API group in three different environments: test, pre-release, and release environments. The test and pre-release environments are used by testers to test or debug APIs. The release environment is where the APIs can be used.

You can use the environment management function for API groups to set environment parameters for the test, pre-release, and online environments. The environment parameter is a common constant that can be customized for each environment. When you call an API, you can place the environment parameter in any location of the request. API Gateway identifies the environment based on the environment parameter in your request.

24.3.15. Mock mode

Mock mode



A project is typically developed by multiple partners working together toward a specific goal. The interdependence among various stakeholders often restricts individual members during the process, and misunderstandings may affect the development process or even delay the project schedule. You can mock expected responses to be returned to API callers during the project development process. This can greatly reduce misunderstanding among partners and greatly improve the development efficiency.

API Gateway supports the Mock mode.

24.4. Benefits

- **Easy maintenance**

  After you register APIs in API Gateway, API Gateway handles all the API management issues such as documentation maintenance, version management of APIs, and SDK maintenance. This significantly reduces routine maintenance costs.

- **High performance**

  API Gateway maintains persistent connections between clients and API Gateway itself by supporting HTTP/2 and WebSocket. Both HTTP/2 and WebSocket are efficient binary protocols.

  API Gateway uses a distributed deployment and scales out automatically to handle a large number of API requests with low latency. API Gateway offers reliable and efficient features for your back-end services.

- **Stability**

  API Gateway has provided services to Alibaba Cloud public cloud users for over two years, and has a proven track record of performance. API Gateway provides stable services even in uncommon cases such as when over-sized packets are received, or when back-end services are unstable and slow to respond.

- **Security**

  API Gateway implements SSL encryption in the full link of communication to protect all data against eavesdropping during transmission.

  API Gateway implements signature verification in the full link of communication to prevent data tampering during transmission.

API Gateway enables strict authorization management, anti-replay mechanism, parameter cleaning, IP address-based access control, and precise throttling.This ensures secure, stable and controllable services.

25.Enterprise Distributed Application Service (EDAS)

25.1. What is EDAS?

Enterprise Distributed Application Service (EDAS) is a Platform as a Service (PaaS) platform for application hosting and microservice management, providing full-stack solutions such as application development, deployment, monitoring, and O&M. It supports Dubbo, Spring Cloud, and other microservice runtime environments, helping you easily migrate applications to the cloud.

Diverse application hosting environments

You can select instance-exclusive Elastic Compute Service (ECS) clusters, Container Service Kubernetes clusters, and user-created Kubernetes clusters based on your application systems and resource needs.

Abundant microservice frameworks

You can develop applications and services in the native Dubbo, native Spring Cloud, and High-Speed Service Framework (HSF) frameworks, and host the developed applications and services to EDAS.

- You can host Dubbo and Spring Cloud applications to EDAS by adding dependencies and modifying a few configurations. You have access to the features of EDAS, such as enterprise-level application hosting, service governance, monitoring and alerting, and application diagnosis, without having to build ZooKeeper, Eureka, and Consul. This lowers the costs of deployment and O&M.
- HSF is the distributed remote procedure call (RPC) framework that is widely used within Alibaba Group. It interconnects different service systems and decouples inter-system implementation dependencies. HSF unifies the service publishing and call methods for distributed applications to help you conveniently and quickly develop distributed applications. HSF provides or uses common functional modules, and frees developers from various complex technical details involved in distributed architectures, such as remote communication, serialization, performance loss, and the implementation of synchronous and asynchronous calls.

Comprehensive application management

You can perform end-to-end management, service governance, and microservice management for your applications in the EDAS console.

- Application lifecycle management

  EDAS provides end-to-end application management, allowing you to deploy, scale out, scale in, stop, and delete applications. Applications of all sizes can be managed in the EDAS console.

- Service governance

  EDAS integrates a wide variety of service governance components, such as auto scaling, throttling and degradation, and health check, to deal with unexpected traffic spikes and crashes caused by dependencies. This greatly improves platform stability.

- Microservice management

  EDAS provides the service topology, service report, and trace query features to help you manage every component and service in a distributed system.

Comprehensive monitoring and diagnosis

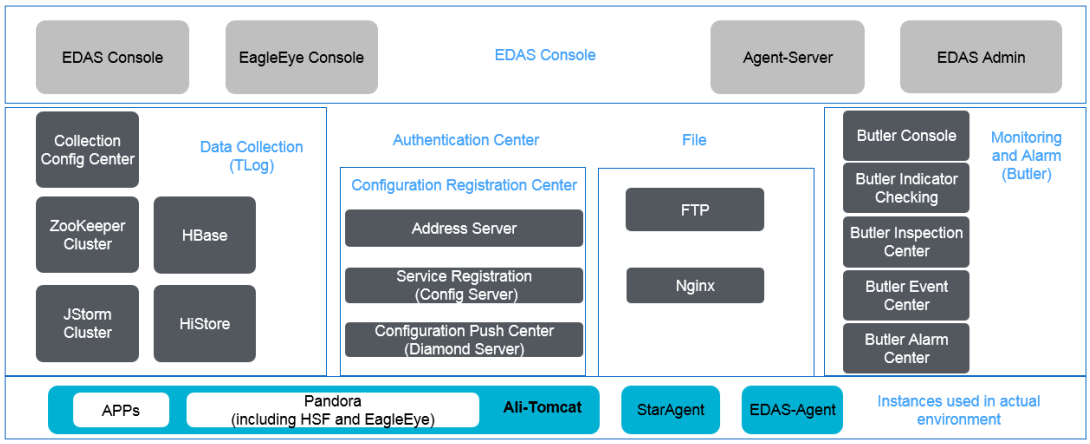
You can monitor the status of resources and services in applications in the EDAS console to promptly identify problems and quickly locate their causes through the logging and diagnosis components.

EDAS is connected to the Application Real-Time Monitoring Service (ARMS) to monitor the health status of application resources and services at the Infrastructure as a Service (IaaS) layer in real time, helping you quickly locate problems.

25.2. Architecture

EDAS consists of the console, data collection system, configuration registry, and authentication center. EDAS architecture shows the EDAS architecture.

EDAS architecture



- EDAS console

  It is a GUI where you can directly use EDAS system functions. In the console, you can implement resource management, application lifecycle management, O&M control, service governance, three-dimensional monitoring, and digital operations.

- Data collection system

  It collects trace logs and the runtime statuses of EDAS clusters and all customer application instances, and summarizes, computes, and stores data in real time.

- Configuration registry

  It is a central server used to publish and subscribe to HSF services (RPC framework) and push distributed configurations.

- Authentication center

  It controls permissions for user data to ensure data security.

- O&M system

  It is a major tool of EDAS for daily monitoring and alarms of all EDAS components.

- Command channel system

  It is a control center that remotely sends commands to application instances.

- File system

  It stores WAR packages and required components, such as JDK and Ali-Tomcat, uploaded by users.

25.3. Features and principles

25.3.1. Full compatibility with Apache Tomcat containers

As the basic container for running EDAS applications, EDAS containers integrate with the Alibaba middleware technology stack to greatly improve the startup, monitoring, stability, and performance of containers. Also, EDAS containers are fully compatible with Apache Tomcat.

25.3.2. Application-centric PaaS platform

Application management and O&M

In the visual EDAS console, you can perform application lifecycle management tasks in a single place, including creating, deploying, starting, stopping, scaling out, scaling in, and deleting applications, thereby implementing full-process application management. With Alibaba's rich experience in O&M of ultra-large scale clusters, EDAS allows you to easily operate and maintain an application that has thousands of instances.

Auto scaling

EDAS supports scaling out and in applications both manually and automatically. With real-time monitoring of CPU, memory, and workload, you can scale out and in your applications in seconds.

Primary account and RAM users

EDAS provides a unique primary Alibaba Cloud account and RAM user system that allows you to establish primary account and RAM user relationships on the EDAS platform based on the organization of your enterprise's departments, teams, and projects. In addition, ECS resources are organized based on primary account and RAM user relationships to simplify resource allocation.

Role and permission control

The maintenance of applications normally involves application development owners, application maintenance owners, and instance owners. Considering that different roles perform different management activities on an application, EDAS provides a role and permission control mechanism that allows you to define roles and assign permissions for different accounts.

25.3.3. Rich distributed services

Distributed service framework

Since 2007, as the e-commerce platforms of the Alibaba Group continuously developed to distributed architectures, the self-developed distributed service framework High Speed Framework (HSF) and Dubbo came into being. Built on a high-performance network communication framework, HSF is a distributed service framework for enterprise Internet architectures. It provides proven features, such as service publishing, registration, calling, routing, authentication, throttling, degradation, and distributed tracing.

Distributed configuration management

The transformation from a centralized system to a distributed system makes it a challenge to manage the configuration information on each instance of the distributed system in real time. EDAS provides efficient distributed configuration management that allows you to centrally manage all configuration information across the distributed system in the EDAS console. More importantly, EDAS allows you to modify the configuration information in the console and notifies all the instances of this modification in seconds.

Distributed task scheduling

SchedulerX, a task scheduling service integrated in EDAS, allows you to configure any single-instance or distributed tasks for periodic scheduling. It also provides the ability to manage the running periods and query the running history of the tasks. It applies to task scheduling scenarios such as migrating historical data at two o'clock every morning, triggering a task every five minutes, or sending a monthly report on the first day of each month.

25.3.4. Maintenance management and service governance

Service authentication

HSF is designed to ensure the reliability and security of each distributed call. Strict authentication is implemented in every phase, from service registration and subscription to service calling.

Service throttling

EDAS can apply a number of throttling rules on each application to control service traffic and ensure service functionality. EDAS supports configuring throttling rules by QPS and thread to ensure maximum stability at traffic peaks.

Service degradation

Contrary to service throttling, service degradation pinpoints and blocks poor services that your application calls. This feature ensures the stable operation of your application and prevents the functionality of your application from being compromised by dependency on poor services. EDAS allows you to configure degradation rules by response time, which effectively blocks poor services at traffic peaks.

25.3.5. Three-dimensional monitoring

Distributed tracing

EDAS EagleEye analyzes every service call, sent message, and access to the database within the distributed system, so you can precisely identify system bottlenecks and risks.

Service call monitoring

EDAS can fully monitor the service calls made by your application in terms of the QPS, response time, and error rate of your services.

Infrastructure monitoring

EDAS can thoroughly monitor the running status of your application in terms of basic metrics such as CPU, memory, workload, network, and disk.

25.4. Performance metrics

Metric	Specifications
Access request processing capability	In simple calling scenarios, the size of a 1 KB service request message when the uncertain response time of a service provider is not considered, which can be linearly expanded.
Features	Enterprise Distributed Application Service (EDAS) provides Java containers that integrate multiple pieces of Internet middleware and is fully compatible with Apache Tomcat.
	EDAS implements application lifecycle management, including publishing, starting, stopping, and scaling out or in applications.
	EDAS monitors basic hardware metrics.
	EDAS monitors Java containers.
	EDAS provides a comprehensive service authentication mechanism.
	EDAS enables distributed RPCs and processes messages and transactions with multiple data sources.
	EDAS enables logging, inspection, monitoring, and tracing for service links and system metrics.
	EDAS pushes distributed system configurations.
Reliability	The data system uses multi-level cache and primary/secondary storage solutions.
Scalability	Service nodes can be continuously scaled out and in.
Proven technology	It is based on the highly available and high-performance distributed cluster technology products that have been used within Alibaba Group for a long period of time. In addition, the EDAS team members have rich experience in this field.

26.MaxCompute

26.1. What is MaxCompute?

26.1.1. Overview

MaxCompute is an offline data processing service developed by Alibaba Cloud based on the Apsara system. It is capable of processing large volumes of data. MaxCompute can process terabytes or petabytes of data in scenarios that do not have high real-time processing requirements. MaxCompute is used in fields such as log analysis, machine learning, data warehousing, data mining, and business intelligence.

MaxCompute provides an easy-to-use approach to analyze and process large amounts of data without deep knowledge of distributed computing. MaxCompute is widely implemented by Alibaba across its businesses for tasks such as data warehousing and BI analysis for large Internet enterprises, website log analysis, e-commerce transaction analysis, and exploration of user characteristics and interests.

MaxCompute provides the following features:

- Data channel
  - Tunnel: provides highly-concurrent offline upload and download services. The tunnel service enables you to upload or download large volumes of data to or from MaxCompute. You must use a Java API to access the tunnel service.
  - DataHub: provides real-time upload and download services. Unlike data uploaded through the tunnel service, data uploaded through DataHub is available immediately.
- Computing and analysis
  - SQL: MaxCompute stores data in tables, and provides SQL query capabilities to manipulate the data. MaxCompute can be used as database software capable of processing terabytes or petabytes of data. MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE. The SQL syntax used in MaxCompute is different from that in Oracle and MySQL. SQL statements from other database engines cannot be migrated seamlessly to MaxCompute. MaxCompute SQL responds to queries within a few minutes or seconds, instead of milliseconds. MaxCompute SQL is easy to learn. You can get started with MaxCompute SQL based on your prior experience of database operations, without having a deep understanding of distributed computing.
  - MapReduce: Initially proposed by Google, MapReduce is a distributed data processing model that has become popular and widely implemented for a variety of business scenarios. This topic briefly describes the MapReduce model. You must have a basic knowledge of distributed computing and relevant programming experience before using MapReduce. MapReduce provides a Java API.
  - Graph: a processing framework designed for iterative graph computing. Graph computing jobs use graphs to build models. A graph is a collection of vertices and edges that have values. MaxCompute Graph iteratively edits and evolves graphs to obtain analysis results.

  - Unstructured data access and processing (integrated computing scenarios): Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

    MaxCompute allows you to process the following data sources by creating external tables:

    - Internal data sources: OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.
    - External data sources: HDFS (Open Source), ApsaraDB for MongoDB, and Hbase.
  - Unstructured data access and processing in MaxCompute: By reading data from and writing data to volumes, MaxCompute can store unstructured data, which otherwise must be stored in an external storage system.

- Spark on MaxCompute: a big data analytics engine designed by Alibaba Cloud to provide big data processing capabilities for Alibaba, government agencies, and enterprises.
- Elasticsearch on MaxCompute: an enterprise-class system to retrieve information from large volumes of data and provide near-real-time search performance for government agencies and enterprises.

26.1.2. Features and benefits

Product features

MaxCompute is a distributed system designed for big data processing. As one of the core services in the Alibaba Cloud computing solution, MaxCompute is used to store and compute structured data. It is also a basic computing component of the Alibaba Cloud big data platform. MaxCompute is designed to support multiple tenants and provide features, such as data security and horizontal scaling. The service provides centralized programming interfaces for various data processing tasks of different users based on an abstract job processing framework. MaxCompute has the following features:

- Uses a distributed architecture that can be scaled as needed.
- Provides an automatic storage and fault tolerance mechanism to ensure high data reliability.
- Allows all computing tasks to run in sandboxes to ensure high data security.
- Uses RESTful APIs to provide services.
- Supports both uploads and downloads of high-concurrency, high-throughput data.
- Supports two service models: the offline computing model and the machine learning model.
- Supports data processing methods based on programming models such as SQL, MapReduce, Graph, and MPI.
- Supports multiple tenants, which allows multiple users to collaborate on data analytics.
- Provides user permission management based on access control lists (ACLs) and policies, which allows you to configure flexible data access control policies to prevent unauthorized access to data.
- Supports Elasticsearch on MaxCompute for enhanced applications.
- Supports Spark on MaxCompute for enhanced applications.
- Supports the access to and processing of unstructured data.
- Supports the deployment of multiple clusters in a single region.
- Supports multi-region deployment.
- Uses the column store method, and supports Key Management Service (KMS) to encrypt data files.
- Stores audit logs and dumps them to a specified server directory for long-term storage and

management.

Benefits

- **Excellent big data cloud service and real data sharing platform in China**: MaxCompute can be used for data warehousing, mining, analytics, and sharing. Alibaba Group implements this unified data processing platform in several of its own services, such as Aliloan, Data Cube, DMP (Alimama), and Yu'e Bao.
- **Support for large numbers of clusters, users, and concurrent jobs**: A single cluster can contain more than 10,000 servers and maintain 80% linear scalability. A single MaxCompute service supports more than 1 million servers in multiple clusters without limits. However, linear scalability is slightly affected. It also supports the local multi-data center mode. In addition, it supports more than 10,000 users, more than 1,000 projects, and more than 100 departments of multiple tenants. It can also support more than 1 million jobs (daily submitted jobs on average) and more than 20,000 concurrent jobs.
- **Big data computing at your fingertips**: You do not need to worry about the storage difficulties and prolonged computing time caused by the increase of the data volume. MaxCompute automatically expands the storage and computing capabilities of clusters based on the volume of data that needs to be processed. This allows you to focus on data analytics and mining to maximize your data value.
- **Out-of-the-box service**: You do not need to worry about cluster creation, configuration, and O&M. Only a few simple steps are required to upload data, analyze data, and obtain analysis results in MaxCompute.
- **Secure and reliable data storage**: The multi-level data storage and access control mechanisms are used to protect user data against loss, leak, and interception. These mechanisms include multi-replica technology, read/write request authentication, and application and system sandboxes.
- **Reliable management nodes**: Multi-node cluster architecture is used. The management nodes of each component feature high availability. Faults that occur on O&M management nodes do not affect normal business operations.
- **Powerful fault tolerance**: MaxCompute supports automatic fault tolerance for the failures of server hard disks in a cluster and supports hot swapping of hard disks. In the event of a hard disk failure, services can be restored within two minutes.
- **Comprehensive storage space management**: MaxCompute allows you to query information about both the storage capacity and usage of distributed file systems. It supports data lifecycle management. MaxCompute also allows you to store data in different locations based on the data value or tag. For example, you can write temporary files to SSDs to accelerate I/O operations. This facilitates efficient use of cluster data. MaxCompute also supports the self-optimizing Zstandard compression algorithm, which achieves the optimal compression ratio.
- **Comprehensive data backup**: MaxCompute allows you to perform full or incremental data backup and restore data from storage media. It also allows you to back up data for clusters in different data centers. This meets the requirements of mutual data backups among multiple data centers. You can use Apsara Bigdata Manager (ABM) to manage the backup process in a visualized manner.
- **Secure and reliable access control**: MaxCompute allows you to manage data access permissions, including logon permissions, table creation permissions, read/write permissions, and whitelist-related permissions. It also allows you to use the Apsara Stack Cloud Management (ASCM) console to manage administrative permissions, including administrator classification. You can use the ASCM console to manage user permissions in a centralized manner. You can manage the access control features of all components in the system. You can also block common users from querying access control details and simplify access control for administrators. This improves the usability and user

experience of access control.

- **Multi-tenancy for multi-user collaboration**: By configuring different data access policies, you can enable multiple data analysts in an organization to work together and make data accessible to users with permissions granted by the organization. This ensures data security and maximizes productivity.
  - **Isolation**: You can submit the tasks of multiple tenants (projects) to different queues for concurrent running. Resources are isolated among tenants.
  - **Permission**: You can manage different tenants in a centralized manner and perform dynamic configuration, management, isolation, and usage statistics of tenant resources. The management of multi-level tenants is supported.
  - **Scheduling**: MaxCompute supports multi-tenant scheduling for multiple clusters and multiple resource pools.
- **Multi-region deployment**: You can specify compute clusters to efficiently use computing resources. Data exchanges between clusters are completed within MaxCompute, and data replication and synchronization between clusters are managed based on configured policies. Therefore, cross-region data processing is no longer involved, which significantly reduces the waiting time for data processing.
- **Multi-device support**: You can use CPUs, hard disks, memory, and network interface controllers with different specifications in a single-component cluster without an effect on cluster running performance. This ensures maximum compatibility with existing devices.

26.1.3. Benefits

Compared with traditional databases, MaxCompute has the following benefits.

Comparison of benefits

Benefit	Traditional databases	MaxCompute
System scalability	Disks cannot be shared across more than 100 nodes. Table and database sharding causes application data collision, resulting in massive computing overhead. This significantly compromises application analysis capabilities.	MaxCompute supports more than 10,000 nodes that can store more than 1.5 EBs of data. For example, during Alibaba's Double 11 event, MaxCompute processed more than 300 PBs of data in six hours.
Data type support	Cannot process unstructured data.	Can process both structured and unstructured data.
High availability	Redundant storage solutions are not available. Traditional backup and recovery approaches are inapplicable to large volumes of data (measured in PBs), and a single point of failure can cause the entire database to become unavailable.	Provides the shared-nothing architecture and multi-replica data model. This eliminates single points of failure.

Benefit	Traditional databases	MaxCompute
Complex computing capability	Iterative computing and graph computing capabilities are not available. The disk sharing technology and complex computing operations result in massive data exchanges between nodes, imposing tremendous bandwidth pressure.	Provides distributed storage and multiple computing frameworks such as MR, SQL, iterative computing, MPI, and graph computing.
Concurrency	A single large-scale computing task (such as index computing) can consume all system resources, and incur network and disk (data dictionary) bottlenecks. This makes highly concurrent access impossible.	Provides comprehensive multi-tenant isolation and resource management tools, so that you can easily view cluster resources and manage the resources used by each service. It can support up to 10,000 concurrent access requests.
Performance support	The indexing mechanism makes it difficult to support analytical applications of real-time data. Large amounts of data collision cause analytical predictions to take more than 24 hours, resulting in a performance bottleneck.	Focuses on the concurrent computing of large amounts of data. It provides available real-time data, and multiple high-performance computing capabilities, such as high-performance large-scale offline computing, real-time multi-dimensional analysis of large amounts of data, and stream computing.

26.1.4. Scenarios

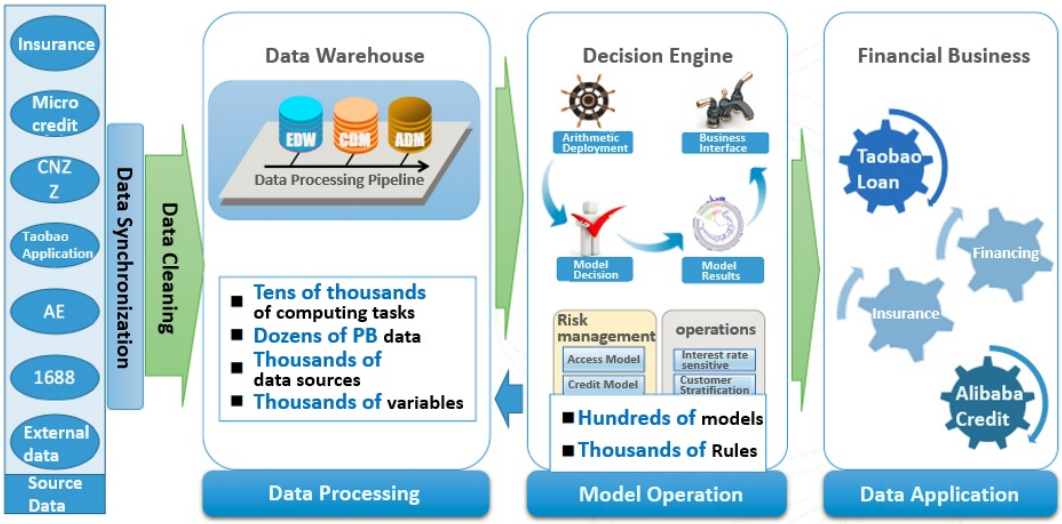
MaxCompute is designed for use in three big data processing scenarios:

- Establishment of SQL-based large data warehouses and BI systems
- Development of big data applications based on MapReduce and MPI distributed programming models
- Development of big data statistics models and data mining models based on statistics and machine learning algorithms

The following describe some real-world scenarios.

Data warehouse construction

Data warehouse construction



MaxCompute enables you to easily build a cloud-based data warehouse. With MaxCompute capabilities such as partitioning, data table statistics, and table life cycle management, you can easily enhance the storage of historical data warehouse information, divide hot and cold tables, and control data quality.

Alibaba's financial data warehousing team has built a sophisticated and powerful data warehousing system based on MaxCompute. This system provides six layers: the source data layer, ODS layer, enterprise data warehousing layer, common dimensional modeling layer, application marketplace layer, and presentation layer.

- The source data layer processes data from all sources, including Taobao, Alipay, B2B, and external data sources.
- ODS provides a temporary storage layer for data import.
- The enterprise data warehousing layer uses the 3NF modeling technique to divide data, including all historical data, by topic (such as item or shop).
- The common dimensional modeling layer uses the dimensional modeling approach to create modeling layers for general business applications. This layer shields the upper layers from changes in business requirements, and provides consistent and actionable data to the upper layers.
- The application marketplace layer is a demand-oriented layer that provides a data marketplace for specific applications.
- The presentation layer provides several data portals and services that can be accessed by applications.

This system architecture inevitably involves tasks such as metadata management.

The financial data warehouse is used to perform offline computing tasks based on MaxCompute SQL. It also uses a series of metric rules and algorithms to make decisions offline for online decision-making.

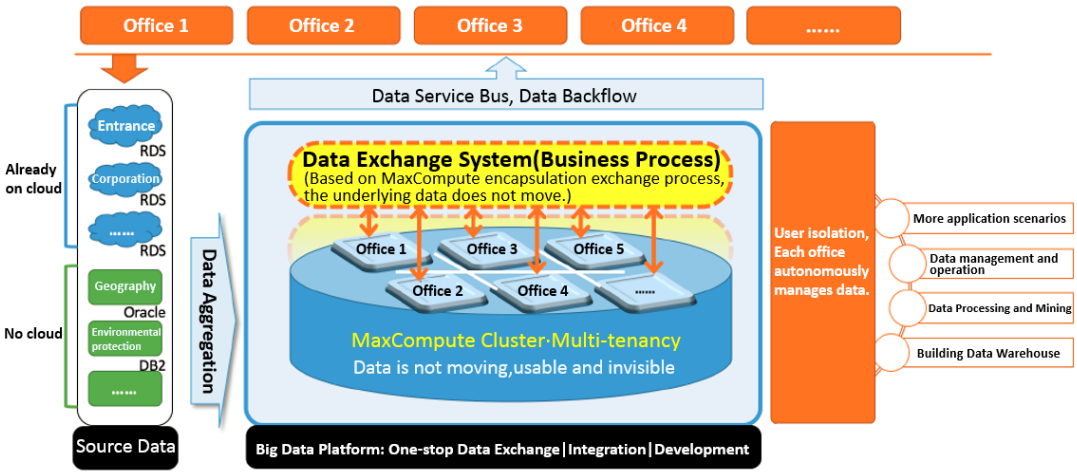
MaxCompute-based data warehouses differ from traditional databases in the following ways:

- **Historical data storage**: MaxCompute is able to store large amounts of data. You do not have to dump historical data to cheaper storage media as you would do in traditional databases.
- **Partitioning**: Traditional databases provide a wide range of partitioning methods such as range partitioning. MaxCompute provides fewer partitioning methods, but are sufficient for use in data warehousing scenarios. Whatever the method, you can build a data warehouse based on the same concept and principle as a table partition.
- **Wide tables**: MaxCompute stores data in fields, making it ideal for creating wide tables.

- **Data integration**: Traditional databases use stored procedures for data processing and integration. MaxCompute splits the logic of these operations into discrete SQL statements. Though the implementation is different, the algorithms are the same. In many years of experience, we found that splitting the operation logic into discrete SQL statements is clearer and more efficient, while stored procedures are more flexible and capable of processing complex logic.

Big data sharing and exchange

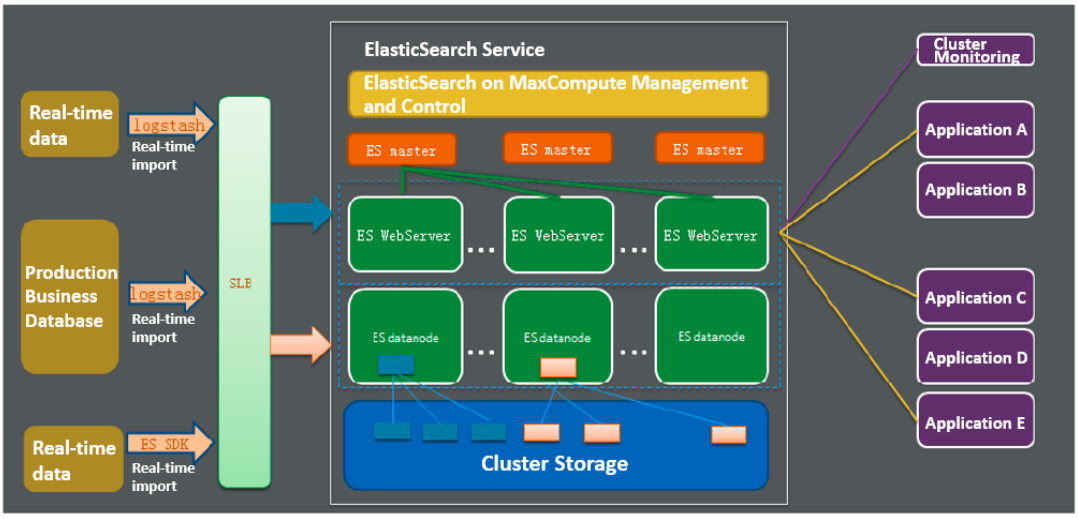
Big data sharing and exchange



MaxCompute provides a wide range of permission management methods and flexible data access control policies. MaxCompute provides a wide range of access control mechanisms, including the ACL authorization, role-based authorization, policy authorization, cross-project authorization, and label security mechanism. MaxCompute provides column-level security solutions. This can meet the security requirements within an organization or across multiple organizations. For projects that demand high security, MaxCompute provides the project protection mechanism to prevent data leakage, and provides logs of all user operations to facilitate retrospective audits.

Typical applications of Elasticsearch on MaxCompute

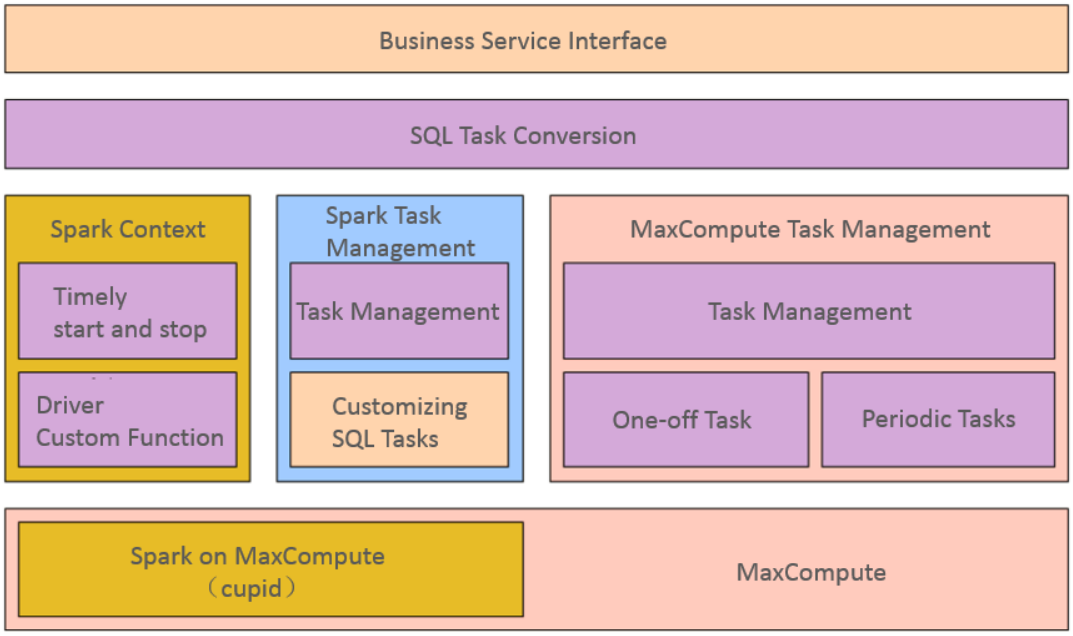
Typical applications



Elasticsearch on MaxCompute allows you to launch a set of Elasticsearch services by submitting jobs in a MaxCompute cluster. Native Elasticsearch code is not modified when applied in a project. **Elasticsearch on MaxCompute** runs in the same way as native Elasticsearch clusters.

Typical applications of Spark on MaxCompute

Typical applications



Spark on MaxCompute provides business computing platform and applications in Client mode. The preceding figure shows the application framework.

26.1.5. Service specifications

26.1.5.1. Software specifications

26.1.5.1.1. Overview

This section describes the software specifications of MaxCompute.

26.1.5.1.2. Control and service

Specifications

Item	Description
Number of control nodes	Greater than or equal to 3.
Number of MaxCompute front-end servers	Greater than or equal to 2. MaxCompute front-end servers can be deployed together with control nodes.
Number of tunnels	Greater than or equal to 2. Tunnels can be deployed together with compute nodes.

Item	Description
Number of DataHubs	Greater than or equal to 2. DataHubs can be deployed together with compute nodes.

26.1.5.1.3. Data storage

Specifications

Item	Description
Logical storage capacity per node	12 TB
Total storage capacity	The storage capacity can be scaled out by adding more nodes.

Note The size of logically stored data to a large extent determines the size of the cluster to be evaluated.

26.1.5.1.4. Size of a single cluster

Specifications

Item	Description
Offline computing cluster	An offline computing cluster can contain 3 to 10,000 machines.

26.1.5.1.5. Projects

Specifications

Item	Description
Creation of projects	Supported.
Acquisition of project metadata	Supported.
Deletion of projects	Supported.
Setting of the default lifecycle of tables	Supported.
Number of supported projects	Over 1,000

26.1.5.1.6. User management and security and access control

Specifications

Item	Description
Cross-project access	Supported. You can authorize cross-project access to organize tables and resources as packages and install them in other projects.
Service (odps_server and tunnel) authentication and access control	Supported. AccessKey ID and AccessKey Secret can be used to authenticate users and control their permissions.
Prevention of data outflow from a project	You can prevent data outflow and specify exceptions when necessary.
Label-based security	Label-based security (LabelSecurity) can be set to enable column-level access control. **Note** LabelSecurity is a mandatory access control policy that provides a wide range of security level settings.
Authorization to users	Supported.
Authorization to roles	Supported. You can customize roles and assign roles to users. Different roles are granted different permissions.
Project-specific authorization	The following permissions can be granted on a project: • View project information (excluding any project objects), such as the creation time. • Update project information (excluding any project objects), such as comments. • View the list of all object types in the project. • Create tables in the project. • Create instances in the project. • Create functions in the project. • Create resources in the project. • Create volumes in the project. • Grant all preceding permissions.

Item	Description
Table-specific authorization	The following permissions can be granted on a table: • Read table metadata. • Read table data. • Modify table metadata. • Overwrite or add table data. • Delete the table. • Grant all preceding permissions.
Function-specific authorization	The following permissions can be granted on a function: • Read. • Update. • Delete. • Grant all preceding permissions.
Authorization for resources, instances, jobs, and volumes	The following permissions can be granted on a resource, instance, job, or volume: • Read. • Update. • Delete. • Grant all preceding permissions.

Item	Description
Sandbox protection	The sandbox mechanism can restrict access to system resources in MapReduce and UDF programs. Specific restrictions are as follows: • Direct access to local files is not allowed. You can only read resource information and generate log information through System.out and System.err. **Note** You can view log information by running the Log command on the MaxCompute client. • Direct access to Apsara Distributed File System is not allowed. • JNI calls are not allowed. • Java threads cannot be created, and Linux commands cannot be executed by sub-threads. • Network access operations such as acquiring local IP addresses are not allowed. • Java reflection is not allowed. You cannot force access to protected or private members to be valid.
Control over the quotas of storage and computing resources	Supported. You can limit the number of files and used disk capacity in a project. You can also use quotas to limit the available CPU and memory capacity of the project.

26.1.5.1.7. Resource management and task scheduling

Specifications

Item	Description
File count quota and storage capacity quota	The quotas vary with projects.
Configuration of CPU quota for a resource group	You can configure the minimum or maximum number of virtual CPUs that can be used by a resource group.
Configuration of memory quota for a resource group	You can configure the minimum or maximum amount of virtual memory that can be used by a resource group.
Resource preemption	Preemption of resources within a quota group is supported.
Task scheduling methods	Fair scheduling and first-in-first-out (FIFO).

Item	Description
Configuration of task priorities	By default, task priorities are assigned in a project. You can configure the priorities as needed.
Restart of a failed task	Supported.
Speculative execution of a task	Supported.

26.1.5.1.8. Data tables

Specifications

Item	Description
Data storage methods	CFile data exclusive to MaxCompute is stored in columns in Apsara Distributed File System.
Data compression	Supported. The efficiency of compression is dependent on the data format. The compression ratio between the original and compressed data is 3:1. Infrequently accessed data can be archived in RAID to reduce the storage space it occupies by 50%.
Lifecycle	Supported.
Basic data types	BigInt, String, Boolean, Double, DateTime, and Decimal.
Partitions	Supported. Only String type partitions are supported.
Maximum number of columns	1,024
Maximum number of partitions	60,000
Partition levels	A table can contain up to five partition levels.
Views	Supported. A view can only contain one valid SELECT statement. Materialized views are not supported.
Statistics	Supported. You can define statistical metrics for data tables and view, analyze, and delete statistics.
Comments	Supported. You can make comments for both tables and columns. Comments can be up to 1024 characters in length.

26.1.5.1.9. SQL

26.1.5.1.9.1. DDL

Specifications

Item	Description
Creation of tables	Supported.
Deletion of tables	Supported.
Renaming of tables	Supported.
Creation of views	Supported.
Deletion of views	Supported.
Renaming of views	Supported.
Adding of partitions	Supported.
Deletion of partitions	Supported.
Adding of columns	Supported.
Modification of column names	Supported.
Modification of comments	Supported. You can modify comments for tables and columns.
Modification of the lifecycle of tables	Supported.
Disabling of the lifecycle for specific table partitions	Supported. The command syntax is as follows: `ALTER TABLE table_name [partition_spec] ENABLE|DISABLE LIFECYCLE`
Emptying of data from non-partitioned tables	Supported. The command syntax is as follows: `TRUNCATE TABLE table_name`
Modification of table owners	Supported.
Modification of the time when a table or partition was last modified	Supported.

26.1.5.1.9.2. DML

Specifications

Item	Description

Item	Description

Dynamic partition filtering	Supported. This technique can reduce the amount of data to be read. The command syntax is as follows:
	`select_statment`
	`FROM from_statement`
	`WHERE PT1 IN (SUBQUERY) AND PT2 IN (SUBQUERY)... ;`

Multiple outputs	Supported. A single SQL statement can contain up to 128 outputs. **Note** In each output, you can only specify once whether to target a partition in a partitioned table or target a non-partitioned table.
Data update and overwriting	Supported. Batch update is supported.
Aggregation	Supported.
Sorting	Supported. Sorting must be performed with the limit syntax.
Nested subqueries	Supported.
Joins	Supported. SQL joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
UNION ALL	Supported.
CASE WHEN	Supported.
Relational operations	Supported.
Mathematical operations	Supported.
Logical operations	Supported.
Implicit conversions	Supported.

Item	Description
MAPJOIN	Supported. To speed the JOIN operation when volume of data is small, SQL loads all specified small tables into the memory of a program executing the JOIN operation. The default maximum data size is 512 MB. The maximum size cannot exceed 2 GB. Up to six small tables can be specified. **Note** Take note of the following limits: • The left table of a LEFT OUTER JOIN clause must be a large table. • The right table of a RIGHT OUTER JOIN clause must be a large table. • Both the left and right tables of an INNER JOIN clause can be large tables. • MAPJOIN cannot be used in a FULL OUTER JOIN clause. • MAPJOIN supports small tables as subqueries. • When MAPJOIN is used and a small table or subquery is referenced, you must reference the alias of the small table or subquery. • MAPJOIN supports both non-equivalent JOIN conditions and multiple conditions connected by using OR statements.
Query of the execution plans of DML statements	Supported. The description of the final execution plan corresponding to a DML statement can be displayed. The command syntax is as follows: `EXPLAIN <DML query>;`

26.1.5.1.9.3. Built-in functions

Specifications

Item	Description
Built-in functions	Supported. Built-in functions include string functions, date functions, mathematical functions, regular functions, and window functions.

26.1.5.1.9.4. User-defined functions

Specifications

Item	Description
Scalar functions	Supported. You can use the Java SDK and Python SDK to write scalar functions.

Item	Description
Aggregate functions	Supported. You can use the Java SDK and Python SDK to write aggregate functions.
Table functions	Supported. You can use the Java SDK and Python SDK to write table functions.
Implicit conversions	Supported.

26.1.5.1.10. MapReduce

26.1.5.1.10.1. Programming support

Specifications

Item	Description
Java language	Supported.
Standalone debugging mode	Supported.
Extended MapReduce model	Supported. A Map operation can be followed by any number of Reduce operations. Example: Map-Reduce-Reduce.

26.1.5.1.10.2. Job size

Specifications

Item	Description
Maximum number of mappers	100,000
Maximum number of reducers	2,000
Setting of the number of mappers and reducers	Supported. You can change the number of mappers by changing the input volume of each Map worker. By default, the number of reducers is set at 25% of the number of mappers. You can change this proportion to suit your business needs.
Setting of the memory of mappers and reducers	Supported. The default memory of a mapper or reducer is 2 GB.

Note The maximum numbers of mappers and reducers are related to the cluster size.

26.1.5.1.10.3. Input and output

Specifications

Item	Description
Input and output of a table	Supported.
Processing of unstructured data	Supported. Volumes are suited to store unstructured data. MaxCompute MapReduce can be used to process unstructured data.
Input and output of multiple tables	Supported. The numbers of inputs and outputs cannot exceed 128.
Reading of resources	Supported. A single task can reference up to 256 resources. The total size of all referenced resources cannot exceed 2 GB. Note The maximum number of read attempts for a resource is 64.

26.1.5.1.10.4. MapReduce computing

Specifications

Item	Description
Custom setup, map, and cleanup methods of mappers	Supported.
Custom setup, reduce, and cleanup methods of reducers	Supported. Transmitted messages are processed in the next iteration.
Custom partition columns or partitioners	Supported.
Configuration of mapper output columns to be sorted and grouped by keys	Supported. Note that custom key comparators are not supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Map-only jobs	Supported. To implement Map-only jobs, set the number of Reduce jobs to 0.
Configuration of job priorities	Supported.

26.1.5.1.11. Graph

26.1.5.1.11.1. Programming support

Specifications

Item	Description
Java language	Supported.
Standalone debugging mode	Supported.

26.1.5.1.11.2. Job size

Specifications

Item	Description
Maximum number of concurrent workers	1,000
Custom worker CPU and memory	Supported. By default, a worker has two CPU cores and 4 GB of memory. A worker can have up to eight CPU cores and 12 GB of memory.

26.1.5.1.11.3. Graph loading

Specifications

Item	Description
Loading of graph data from MaxCompute tables	Supported.
Division of graphs by vertex	Supported.
Custom partitioners	Supported.
Custom split size	Supported. The default split size is 64 MB.
Custom conflict logic upon data loading	Supported. For example, creating duplicate vertices and edges is considered a conflict logic.

26.1.5.1.11.4. Iterative computing

Specifications

Item	Description
Bulk Synchronous Parallel (BSP) computing model	Supported.
Transmission of messages between vertices	Supported. Transmitted messages are processed in the next iteration.

Item	Description
Multiple iteration termination conditions	1. The maximum number of iterations is reached. 2. All vertices enter the halted state. 3. An aggregator determines to terminate the iteration.
Automatic checkpoint mechanism	Supported.
Custom aggregators	Supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Custom conflict logic	Supported. For example, sending messages to a non-existent vertex is considered a conflict logic.
Writing of computing results to MaxCompute tables	Supported.
Configuration of job priorities	Supported.

26.1.5.1.12. Processing of unstructured data

26.1.5.1.12.1. Processing of Table Store data

Specifications

Item	Description
Table Store data types	A variety of data types are supported.

26.1.5.1.12.2. Processing of OSS data

Specifications

Item	Description
User-defined split and range functions	Supported.
User-defined maximum number of concurrent mappers	Supported.
User-defined file list	Supported.

26.1.5.1.12.3. Multiple data sources

Specifications

Item	Description
Support for various open-source data formats through the STORED AS syntax	Supported data formats include PARQUET, ORC, SEQUENCEFILE, TEXTFILE, and AVRO.

26.1.5.1.13. Spark on MaxCompute

26.1.5.1.13.1. Programming support

Specifications

Item	Description
Native Apache Spark APIs	Supported. You can use native Spark APIs to write code and process data stored in MaxCompute.
Native methods to submit Spark jobs	Supported.
Multiple native Spark components	Spark SQL, Spark MLlib, GraphX, and Spark Streaming are currently supported.
Multiple programming languages	MaxCompute data can be processed using Scala, Python, Java, and R languages.

26.1.5.1.13.2. Data sources

Specifications

Item	Description
Processing of unstructured data	Supported. You can use Spark APIs to write code and process data stored in OSS and Table Store.
Processing of data from MaxCompute tables and resources	Supported.

26.1.5.1.13.3. Scalability

Specifications

Item	Description
Deep integration of Spark and MaxCompute	Supported. Spark and MaxCompute share cluster resources. Spark resources can be scaled from large-scale MaxCompute clusters.

26.1.5.1.14. Elasticsearch on MaxCompute

26.1.5.1.14.1. Programming support

Specifications

Item	Description
Native Elasticsearch APIs	Supported.

26.1.5.1.14.2. System capabilities

Specifications

Item	Description
Real-time analysis and retrieval of data at the petabyte level	Supported.
Web-based display for basic server metrics	Supported. A user-friendly O&M platform for index databases and full-text retrieval clusters can be used to monitor the status of index databases and machines in real time.
Data snapshot technology based on Apsara Distributed File System	Supported. Rapid data backup and recovery can be performed to ensure data reliability.
Millisecond-level response to keyword-based and comprehensive searches and second-level response to fuzzy searches	Supported.
Real-time analysis and retrieval of imported data and query response times within 500 milliseconds	Supported. The storage architecture is powered by the distributed cache-accelerated block device technology.
In-memory off-heap storage and processing of index data and fine-grained memory management	Supported.

26.1.5.1.15. Other extensions

The following extended plug-ins and tools are both client-specific and open-source. You can download the plug-ins and tools at https://github.com/aliyun/.

Specifications

Item	Description
R language support	RODPS is a plug-in for the MaxCompute client to support the R language.
Plug-ins and tools	Eclipse plug-ins and command line tools are available.
OGG	OGG plug-ins synchronize data from OGG to DataHub.

Item	Description
Flume	Flume plug-ins synchronize data from Flume to DataHub.
FluentD	FluentD plug-ins synchronize data from FluentD to DataHub.
JDBC	JDBC interfaces are partially supported.
Sqoop	Sqoop can be used to exchange data with MaxCompute.

26.1.5.2. Hardware specifications

The following table lists the hardware specifications of MaxCompute.

Hardware specifications

Node type	Server configuration	Number of nodes	Description
Management node	• CPU: dual-socket 8-core or higher • Memory: 256 GB or higher • Disk: two 4 TB NVMe U.2 SSDs • NIC: two 10 GE NICs for network bonding	N/A	We recommend that you use Intel Platinum 81xx series processors or higher configurations.

Node type	Server configuration	Number of nodes	Description
Control node	• CPU: dual-socket 8-core or higher • Memory: 128 GB or higher • Disk: one 4 TB SATA HDD with 7200 RPM performance • NIC: two 10 GE NICs for network bonding	8/13	• We recommend that you use Intel Platinum 81xx series processors or higher configurations. • When the number of data nodes is less than 500, the number of control nodes is 8. When the number of data nodes is more than 500, the number of control nodes is 13. • We recommend that you deploy data nodes in containers when the number of data nodes is less than 500. • When all control nodes are physical servers and the number of data nodes is less than 1,000, you can implement a hybrid deployment of control nodes and data nodes based on your actual needs. • The system disk capacity is greater than or equal to 240 GB.
Hybrid deployment of management nodes and control nodes	• CPU: dual-socket 8-core or higher • Memory: 256 GB or higher • Disk: one 4 TB NVMe U.2 SSD • NIC: two 10 GE NICs for network bonding	N/A	• Hybrid deployment is recommended when the number of data nodes is less than 500 and is not expected to be increased. • Assume that the number of data nodes is approximately 500 and is expected to increase to more than 500. When you deploy the nodes for the first time, we recommend that you deploy them separately on physical servers.

Node type	Server configuration	Number of nodes	Description
Data node	• CPU: dual-socket 8-core or higher • Memory: 128 GB or higher • Disk: twelve 2 TB, 4 TB, 6 TB, or 8 TB SATA HDDs with 7200 RPM performance • NIC: two 10 GE NICs for network bonding	Depends on the amount of data.	• We recommend that you use Intel Golden 61xx series processors or higher configurations. • The recommended ratio of core quantity to memory capacity is 1:4. • We recommend that you add a 4 TB NVMe U.2 SSD when the number of cores is greater than or equal to 48. • Number of nodes = [(Total planned data volume × Data expanding rate (1.3) × Data compression rate (1) × Number of replicas (3))/Disk utilization rate (0.85)/Disk formatting loss (0.9)/((Number of disks (12) - Number of system reserved blocks (1)) × Disk capacity (8 TB))] rounded up.

Note

- We recommend that you use the preceding configurations in offline scenarios as needed.
- We recommend that you do not use two or more machine types for compute nodes of MaxCompute.
- We recommend that you do not use both 1 GE and 10 GE NICs for MaxCompute.
- The configuration of machines to be added cannot be lower than that of the existing machines.
- The reuse of compute nodes needs to be evaluated together with the business side.

26.1.5.3. Specifications of DNS resources

Specifications

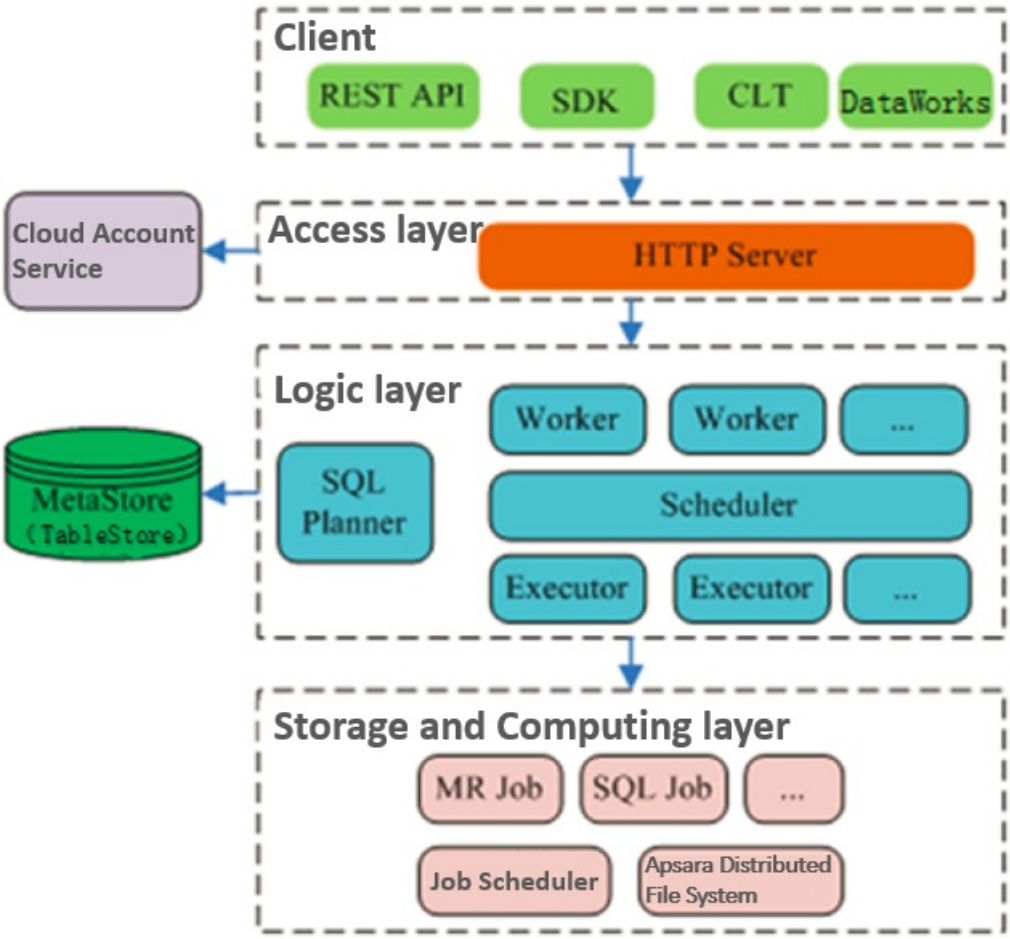
Resource name	Domain name	Description
	odps_frontend_server_inner_dns	The internal domain name of the MaxCompute front-end server. This domain name is not subject to VPC.

Resource name odps_frontend	Domain name	Description
	odps_frontend_server_public_dns	The private domain name of the MaxCompute front-end server.
	odps_frontend_server_internet_dns	The public domain name of the MaxCompute front-end server.
tunnel_frontend	odps_tunnel_frontend_server_inner_vip	The internal domain name of the front-end server for MaxCompute Tunnel. This domain name is not subject to VPC.
	odps_tunnel_frontend_server_public_vip	The private domain name of the front-end server for MaxCompute Tunnel.
	odps_tunnel_frontend_server_internet_vip	The public domain name of the front-end server for MaxCompute Tunnel.
cupid_web_proxy	odps_jobview_dns	The internal domain name of the MaxCompute Jobview. This domain name is not subject to VPC.
logview	odps_logview_inner_dns	The internal domain name of the MaxCompute Logview. This domain name is not subject to VPC.
	odps_logview_public_dns	The private domain name of the MaxCompute Logview.
web_console	odps_webconsole_inner_dns	The internal domain name of the MaxCompute Web console. This domain name is not subject to VPC.
	odps_webconsole_public_dns	The private domain name of the MaxCompute Web console.

26.2. Architecture

MaxCompute architecture shows the MaxCompute architecture.

MaxCompute architecture



The MaxCompute service is divided into four parts: **client**, **access layer**, **logic layer**, and **storage and computing layer**. Each layer can be scaled out.

The following methods can be used to implement the functions of a MaxCompute client:

- **API**: RESTful APIs are used to provide offline data processing services.
- **SDK**: RESTful APIs are encapsulated in SDKs. SDKs are currently available in programming languages such as Java.
- **Command line tool (CLT)**: This client-side tool runs on Windows and Linux. CLT allows you to submit commands to manage projects and use DDL and DML.
- **DataWorks**: DataWorks provides upper-layer visual ETL and BI tools that allow you to synchronize data, schedule tasks, and create reports.

The access layer of MaxCompute supports HTTP, HTTPS, load balancing, user authentication, and service-level access control.

The logic layer is at the core of MaxCompute. It supports project and object management, command parsing and execution logic, and data object access control and authorization. The logic layer is divided into control and compute clusters. The control cluster manages projects and objects, parses queries and commands, and authorizes access to data objects. The compute cluster executes tasks. Both control and compute clusters can be scaled out as required. The control cluster is comprised of three different roles: Worker, Scheduler, and Executor. These roles are described as follows:

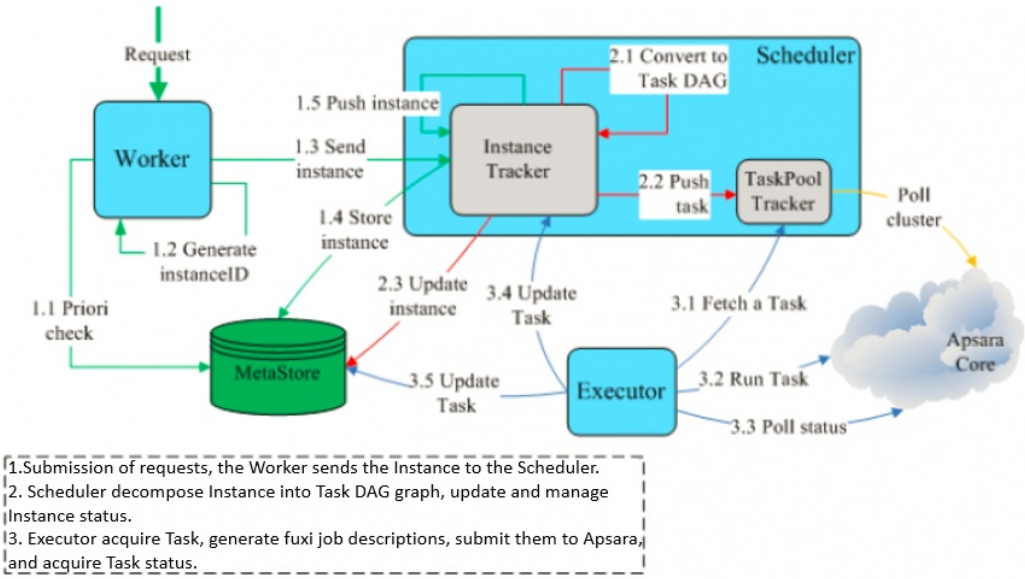
- **The Worker role processes all RESTful requests** and manages projects, resources, and jobs. Workers forward jobs that need to launch Fuxi tasks (such as SQL, MapReduce, and Graph jobs) to the

Scheduler for further processing.

- **The Scheduler role schedules instances**, splits instances into multiple tasks, sorts tasks that are pending for submission, and queries resource usage from FuxiMaster in the compute cluster for throttling. If there are no idle slots in Job Scheduler, the Scheduler stops processing task requests from Executors.
- **The Executor role is responsible for launching SQL and MapReduce tasks**. Executors submit Fuxi tasks to FuxiMaster in the compute cluster and monitor the operating status of these tasks.

In summary, when you submit a job request, the Web server at the access layer queries the IP addresses of registered Workers and sends API requests to randomly selected Workers. The Workers then send these requests to the Scheduler for scheduling and throttling. Executors actively poll the Scheduler queue. If the necessary resources are available, the Executors start executing tasks and return the task execution status to the Scheduler. The following figure shows the corresponding business execution logic.

Business execution logic



The storage and computing layer of MaxCompute is a core component of the proprietary cloud computing platform developed by Alibaba Cloud. As the kernel of the Apsara system, this layer runs in the compute cluster independent of the control cluster. The preceding MaxCompute architecture diagram illustrates only major modules of the Apsara kernel, such as Apsara Distributed File System and Job Scheduler.

Among the modules, **Apsara Distributed File System** is designed to aggregate the storage resources of a large number of machines and provide users with reliable large-scale distributed storage services. Apsara Distributed File System is an important part of the Apsara kernel.

Apsara Distributed File System includes three masters and multiple chunkservers. A master is responsible for the storage and management of file metadata, while a chunkserver is responsible for data storage. Identical blocks of data are stored on multiple chunkservers to ensure its reliability. In normal cases, data is stored in Apsara Distributed File System in three copies. All MaxCompute data files are stored in Apsara Distributed File System. The data files can be found in the /product/aliyun/odps directory. It is important to note that masters operate in hot standby mode. Only one master operates at a time.

- **master**:
  i. PanguMaster maintains metadata for the entire file system, including namespaces, file-to-block

mappings, and data block storage addresses.

ii. PanguMaster is the heart of the distributed file storage system and controls system-level activities such as garbage collection for isolated data blocks, data merging among chunkservers, chunkserver health check, and recovery of data blocks lost due to a down chunkserver.

iii. PanguMaster also manages data access requests originating from multiple clients at the same time to ensure the integrity of data in the cluster.

iv. PanguMaster only allows the client to perform operations on metadata. Data transmissions are conducted directly among chunkservers.

- **chunkserver**: The files in Apsara Distributed File System are divided into fixed-size storage units called chunks. A machine that stores chunks is called a chunkserver. PanguMaster assigns a 128-bit ID to a chunk when the chunk is created. The Apsara Distributed File System client reads the chunks stored in disks based on chunk IDs.

To provide better support for the processing of structured data in MaxCompute tables, the MaxCompute team has implemented a special Apsara Distributed File System file format called CFile.

- CFile is a file format based on column storage. It is designed to reduce invalid disk read operations during offline data processing. Data in the file is clustered by column into blocks and compressed to reduce storage space. This means that in offline data processing scenarios, you only need to read the required data. This avoids unnecessary disk operations, improves disk read efficiency, and reduces network bandwidth consumption.
- The CFile storage structure can be logically divided into three areas: data area, index area, and header area. The data area stores the user data that is clustered by column and uses blocks as organizational units. The index area stores the indexes corresponding to the data blocks of each column, which includes the starting position of each block in the file, the length of a compressed block, and the data amount in a block (for variable-length data types such as string). The header area stores the metadata of each column in the file, such as the starting position of the column index, index length, column type information, compression method, and row count and version of user data.
- MaxCompute supports the following data types:
  - Bigint: represents an 8-byte signed integer.
  - Boolean: represents a logical true or false value.
  - Double: represents an 8-byte double-precision floating-point number.
  - String: represents a string in UTF-8 format. MaxCompute functions automatically assume that string objects contain UTF-8 encoded strings. If the string is encoded in other formats, an error occurs.
  - Datetime: represents a date and time in the YYYY-MM-DD HH:mm:SS format. Example: 2012-01-02 10:09:25

Job Scheduler is a module for resource management and task scheduling in the Apsara kernel. It also provides a basic programming framework for application development. It is designed to make full use of the hardware resources of the entire cluster to meet the computing requirements of users and systems. Job Scheduler supports the processing of two application types: a low-latency online service called FuxiService and a high-throughput offline processing application called FuxiJob. Job Scheduler is similar to YARN in Hadoop.

- **FuxiService**: a resident process in Job Scheduler. You can send requests to create and destroy a service. Job Scheduler does not proactively destroy a service process.
- **FuxiJob**: a temporary task in Job Scheduler. When a task ends, the resources are released and reclaimed by Job Scheduler.

Job Scheduler schedules and allocates cluster storage and computing resources to upper-layer applications. Job Scheduler is able to manage computing resource quotas, access control policies, and job priorities to ensure that resources are shared effectively. Job Scheduler provides a data-driven, multi-level parallel computing framework, which is similar to the MapReduce programming model. The framework is ideal for complex applications such as large-scale data processing and large-scale computing.

Job Scheduler has two masters and multiple Tubos. Masters operate in cold standby mode. Only one master operates at a time. A Tubo process is started on each compute node to manage available resources of each machine, such as the CPU, memory, hard disk, and network, and record the resources used on each machine. The Tubo process on each machine reports the resource usage to FuxiMaster, which centrally manages and schedules the resources.

26.3. Features

26.3.1. Tunnel

26.3.1.1. Overview

Data upload and download tools provided by MaxCompute are compiled based on the Tunnel SDK. This topic describes the major APIs of the Tunnel SDK.

The usage of the SDK varies according to the version. For specific information, see SDK Java Doc.

Major APIs

API	Description
TableTunnel	An entry class of the MaxCompute Tunnel service.
TableTunnel.UploadSession	A session that uploads data to a MaxCompute table.
TableTunnel.DownloadSession	A session that downloads data from a MaxCompute table.
InstanceTunnel	An entry class of the MaxCompute Tunnel service.
InstanceTunnel.DownloadSession	A session that downloads data from a MaxCompute instance. This session applies only to SQL instances that start with the SELECT keyword and are used to query data.

Note The tunnel endpoint supports automatic routing based on the MaxCompute endpoint settings.

26.3.1.2. TableTunnel

This topic describes the TableTunnel API.

Definition

Definition:

```
public class TableTunnel {
public DownloadSession createDownloadSession(String projectName, String tableName);
public DownloadSession createDownloadSession(String projectName, String tableName, PartitionSpec par
titionSpe c);
public UploadSession createUploadSession(String projectName, String tableName);
public UploadSession createUploadSession(String projectName, String tableName, PartitionSpec partitionS
pec);
public DownloadSession getDownloadSession(String projectName, String tableName, PartitionSpec partiti
onSpec, String id);
public DownloadSession getDownloadSession(String projectName, String tableName, String id);
public UploadSession getUploadSession(String projectName, String tableName, PartitionSpec partitionSpe
c, String id);
public UploadSession getUploadSession(String projectName, String tableName, String id); public void setEn
dpoint(String endpoint);
}
```

Description:

- **Lifecycle**: the duration from the creation of the TableTunnel instance to the end of the program.
- TableTunnel provides a method to create UploadSession and DownloadSession objects. TableTunnel.UploadSession is used to upload data, and TableTunnel.DownloadSession is used to download data.
- A session refers to the process of uploading or downloading a table or partition. A session consists of one or more HTTP requests to Tunnel RESTful APIs.
- Upload sessions of TableTunnel use the INSERT INTO semantics. Multiple upload sessions of the same table or partition does not affect each other, and the data uploaded in each session is stored in an independent directory.
- In an upload session, each RecordWriter is matched with an HTTP request and is identified by a unique block ID. The block ID is the name of the file corresponding to the RecordWriter.
- If you use the same block ID to enable a RecordWriter multiple times in the same session, the data uploaded by the RecordWriter that calls the close() function last will overwrite all previous data. This feature can be used to retransmit data of a block when data upload fails.

API implementation process

1. The RecordWriter.write() function uploads your data as files to a temporary directory.
2. The RecordWriter.close() function moves the files from the temporary directory to the Data directory.
3. The session.commit() function moves each file in the Data directory to the directory where the corresponding table is located and updates the table metadata. This way, data moved into a table by the current task will be visible to the other MaxCompute tasks such as SQL and MapReduce.

API limits

- The value of a block ID must be greater than or equal to 0 and less than 20000. The size of data to be uploaded in a block cannot exceed 100 GB.
- A session is uniquely identified by its session ID. The lifecycle of a session is 24 hours. If your session times out due to the transfer of large volumes of data, you must transfer your data in multiple sessions.

- The lifecycle of an HTTP request corresponding to a RecordWriter is 120 seconds. If no data flows over an HTTP connection within 120 seconds, the server closes the connection.

  > **Note** HTTP has an 8 KB buffer. When you call the RecordWriter.write() function, your data may be saved to the buffer and no inbound traffic flows over the corresponding HTTP connection. In this case, you can call the TunnelRecordWriter.flush() function to forcibly flush data from the buffer.

- When you use a RecordWriter to write logs to MaxCompute, the RecordWriter may time out due to unexpected traffic fluctuations. Therefore, we recommend that you:
  - Do not use a RecordWriter for each data record. Otherwise, a large number of small files are generated, because each RecordWriter corresponds to a file. This affects the performance of MaxCompute.
- Do not use a RecordWriter to write data until the size of cached code reaches 64 MB.
- The lifecycle of a RecordReader is 300 seconds.

26.3.1.3. InstanceTunnel

This topic describes the InstanceTunnel API.

Definition:

```
public class InstanceTunnel{
 public DownloadSession createDownloadSession(String projectName, String instanceID);
 public DownloadSession createDownloadSession(String projectName, String instanceID, boolean limitEnab
led);
 public DownloadSession getDownloadSession(String projectName, String id);
 }
```

Parameter description:

- **projectName**: the name of a project.
- **instanceID**: the ID of an instance.

Limits: Although InstanceTunnel provides an easy way to obtain instance execution results, it is subject to the following permission limits to ensure data security:

- If the number of records does not exceed 10,000, all users who have the read permission on the specified instance can use InstanceTunnel to download the data. This is also applicable to the scenario of calling a Restful API to query data.
- If the number of records exceeds 10,000, only users who have the permission to read all the source tables from which the specified instance queries data can use InstanceTunnel to download the data.

26.3.1.4. UploadSession

This topic describes the UploadSession interface.

Definition

```
public class UploadSession {
UploadSession(Configuration conf, String projectName, String tableName, String partitionSpec) throws Tu
nnelException;
UploadSession(Configuration conf, String projectName, String tableName, String partitionSpec, String uplo
adId) throws TunnelException;
public void commit(Long[] blocks); public Long[] getBlockList();
public String getId();
public TableSchema getSchema();
public UploadSession.Status getStatus(); public Record newRecord();
public RecordWriter openRecordWriter(long blockId);
public RecordWriter openRecordWriter(long blockId, boolean compress);
}
```

The following section describes the UploadSession interface.

Description

Item	Description
Lifecycle	The lifecycle of an upload instance starts when the instance is created and ends when data is uploaded.
Create a data upload instance	Create a data upload instance by calling a constructor method or by using TableTunnel. • The synchronous request mode is used. • The server creates a session for the data upload instance and generates a unique upload ID. You can run the getId command to obtain the upload ID.
Upload data	• The asynchronous request mode is used. • Call the openRecordWriter method to generate a RecordWriter. The blockId parameter identifies the data to upload and the position of the data in the table. The value of the blockId parameter ranges from 0 to 20000. If the upload fails, you can upload the data again based on the block ID.
View the status of an upload session	• The synchronous request mode is used. • Call the getStatus method to obtain the status of an upload session. • Call the getBlockList method to obtain the block IDs of successful upload sessions. You must check the block IDs of all upload sessions to identify failed upload sessions. Then, upload the data of failed sessions again.
Complete a data upload	• The synchronous request mode is used. • Call the commit(Long[] blocks) method. The blocks parameter indicates the list of the block IDs of successful upload sessions. The server verifies the list. • The verification enhances data accuracy. If the provided list of block IDs is different from the list on the server, an error is reported.

Item	Description
Status	• UNKNOWN: the initial state of a session. • NORMAL: An upload session is created. • CLOSING: The server sets the upload session to the CLOSING state before it calls the COMPLETE method to complete the data upload. • CLOSED: The data upload is complete. The data is moved to the directory of the result table. • EXPIRED: The upload session times out. • CRITICAL: An error occurs.

Notice

- Each block ID in an upload session must be unique. If you use a block ID to open a RecordWriter, write data, and then call the CLOSE and COMMIT methods, you cannot use this block ID to open another RecordWriter.
- The maximum size of a block is 100 GB. Make sure that the volume of data written to each block is greater than 64 MB. Otherwise, computing performance is severely reduced.
- The lifecycle of a session is 24 hours.
- Before you call the openRecordWriter method to write data, we recommend that you prepare data. A network action is triggered every time an openRecordWriter writes 8 KB of data. If no network actions are triggered within 120 consecutive seconds, the server closes the connection and the openRecordWriter becomes unavailable. If this happens, you must open a new openRecordWriter.
- The overwrite mode is added in the COMMIT method. You can use the overwrite mode to submit data. If you use the overwrite mode, the data in this commit overwrites the existing data of a table or partition.

  Notice If multiple sessions are concurrently executed and the overwrite mode is used to submit data, undefined behavior is generated. This may cause data inaccuracy. If you use the overwrite mode to submit data, you must control concurrent commits.

26.3.1.5. DownloadSession

This topic describes the DownloadSession class.

API definition:

```
public class DownloadSession {
DownloadSession(Configuration conf, String projectName, String tableName, String partitionSpec) throws
TunnelException
DownloadSession(Configuration conf, String projectName, String tableName, String partitionSpec, String d
ownloadId) throws TunnelException
public String getId()
public long getRecordCount() public TableSchema getSchema()
public DownloadSession.Status getStatus()
public RecordReader openRecordReader(long start, long count)
public RecordReader openRecordReader(long start, long count, boolean compress)
}
```

DownloadSession API description.

DownloadSession API

Parameter	Description
Lifecycle	From the creation of the Download instance to the end of the download process.
Purpose	Creates a Download instance by calling a constructor method or using TableTunnel. • Request mode: Synchronous. • The server creates a session for this Download and generates a unique download ID to mark the Download. The console can get data with a get ID. The operation has a high overhead. The server creates indexes for the data files. If many data files exist, the operation takes a long time. Then the server returns the total number of records, and starts concurrent downloads according to the number of records.
Download data	• Request mode: Asynchronous. • Call openRecordReader to generate a RecordReader instance. The Start parameter marks the start position of record for this download. The value of Start is equivalent to or greater than 0. The Count parameter marks the number of records for this download. The value of Count is greater than 0.
View the download process	• Request mode: Synchronous. • Call getStatus to get the download status.
Status	• UNKNOWN: the initial value that is set when the server creates a session. • NORMAL: The download object is successfully created. • CLOSED: The download session is completed. • EXPIRED: The download session times out.

26.3.1.6. TunnelBufferedWriter

This topic describes the TunnelBufferedWriter interface.

The upload process is complex due to limits on block management and connection timeout on the server. The Tunnel SDK provides an enhanced RecordWriter, TunnelBufferWriter, to simplify the upload process.

The TunnelBufferedWriter interface is defined as follows:

```
public class TunnelBufferedWriter implements RecordWriter {
  public TunnelBufferedWriter(TableTunnel.UploadSession session, CompressOption option) throws IOExc
eption;
  public long getTotalBytes();
  public void setBufferSize(long bufferSize);
  public void setRetryStrategy(RetryStrategy strategy);
  public void write(Record r) throws IOException;
  public void close() throws IOException;
}
```

A TunnelBufferedWriter object is described as follows:

- **Lifecycle**: the duration from the time RecordWriter is created to the time the data upload ends.
- **TunnelBufferedWriter instance**: You can call the openBufferedWriter interface of UploadSession to create a TunnelBufferedWriter instance
- **Data upload**: When you call the Write interface, data is first written to the local cache. After the cache is full, the data is submitted to the server in batches to avoid connection timeout. In addition, if the upload fails, the system automatically retries the upload operation.
- **End upload**: Call the Close interface and then call the Commit interface of UploadSession to end the upload process.
- **Buffer control**: You can use the setBufferSize interface to modify the memory occupied by the buffer (in bytes), preferably 64 MB or more to prevent the server from generating too many small files, which may affect performance. The valid range is 1 MB to 1000 MB. The default value is 64 MB, which is recommended in most cases.
- **Retry policy settings**: You have three retry avoidance policies to choose from: EXPONENTIAL_BACKOFF, LINEAR_BACKOFF, and CONSTANT_BACKOFF. For example, the following code segment sets the Write retry count to 6. To avoid unnecessary retries, each retry is performed only after exponentially ascending intervals of 4s, 8s, 16s, 32s, 64s, and 128s by default.

  ```
  RetryStrategy retry
   = new RetryStrategy(6, 4, RetryStrategy.BackoffStrategy.EXPONENTIAL_BACKOFF)
  writer = (TunnelBufferedWriter) uploadSession.openBufferedWriter();
  writer.setRetryStrategy(retry);
  ```

  > **Note** We recommend that you do not adjust the preceding settings.

26.3.2. SQL

MaxCompute SQL is a structured query language whose syntax is similar to Oracle, MySQL, and Hive SQL. MaxCompute SQL can be regarded as a subset of standard SQL. However, MaxCompute SQL is not equivalent to a database, because it does not possess many characteristics that a database has, such as transactions, primary key constraints, and indexes.

MaxCompute SQL is applicable to scenarios that have large amounts of data (measured in TBs) and that do not have high real-time processing requirements. It takes a relatively long time to prepare and submit each job. Therefore, MaxCompute SQL is not optimal for services that need to process thousands of transactions per second.

26.3.3. MapReduce

MapReduce is a programming model equivalent to Hadoop MapReduce. This model is used for parallel MaxCompute operations on TB-level large-scale datasets.

You can use the MapReduce Java API to write MapReduce programs to process MaxCompute data. The Map and Reduce concepts are borrowed from functional and vector programming languages. This helps programmers run their programs on distributed systems without having to perform distributed parallel programming.

MapReduce works only after you specify a Map function and a concurrent Reduce function. The Map function maps a group of key-value pairs to another group of key-value pairs. The Reduce function ensures that all elements in the mapped key-value pairs share the same key group.

MaxCompute MapReduce has the following characteristics:

- Provides Hadoop-style MapReduce functions designed for MaxCompute (used to process tables and volumes).
- Supports the input and output of only built-in data types of MaxCompute.
- Supports the input and output of multiple tables to different partitions.
- Capable of reading resources.
- Does not allow you to use views as data inputs.
- Provides a limited sandbox security environment.

The following procedure shows how MapReduce processes data:

1. Before you perform Map operations, ensure that partition is set for the input data. The input data is divided into equally sized blocks called partitions. Each partition is processed as the input of a single Map worker so that multiple Map workers can work in parallel.
2. After partitioning, multiple Map workers start processing the data in parallel. Each Map worker reads its respective partition data, computes the data, and exports the result to Reduce.

   > **Note** When a Map worker generates data, it must specify a key for each output record. The key determines the Reduce worker for which the data entry is targeted. Multiple keys may correspond to a single Reduce worker. Data entries with the same key are sent to the same Reduce worker. A single Reduce worker may receive data entries for multiple keys.

3. Before entering the Reduce stage, the MapReduce framework sorts data based on Key values to make data entries with the same Key value adjacent. If you specify Combiner, the framework will call Combiner to combine data entries that share the same Key value.

   > **Note** You can customize the Combiner logic. Unlike the typical MapReduce framework protocol, MaxCompute requires the input and output parameters of Combiner to be consistent with those of Reduce. This process is generally called Shuffle.

4. When entering the Reduce stage, data entries with the same Key value will be in the same Reduce worker. A single Reduce worker may receive data from multiple Map workers. Each Reduce worker

performs the Reduce operation on multiple data entries with the same Key value. After the Reduce operation, all data of the same key is converted into a single value.

> **Note** This topic only provides a brief introduction to MapReduce. For more information, see related documentation.

26.3.4. Graph

Graph is the computing framework of MaxCompute designed for iterative graph processing. It provides programming interfaces similar to Pregel, allowing you to develop efficient machine learning and data mining algorithms.

Large amounts of data on the Internet is structured as graphs, such as social networking and logistics information. Graph computing models are iterative computing models. Throughout the entire computing process, multiple iterations are performed to achieve convergence. For example, for machine learning algorithms that require iterative learning model parameters, Graph is more suited than MapReduce. In common usage scenarios, you can abstract a question as a graph. Then, you can set the vertex as the center of the graph, and use supersteps for iterative updating.

MaxCompute Graph currently works in two modes:

- Offline mode: suitable for large-scale computing. Similar to MapReduce jobs, this mode involves loading and computing.
- Interactive mode: suitable for small-scale computing. You can implement a UDF and then use the command line for interaction.

In offline mode, loading and computing are independent processes. Loaded data resides in the memory. You can apply different computing logics to the loaded data. For example, the risk control department may load a set of data once a day. The operations personnel will apply different query logics to the data to view the relationships between the data.

MaxCompute Graph has been applied to many businesses in Alibaba. For example, weighted PageRank algorithms are used to compute influence metrics for Alipay users, and variational Bayesian EM models are used to predict users' car brands based on the properties of the items purchased by users.

26.3.5. Unstructured data processing (integrated computing scenarios)

Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

You can run a simple DDL statement to create an external table in MaxCompute, and associate MaxCompute tables with external data sources. This table can then act as an interface between MaxCompute and external data sources. The external table can be accessed in the same way as a MaxCompute table, and computed by MaxCompute SQL.

MaxCompute allows you to process the following data sources by creating external tables:

- Internal data sources: OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.
- External data sources: HDFS (open source), ApsaraDB for MongoDB, and Hbase.

26.3.6. Unstructured data processing in MaxCompute

MaxCompute has the following problems when processing unstructured data: MaxCompute stores data as volumes and must export generated unstructured data to an external system for processing.

To alleviate these problems, MaxCompute uses external tables to enable connections between MaxCompute and various data types. MaxCompute uses external tables to read and write data volumes as well as process unstructured data from external sources such as OSS.

26.3.7. Enhanced features

26.3.7.1. Spark on MaxCompute

26.3.7.1.1. Open-source platform - Cupid

26.3.7.1.1.1. Overview

MaxCompute is a big data solution independently developed by Alibaba Cloud that leads the industry in scale and stability. The big data open-source communities are actively developing big data solutions. All kinds of systems are rapidly emerging and growing to meet various requirements. To better serve users and to diversify the MaxCompute ecosystem, the MaxCompute team has developed the Cupid platform to connect MaxCompute with open-source communities. The Cupid platform integrates the diversity of open-source communities with the scale and stability of the Apsara system.

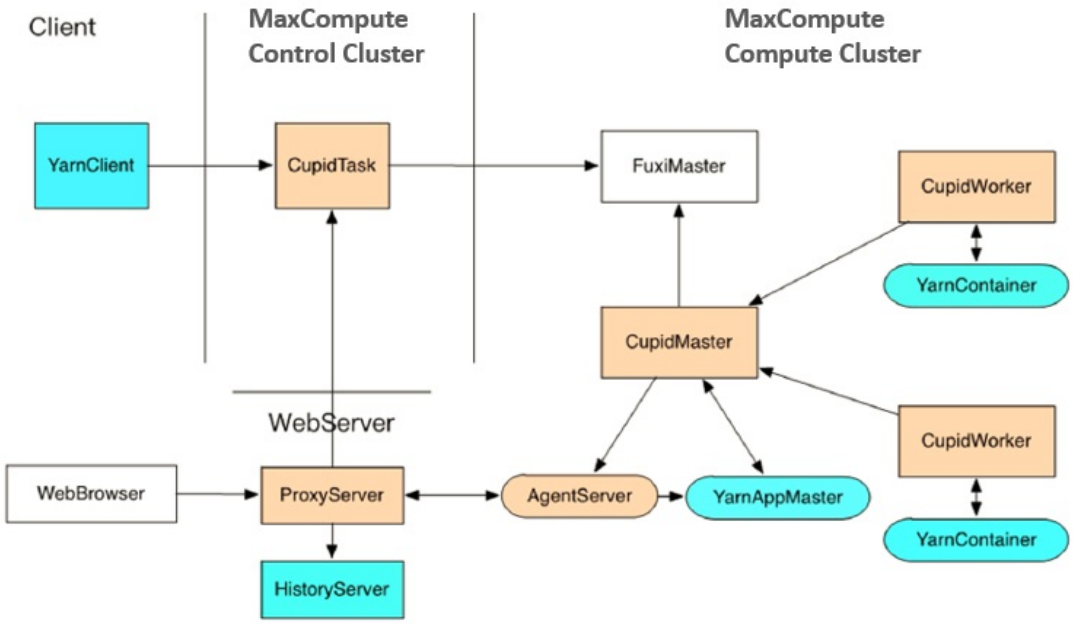
The software stacks of open-source communities and the Apsara system are similar with slight differences.

Most open-source communities use HDFS as a distributed file system, while the Apsara system uses Apsara Distributed File System. Most open-source communities use YARN as a distributed scheduling system, while the Apsara system uses Job Scheduler. On top of Job Scheduler are the computing engines designed for all kinds of scenarios. Cupid provides compatibility with open-source communities for open-source applications (such as Spark) to run on MaxCompute.

26.3.7.1.1.2. Compatibility with YARN

YARN has three application-oriented APIs: YarnClient, AMRMClient, and NMClient. YarnClient is used to submit applications to a cluster. AMRMClient is used by AppMaster to send messages to Resource Manager to request and release resources. NMClient is used to start and stop application containers.

YARN on MaxCompute



The preceding figure shows the process of submitting a YARN application to be run on MaxCompute. The yellow boxes indicate Cupid components, while the light blue boxes indicate open-source components. The procedure is as follows:

1. Use a Spark client that encapsulates the YarnClient class to access the MaxCompute control cluster and submit a job to FuxiMaster.
2. FuxiMaster starts a CupidMaster. Then, the CupidMaster starts YarnAppMaster based on the YARN protocol.
3. YarnAppMaster interacts with FuxiMaster through CupidMaster to request and release resources.
4. To start a new container, you must first use Tubo in Job Scheduler to start a CupidWorker. The CupidWorker will then start the container based on the YARN protocol.

> **Note** Typically, YarnAppMaster provides a UI. The UI is implemented through Cupid based on a proxy mechanism.

26.3.7.1.1.3. Compatibility with FileSystem

Most open-source communities use HDFS as a distributed storage solution. The FileSystem API provided by Hadoop is compatible with Alibaba Cloud OSS and Amazon S3. Apsara Distributed File System is compatible with FileSystem API. Open-source jobs submitted to MaxCompute can be run natively on Apsara Distributed File System.

> **Note** Apsara Distributed File System does not directly provide external services. The data in Apsara Distributed File System can only be used as intermediate job data, such as checkpoints. You can use OSS to make the data stored in Apsara Distributed File System accessible to other environments.

26.3.7.1.1.4. DiskDrive

Most open-source applications use local file systems for data processing, such as the shuffle and storage modules in Spark. In environments with large clusters, disks are important system resources. Disks must be centrally managed to ensure high availability, performance, and security. In the Apsara system, disks are centrally managed by Apsara Distributed File System. To provide local file system APIs based on Apsara Distributed File System, the Cupid team has designed and implemented the DiskDriverService system by integrating Web-based storage into MaxCompute.

26.3.7.1.2. Feature extensions

26.3.7.1.2.1. Overview

MaxCompute provides the Cupid framework to support open-source applications. This allows Spark to be run on MaxCompute. For ease of use and better integration with MaxCompute, there are several extensions available for Spark on MaxCompute to add features such as the secure isolation of open-source Spark applications, mutual access between MaxCompute data and Spark data, and support for interactions in multi-tenant clusters.

The following sections describe these extensions.

26.3.7.1.2.2. Security isolation

Spark jobs submitted to the MaxCompute computing cluster are run in sandboxes, preventing attacks on the system. A parent-child process architecture is used for the entire system. The Cupid framework runs in the parent process, and Spark runs in the child processes. When Spark requires access to system services, the parent process accesses the services on behalf of Spark by communicating with the child processes.

26.3.7.1.2.3. Data interconnection

An advantage of running Spark on MaxCompute is that resources used by Spark and MaxCompute jobs are shared across all clusters. This allows jobs to directly access their data without having to pull data across different clusters.

This data includes metadata and storage data. For security reasons, Spark must be authenticated through the MaxCompute account system before it can store MaxCompute data. **Spark on MaxCompute** provides OdpsRDD and OdpsDataFrame so that users can use Spark APIs on MaxCompute. Spark SQL has direct access to MaxCompute metadata for SQL optimization and can directly store and retrieve MaxCompute data at the physical layer.

26.3.7.1.2.4. Client mode

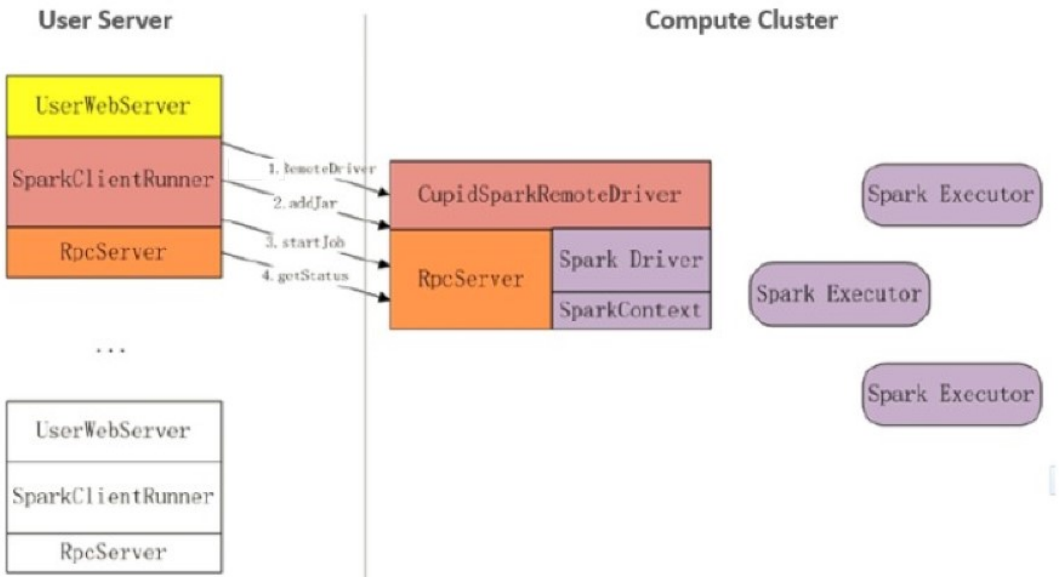
The yarn-cluster and yarn-client modes are commonly used in open-source communities for Spark-related development efforts. In yarn-cluster mode, you can submit a Spark job to a YARN cluster. After the job is run, the client generates a log that indicates the job status. In this mode, you cannot submit a job to a Spark session multiple times in real time, and the client cannot obtain the running status and result of each job. The yarn-client mode takes effect for interactive scenarios. However, to use the yarn-client mode, you need to launch the Spark driver process from the client side. You cannot use a Spark session as a service in this mode. The MaxCompute team has developed the Client mode based on **Spark on MaxCompute** to solve the preceding problems. The Client mode has the following features:

1. The client is a lightweight process that does not require you to launch the Spark driver process.
2. The client provides a set of APIs that can be used to submit jobs in real time to the same Spark session in MaxCompute clusters. It can also monitor the statuses of all jobs in the Spark session.

3. The client can build dependencies between jobs by monitoring job statuses and results.
4. You can compile an application JAR package in real time and submit it to the original Spark session through the client.
5. The client can be integrated into the Web servers of a service, and can also be scaled horizontally.

In Client mode, you need to use CupidSparkClientRunner to start a Spark session in a MaxCompute cluster. Then, you can use CupidSparkClientRunner to perform operations on the client side, such as submitting jobs and viewing the running statuses and results of the jobs. Cached data can be shared between jobs. You can also construct multiple CupidSparkClientRunner objects to interact with the same Spark session. The following figure shows the block diagram of the Spark Client mode.

Spark Client mode



The procedure for using the Spark Client mode is as follows:

1. You submit a job to a MaxCompute cluster to launch CupidSparkRemoteDriver and obtain the SparkClientRunner object.
2. You use SparkClientRunner to add the JAR package that will execute the job to RemoteDrive.
3. SparkClientRunner uses the job classname to submit the job to RemoteDriver. RemoteDriver then runs the job.
4. SparkClientRunner monitors the job status based on the job ID returned after the job is submitted.

26.3.7.1.2.5. Spark ecosystem support

The Spark ecosystem covers diverse components, including MLlib, Streaming, PySpark, SparkR, GraphX, and SQL. **Spark on MaxCompute** provides a complete Spark ecosystem that supports the scaling of original resources in open-source communities. The ecosystem provides consistent user experience with that of open-source communities. **Spark on MaxCompute** also supports access to the Spark UI and historical log files.

26.3.7.2. Elasticsearch on MaxCompute

26.3.7.2.1. Terms

term

An exact value that can be indexed. You can use a term query to search for an exact match.

text

A piece of unstructured data. Typically, a text is parsed into individual terms that are stored in an Elasticsearch index library.

cluster

A collection of one or more nodes that provide external indexing and search services. Elasticsearch is deployed in the Apsara cluster of MaxCompute. Elasticsearch clusters are a part of the Apsara cluster.

node

A logical service in an Elasticsearch cluster. A node can store data and participate in the cluster's indexing and search capabilities.

shard

A single Lucene instance which is a relatively low-level feature managed by Elasticsearch. An Elasticsearch cluster automatically manages all the shards in a cluster. When a node fails, Elasticsearch moves the shards to a different node or adds a new node.

replica

A distinct copy in Elasticsearch. Elasticsearch on MaxCompute allows you to have multiple replicas across different nodes to improve system-level availability. We recommend that you set the default number of replicas for this service to 1.

index

A collection of documents that have similar characteristics. For example, you can have an index for customer data, an index for a product catalog, and another index for order data. An index is identified by a name (that must be all lowercase) that is used to refer to the index when you perform indexing, search, update, and delete operations on the documents in the index. You can define as many indexes as you want in a single Elasticsearch cluster.

type

A logical partition of an index. You can define one or more types in an index. Typically, a type is defined as a document that has a common set of fields.

mapping

A process that defines document fields and their types as well as other index-wide settings. A mapping is similar to a schema definition in a relational database. Each index has a mapping. A mapping can either be defined in advance or automatically generated when you store a document for the first time.

document

A JSON-formatted string which is stored in Elasticsearch, similar to a row in a relational database. Each document has a type and an ID. A document is a JSON object which contains zero or more fields, or key-value pairs.

field

A simple value or a nested structure. Fields are similar to columns in relational database tables. Each field has a field type.

26.3.7.2.2. How Elasticsearch on MaxCompute works

26.3.7.2.2.1. Overview

Elasticsearch on MaxCompute is based on the open source Elasticsearch. It can run the Elasticsearch service on MaxCompute clusters.

On the MaxCompute client, you can start and manage your Elasticsearch service as needed and configure the number of nodes, disk space, memory size, and custom settings. The resources consumed by the Elasticsearch service are counted against your MaxCompute quota.

The following sections describe how **Elasticsearch on MaxCompute** functions work.

26.3.7.2.2.2. How distributed architecture works

Basic principles

An Elasticsearch cluster consists of multiple nodes. MaxCompute ensures high availability by controlling the start and stop of Elasticsearch services and nodes, allocating computing resources, and implementing failover based on a centralized scheduling mechanism.

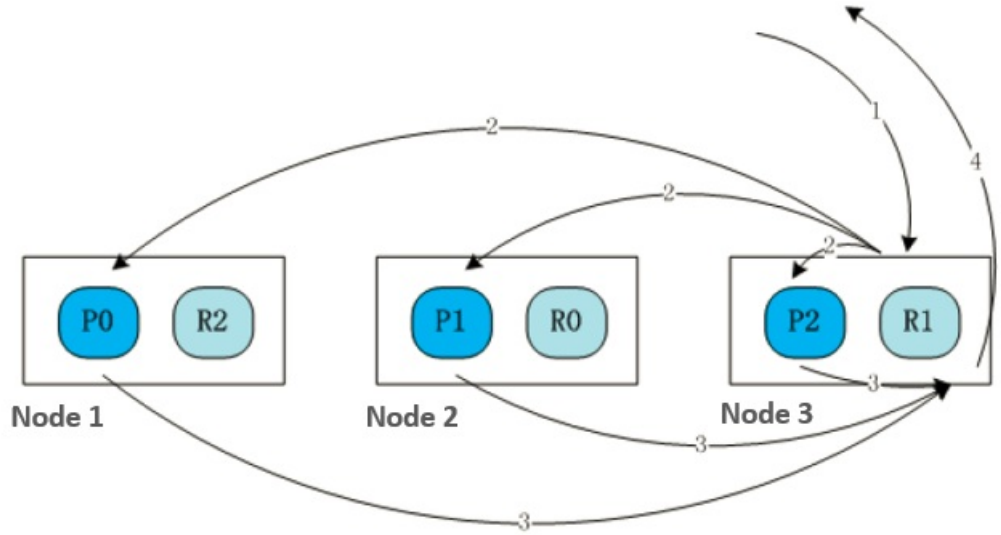
Data is replicated into multiple copies and stored in Apsara Distributed File System. This guarantees that no data is lost due to the failure of a few nodes.

An index is split into multiple shards, which are evenly distributed across multiple nodes in a cluster. The system simultaneously retrieves data shards in multiple nodes, improving retrieval performance.

Implementation process

The following figure shows the distributed retrieval workflow.

Distributed retrieval workflow



As shown in the preceding figure, each cluster consists of three nodes. The index has three shards: P0, P1, and P2. These shards are distributed across the three nodes. Each shard is replicated in 1:1 mode, generating three replicas: R0, R1, and R2. The retrieval process is as follows:

1. A user sends a retrieval request to Node 3.
2. After receiving the request, Node 3 sends a retrieval request (2) to P0, P1, and P2 based on the recorded index shard information.
3. The nodes where P0, P1, and P2 are located search for the requested information in the specified shards. A retrieval result message (3) is sent to Node 3.
4. Node 3 collects the retrieval results from other nodes and returns the retrieval results to the user in an acknowledgment message (4).

When multiple nodes are performing data retrieval at the same time, the retrieval speed is improved. The performance of distributed retrieval increases with the number of nodes.

26.3.7.2.2.3. How full-text retrieval works

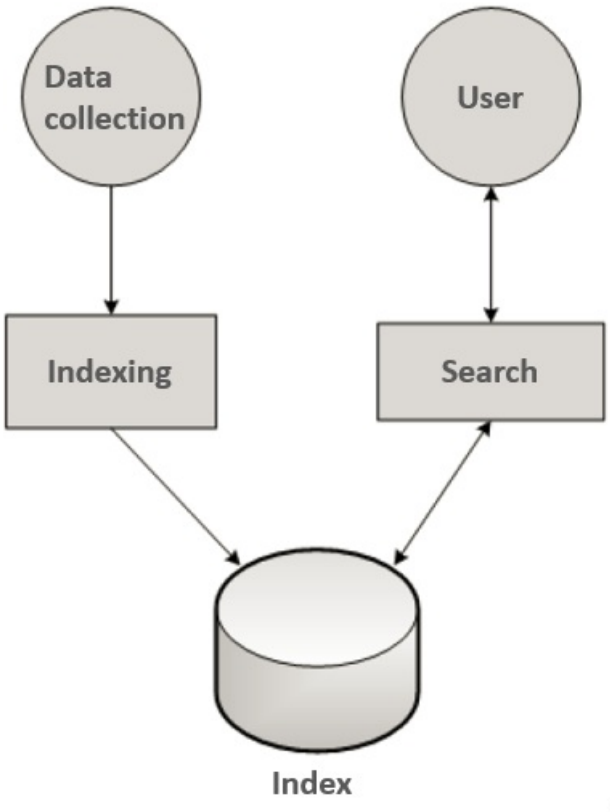
Basic principles

Full-text retrieval refers to techniques used to search for data records containing specified contents from large volumes of texts. In the retrieval process, data in texts is segmented by words, and an inverted index is created based on mappings from words to documents to allow fast document retrieval.

Implementation process

The following figure shows the full-text retrieval process.

Full-text retrieval process



The retrieval process is as follows:

1. The data collection module collects structured and unstructured data, converts the data into the field + value format, and submits the data to the indexing module.
2. The indexing module segments the data, creates inverted indexes based on a predefined indexing method, and saves the indexes. The field type, indexing method, and segmentation rules are configured on the retrieval management page.
3. The search module receives and processes user requests. Requests are parsed to obtain indexes, fields, and query statements, and then matched to records in the inverted indexes.
4. The indexing module returns data that meets user-defined requirements such as sorting rules and request quantity.

26.3.7.2.2.4. How authentication control works

Basic principles

Authentication control is implemented at the entrance used for external services to check whether users have been authorized to access the index libraries.

Implementation process

The authentication control process is as follows:

1. **Elasticsearch on MaxCompute** provides retrieval management and O&M platforms that are only accessible after logon. User account information is verified and authenticated by a centralized authentication module before logon. Any user who fails the authentication is denied access to the platforms.

2. The administrator can use the MaxCompute client to add Elasticsearch users and configure permissions for the users.
3. The system authenticates all users who attempt to access index libraries. After passing authentication, you will be able to retrieve or perform operations on data in the libraries.

26.3.8. Multi-region deployment of MaxCompute

MaxCompute can be deployed across regions. Control clusters are deployed in a unified manner and are used to configure resources and manage computing tasks. Compute clusters are separately deployed in each region to create projects and distribute computing tasks.

The multi-region deployment of MaxCompute has the following features:

- One MaxCompute service can manage multiple clusters in different regions.
- Data exchanges between clusters are completed within MaxCompute, and data replication and synchronization between clusters are managed based on configured policies.
- Metadata is stored in a centralized manner. Therefore, the infrastructure requirements, such as the network connections of different data centers, are relatively high.
- A unified account system is required.
- The development systems for big data applications, such as DataWorks, are used for all clusters in all regions.
- MaxCompute must run in multi-cluster mode to support multi-region deployment.

  > **Note** Take note the following conditions and limits on changes to the cluster mode:
  >
  > - The network bandwidth must be sufficient to support multi-region data synchronization and link redundancy.
  > - Control clusters in the central region have a high latency for basic services such as Alibaba Cloud DNS and Tablestore. Therefore, we recommend that you deploy basic services in the same data center to ensure that network latency remains within 5 ms.
  > - The network latency between control clusters in the central region and compute clusters in other regions is within 20 ms.
  > - Clocks must be synchronized between clusters in different regions and between machines in the same cluster.
  > - The network bandwidth must be sufficient to support data replication among clusters.
  > - Alibaba Cloud DNS is required.
  > - Machines in different clusters can communicate with each other, and the clusters have similar network infrastructure (1-Gigabit or 10-Gigabit).

- The O&M and upgrades for multi-region deployment are different from those for single-cluster deployment. Multi-region deployment requires higher on-site O&M capabilities.
- MaxCompute supports cross-region multi-cluster (sub data centers) distributed computing. It uses the global job scheduling feature of the primary data center to balance the resource usage among clusters. It schedules jobs to the most appropriate cluster based on cluster information, such as the default settings, historical analysis, data distribution, and cluster load. Then, it executes the jobs and generates the query results. MaxCompute supports history- and cost-based optimization policies of SQL queries.

27.DataWorks

27.1. What is DataWorks?

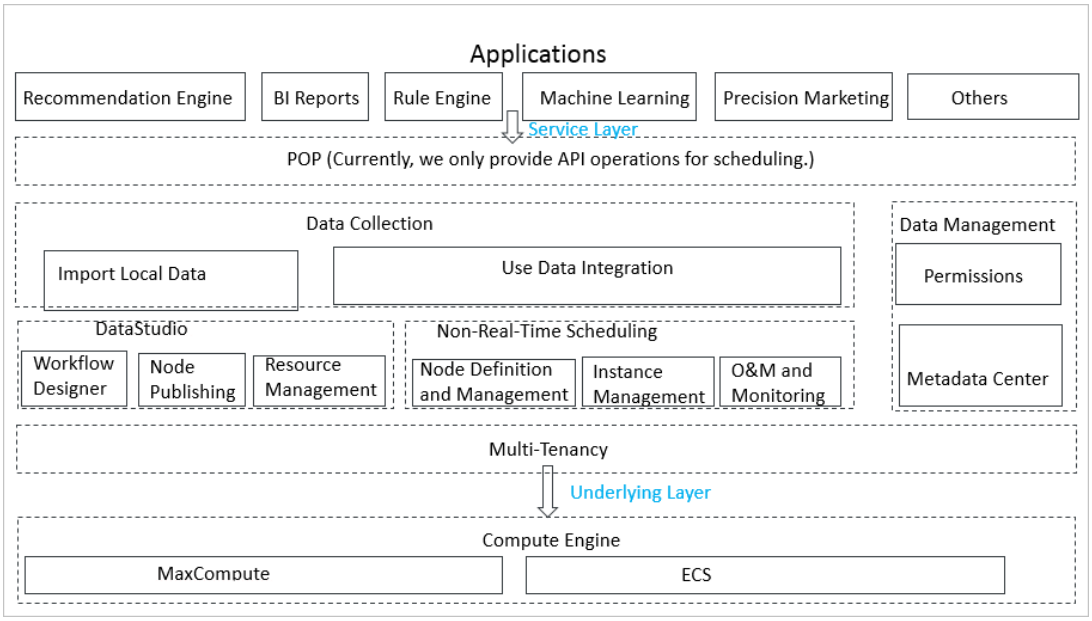
27.1.1. Overview

DataWorks is an all-in-one big data analytics and governance service released by Alibaba Cloud. It provides end-to-end solutions for enterprises and individual users to analyze, manage, schedule, govern, and apply data.

DataWorks is aimed at mining the full value of the data.

- It allows large enterprises to build petabyte-level and even exabyte-level data warehouses. The enterprises can improve their business operations by using the data integration, data asset management, and data analytics features provided by DataWorks.
- Small- and medium-sized enterprises and individual users can build data-based applications, which drive data service innovations.

Service components



DataWorks consists of an integrated development environment (IDE), a scheduling system, a data integration tool, and a data management system.

- IDE: a development tool that can be used to write SQL, MapReduce, or shell code. IDE supports collaborative development and version control. By using the visual process design tool, you can define the dependencies among different nodes.
- Scheduling system: a system that can schedule millions of batch sync nodes in a day. You can manage your nodes online, and view the logs, scheduling status, and monitoring alerts.
- Data integration tool: an integration tool that can be used to configure sync nodes between heterogeneous data stores. More than 80% of databases and storage systems provided by Alibaba Cloud and common data stores such as relational databases, FTP, and Hadoop Distributed File System (HDFS) can be configured as a source or a destination of the sync node. You can also create a node that runs periodically to synchronize data on a periodic basis.

- Data management system: a system that can be used to manage data in MaxCompute and E-MapReduce compute engines. You can manage permissions, view the data lineage, and view the metadata.

27.1.2. Scenarios

DataWorks can be applied to the construction of large data warehouses and data-driven operations.

Construction of large data warehouses

Enterprises can use DataWorks in Apsara Stack to build large data warehouses.

DataWorks can integrate petabytes of data for enterprise customers.

- Storage: provides a scalable data warehouse for petabytes and exabytes of data.
- Data integration: supports data synchronization and integration across heterogeneous data stores to eliminate data silos.
- Data analytics: supports MaxCompute-based big data processing capabilities, programming frameworks such as SQL and MapReduce, and a visualized workflow designer.
- Data management: supports unified metadata management and permission-based data access control.
- Batch scheduling: provides the scheduling of recurring nodes at different intervals, and supports scheduling of millions of concurrent nodes, error alerts, and real-time monitoring of running node instances.

Data-driven operations

- Innovative businesses: Data mining, data modeling, and real-time decision making can be implemented based on big data analytics results provided by DataWorks.
- Small- and medium-sized enterprises: DataWorks allows you to analyze data and put it to commercial use, which help enterprises generate marketing strategies.

27.2. Benefits

This topic describes the benefits of DataWorks.

Capability of processing big data

DataWorks uses MaxCompute as its compute engine, which supports a maximum of 5,000 servers in a single cluster. DataWorks can access data from different clusters, which allows you to process your big data. The offline scheduling system can run millions of concurrent nodes. You can also configure rules and alerts to monitor the running of nodes in real time.

Core capabilities:

- Supports join operations for trillions of records, millions of concurrent nodes, and I/O throughput of up to multiple petabytes each day.
- Allows you to share data across clusters and data centers, and scale out clusters to a maximum of tens of thousands.
- Provides efficient and easy-to-use SQL and MapReduce engines, and supports most standard SQL syntax.
- Protects user data from loss, breach, or theft by using multi-layer data storage and access security mechanisms of MaxCompute, including triplicate backups, read/write request authentication,

application sandboxes, and system sandboxes.

Integrated data processing environment

DataWorks integrates development, scheduling, monitoring, and alerting for nodes, and management of data.

Core capabilities:

- Provides you with all the required features for data processing.
- Provides a visual designer similar to Kettle for you to design and edit workflows.
- Provides a collaborative development environment. You can create and assign roles for varying nodes, such as development, online scheduling, maintenance, and data permission management, without locally processing data and nodes.

Integration from disparate data stores

DataWorks supports reading data from 11 disparate data stores and writing data to 12 disparate data stores. You can also configure dirty data filtering and bandwidth throttling.

Core capabilities:

- Supports reading data from data stores of the following types: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB for RDS, DRDS, MaxCompute, FTP, Object Storage Service (OSS), HDFS, Dameng, and Sybase.
- Supports writing data to data stores of the following types: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB for RDS, DRDS, MaxCompute, AnalyticDB, Memcache, OSS, HDFS, Dameng, and Sybase.
- Supports dirty data filtering and bandwidth throttling.
- Supports recurring nodes, including recurring sync nodes.

Web-based software

You can use DataWorks whenever an internal network or the Internet is available.

Multi-tenancy

DataWorks uses multi-tenancy to isolate data among tenants. Each tenant separately manages their own permissions, data, resources, and members.

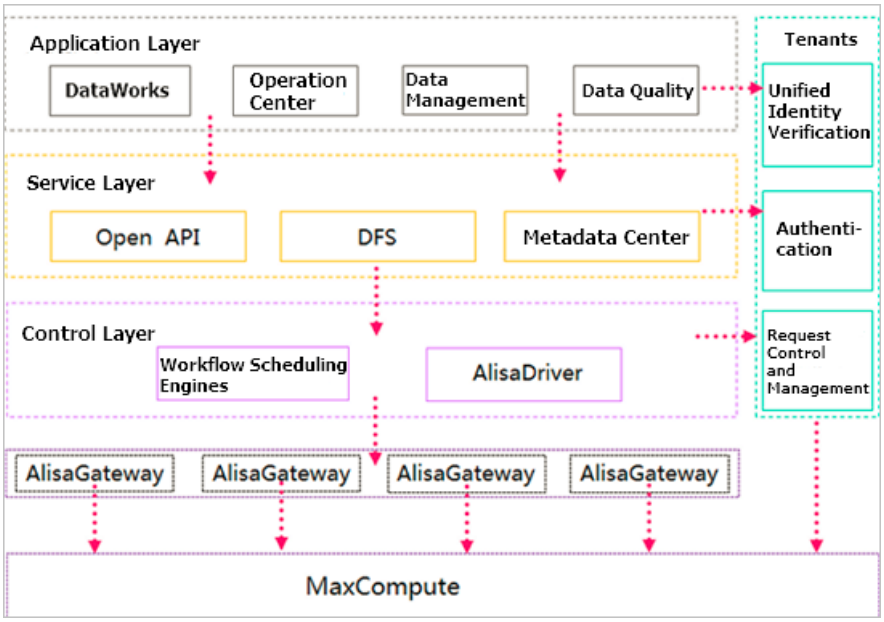
Open platform

DataWorks provides all modules as components and services. You can use the DataWorks API to develop extra features of DataWorks.

27.3. Architecture

This topic describes the system architecture, security architecture, and multi-tenancy model of DataWorks.

System architecture



DataWorks adopts the design of components and services, and consists of the following three layers:

- Control layer: the core of batch data processing in DataWorks. The workflow scheduling engine generates and runs node instances. AlisaDriver coordinates and controls the running of all nodes.
- Service layer: provides services for the application layer and other external applications.
- Application layer: runs on top of the service layer, and provides the graphical interface for user interactions.

Security architecture

The security architecture of DataWorks features error proofing, basic security, and optional security tools.

- Error proofing ensures proper running of DataWorks during coding, deployment, and configuration.
- Basic security ensures the security of data for DataWorks by using features such as resource isolation among tenants, user identity verification, authentication, and log auditing.
- Optional security tools in DataWorks allow you to customize security policies for the protection and management of your system and data.

Multi-tenancy

DataWorks adopts a multi-tenancy model.

- Storage and computing resources are scalable. You can manage your own resources and request resource quotas as needed.
- Tenants are isolated. Each tenant separately manages its own data, permissions, accounts, and roles.

27.4. Services

27.4.1. DataStudio

DataWorks DataStudio provides an all-in-one IDE. In DataStudio, you can build data warehouse models, query data, develop the extract-transform-load (ETL) process, and develop algorithms. In addition, it supports collaborative development and file version control.

Features

- Provides a visual workflow designer similar to Kettle for you to design workflows and manage nodes in each workflow.
- Supports the upload of local files.
- Supports data integration from heterogeneous data stores.

  > Note
  >
  > Data sync nodes support the following data stores:
  >
  > - Sync nodes can read data from the following data stores: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB for RDS, DRDS, MaxCompute, FTP, OSS, HDFS, Dameng, and Sybase.
  > - Sync nodes can write data to the following data stores: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB for RDS, DRDS, MaxCompute, AnalyticDB, Memcache, OSS, HDFS, Dameng, and Sybase.

- Provides a web-based programming and debugging environment that allows you to create SQL, MapReduce, shell (limited support), and data sync nodes.
- Supports node deployment across MaxCompute projects. You can deploy nodes and code to the scheduling system across different workspaces.
- Adopts version control, node locking, and conflict detection mechanisms to facilitate collaborative development.
- Allows you to search for and use MaxCompute tables, resources, and user-defined functions (UDFs).

27.4.2. Data Map

Developed based on Data Management, Data Map uses roles to control the permissions for using different features, such as the permissions for creating and previewing data. Data Map helps you build a better enterprise-level knowledge base.

Data Map allows you to query tables, view details of tables, and manage permissions on tables. You can also add tables to your favorites. For information about Data Map, see Data Governance > Data Map in *DataWorks User Guide*.

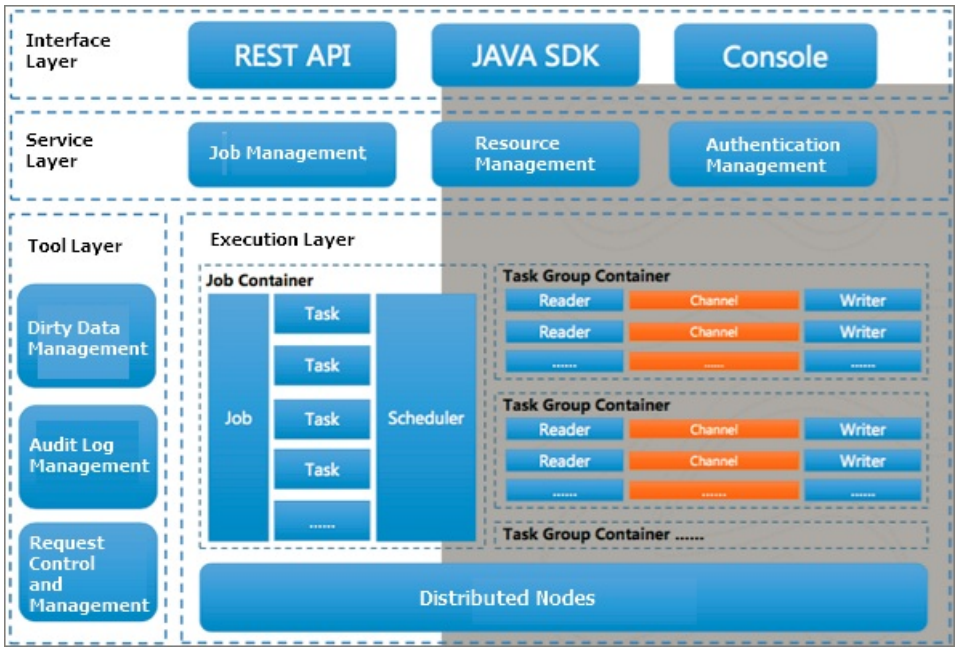
27.4.3. Data Integration

Data Integration is a data synchronization platform that provides stable, efficient, and scalable services. It provides transmission channels for batch data stored in MaxCompute, AnalyticDB, and Realtime Compute. Data Integration implements fast integration on data from heterogeneous data stores.

Data Integration adopts the framework and plug-in model. The framework is used for common operations in data synchronization and transmission. The plug-ins are used to read and write data. Data Integration supports the following plug-ins:

- Reader: reads data from data stores.
- Writer: writes data to data stores.

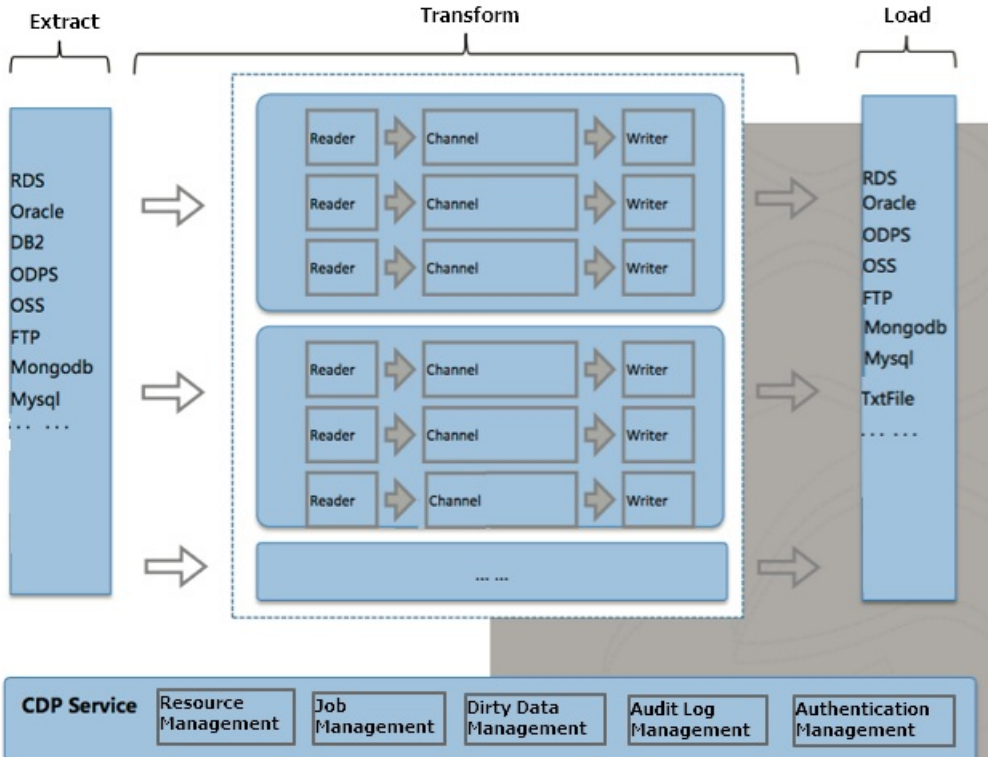
You can develop readers and writers for Data Integration to support more data stores.



Data Integration consists of the interface layer, service layer, and tool and execution layer.

- The interface layer provides three methods of using the Data Integration service: RESTful API, Java SDK, and console.
  - The RESTful API method can be used in multiple language environments. If you are a Java developer, we recommend that you use the Java SDK method to avoid manual processing of authentication, authorization, and underlying HTTP calls.
  - The console is developed based on the command-line tool, which allows you to use the majority of Data Integration functionalities.
  - Data Integration provides a web interface that is developed based on the RESTful API, which is recommended for developers.
- The service layer includes resource management, node management, and authentication management. For more information, see the service overview.
- The tool and execution layer is the core of Data Integration. This layer implements the ETL process. All sync nodes that are committed to Data Integration are run on the execution layer. The execution layer uses DataX as the synchronization engine.

  ETL process



Features

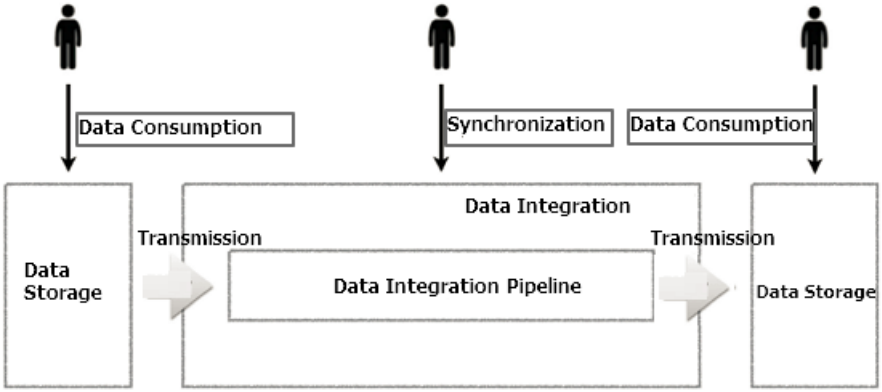
- Various types of data stores
  - Relational databases: MySQL, SQL Server, PostgreSQL, DRDS, Oracle, and general relational databases
  - NoSQL databases: Tablestore and Memcache
  - Big data storage systems: MaxCompute and AnalyticDB for MySQL
  - Semi-structured storage systems: OSS, HDFS, and FTP

  You can use the following Java Database Connectivity (JDBC) URLs when you configure connections to general relational databases such as Dameng, Db2, and PPAS:

  - Dameng: jdbc:dm://ip:port/database
  - Db2: jdbc:db2://ip:port/database
  - PPAS: jdbc:edb://ip:port/database

  Data Integration supports periodic batch synchronization. For example, you can configure a sync node that runs on a daily, weekly, or monthly basis. When the batch sync node starts, a snapshot of source data is taken. The system then reads data from the snapshot and writes the data to the destination data store. Each batch sync node has a lifecycle.

Data Integration processes only data synchronization and transmission. The complete transmission process is under the control of the Data Integration synchronization cluster model. The channels and data flows involved in the synchronization processes are isolated from users. Data Integration does not provide an API for data analysis. To perform data analysis, use DataStudio.



- Consistent data quality
  - Supports conversions between different data types.
  - Accurately identifies, filters, collects, and displays dirty data to ensure the quality of data.
  - Supports node performance reporting, which helps you track node status, such as data volume and dirty data.
- Efficient data transmission
  - Supports one-way data channels, and allows a single process to reach the maximum data transfer rate up to 200 Mbit/s on each server.
  - Adopts a distributed architecture and supports transmission for gigabytes to terabytes of data.
- User-friendly control experience
  - Implements accurate control of channels, record streams, and byte streams.
  - Allows you to rerun any threads, processes, and tasks that fail.
- Clear core design
  - Provides a professional framework and an efficient execution engine. The engine supports common plug-ins, standardizes the process of developing plug-ins, and automatically detects new plug-ins.
  - Provides clearly defined and easy-to-use plug-ins that allow developers to focus on the business instead of the framework.

27.4.4. Tenant management

You can manage workspaces, members, and permissions.

- Workspace configuration

  The Project Management page displays basic workspace settings.
  - Sandbox whitelist: Configure the IP addresses and domains that can access the workspace.
  - Compute engine: View the information about existing compute engines.
- Member management

On the Members page, you can assign or revoke a role from specified members.

- Permission management

  On the Permissions page, you can view the system permissions and their categories.

27.4.5. Data Quality

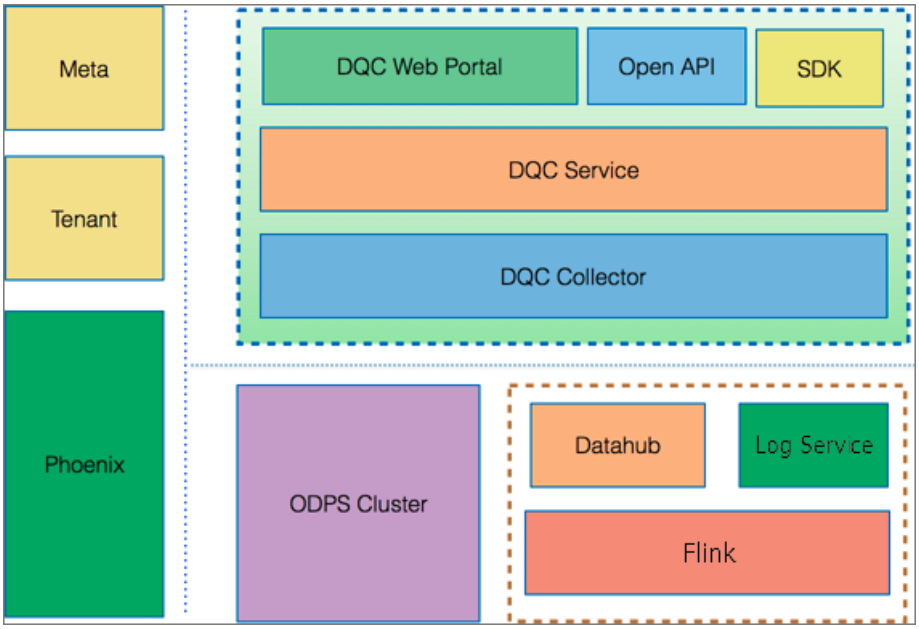
27.4.5.1. Overview

Data Quality is a platform that provides data quality check and management services. You can use it to monitor both real-time and batch data during the entire data processing cycle.

When you use Data Quality to monitor real-time data, it can detect discontinuity, delay, and other user-defined data issues in data streams. When you use Data Quality to monitor batch data, it can detect abnormal data in the production process, protect downstream data from being affected by abnormal data, and promptly notify you about the abnormal data. This helps ensure the correctness of your data.

Data Quality requires the access to the metadata, fields, and tables, and requires user and tenant management. In the scenario of monitoring batch data, Data Quality uses MaxCompute as the compute engine. In the scenario of monitoring real-time data, Data Quality uses the Flink framework as the streaming data processing tool. Data Quality consists of three components: the web portal, the check service, and the data collection service.

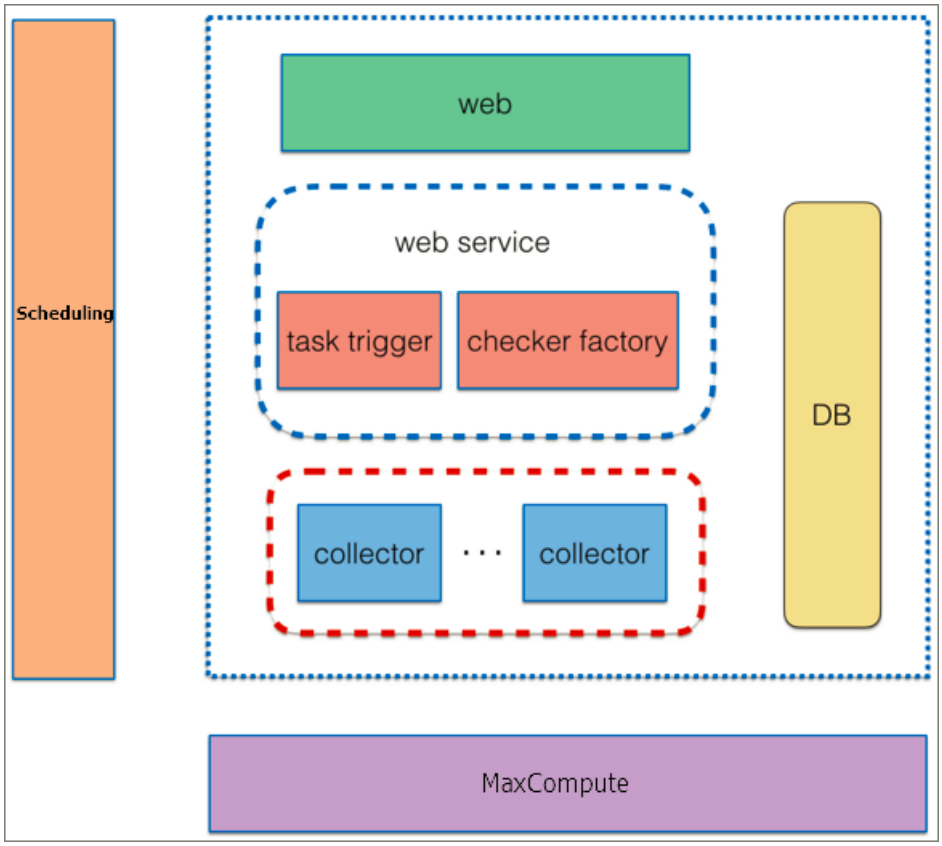
Data Quality architecture



27.4.5.2. Use Data Quality to monitor batch data

This topic describes the architecture, working principles, and benefits of using Data Quality to monitor batch data.

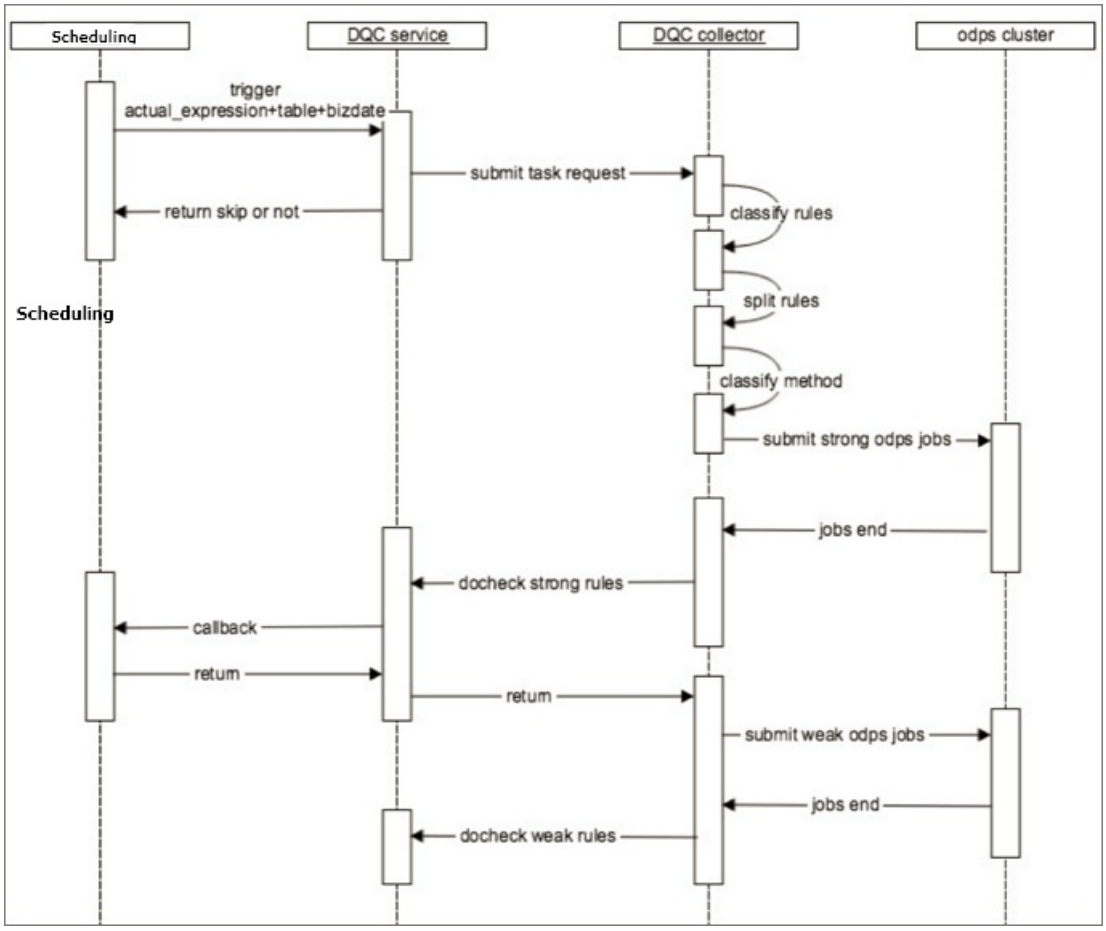
Architecture



Data Quality consists of the web UI, web service, and collector modules.

- Web UI: provides a graphical interface for user interactions. It provides features such as rule management, search by node, subscription management, dashboard, permission control, and cache management.
- Web service: provides access to databases, checks data quality, parses nodes, and triggers nodes. The checker factory module checks samples by using quality check logic such as comparison of fixed value, fluctuation, and variance detection.
- Collector: consists of multiple data collection engines that obtain data samples based on user specified rules. Data collection engines classify the rules based on potency, rule types, and sampling methods. Before the data collection engines send the rules to MaxCompute to obtain data samples, the data collection engines apply logical splitting and combination to the rules.

How it works



Data Quality monitors batch data in the following way:

1. The scheduling system sends a request that triggers the service module to check the quality of data in the specified partitions of a table. The request contains the partition expression, table information, and node schedule.
2. Based on the partition expression, a server in the service module obtains the set of rules that are applied to the current node. The server submits a request for obtaining data samples to data collection engines and returns the request result to the scheduling system. The scheduling system first allocates resources to run nodes that are associated with strong rules.
3. The data collection engines further classify the set of rules based on potency, rule types, and sampling methods. The MaxCompute cluster collects data samples based on the sampling methods.
4. After the data collection engines finish data sampling based on strong rules, the data collection engines instruct the service module to check data quality. After the quality check, the service module sends the check results to the scheduling system, and the scheduling system determines whether to block the node.
5. After the quality check by using strong rules, the service module returns the results to the data collection engines. The data collection engines continue the sampling process, and send the processed data for check based on weak rules. After the weak rule check is complete, the quality check ends.

Benefits

- Data Quality provides built-in rule templates and comprehensive data quality metrics.

The templates support filed and table level rules with a fluctuation threshold or fixed value comparison. You can create rules from the templates to check whether data entries are null or unique or use discrete values, the maximum, minimum, average, or sum to evaluate the data quality. You can also create custom rules for special requirements.

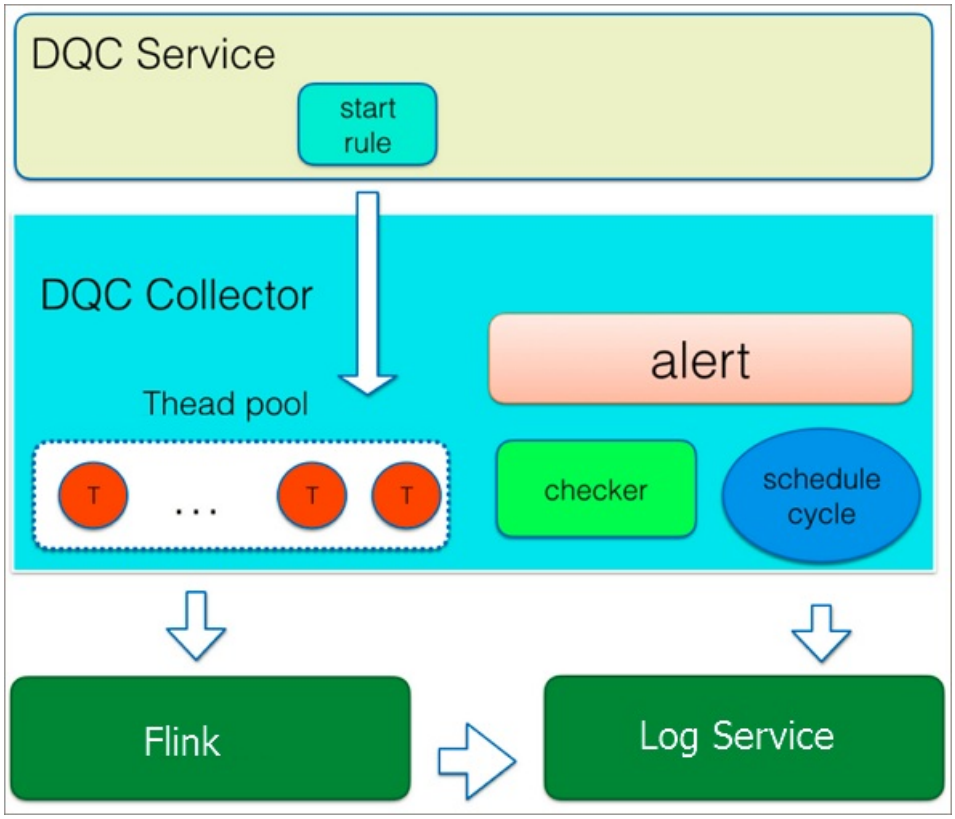
- Data Quality clusters are horizontally scalable. You can add servers if Data Quality reaches the maximum concurrency. Data Quality also includes a reliable fault-tolerance system that ensures that data collection tasks are accurate and consistent.
- Data Quality supports rule classification based on potency and severity levels.

  When you use Data Quality to monitor batch data, you can classify rules into weak and strong rules based on potency. You can also set thresholds to reflect the warning and error severity levels of check results based on the deviation from the expected value. When strong rule check results show a significant deviation from expected values, the node is blocked to protect downstream data against dirty data. This ensures the correctness of data during the data processing cycle.
- Data Quality provides a potency-based execution mechanism that first runs the nodes that are associated with strong rules. The data collection engine supports running nodes based on the potency.
  - If available resources are limited, this mechanism ensures that you first run nodes that are associated with strong rules.
  - If available resources are sufficient, this mechanism allows nodes that are associated with weak rules to run.

27.4.5.3. Use Data Quality to monitor real-time data

This topic describes the architecture, working principles, and benefits of using Data Quality to monitor real-time data.

Architecture



Rules for monitoring real-time data are converted into Flink SQL statements. Data Quality uses Flink to read data from DataHub and write check results to Log Service. The collector module of Data Quality regularly obtains abnormal data from Log Service, writes the data to Redis, and then triggers alerts. The service module of Data Quality synchronizes the alerts from Redis to other databases for users to query.

How it works

Data Quality monitors real-time data in the following way:

1. After you enable a rule, the service module creates a Logstore. The service module uses an SQL parser to declare a dimension table used for referencing a DataHub topic. The service module uses a rule converter to generate a CREATE TABLE statement and combine table operations. Then, the service module submits a Flink node and updates the next quality check time.
2. One of the servers in the service module first establishes a lock to serve as the master. The master collects data from DataHub topics on a regular basis and sends the data to the collector module for quality check.
3. The collector module uses a LogHub consumer to subscribe to the Logstore. Then, the collector module writes abnormal data to Redis, and determines whether to send alerts.
4. The service module starts the Quartz scheduler worker service, and writes the data from Redis to another database for users to query.

Benefits

- Monitors data discontinuity and latency in real time in multiple scenarios. It can join multiple streams and dimension tables from one data store, and allows you to write Flink SQL statements to define your own business rules.
- Supports monitoring on data latency at the level of seconds.

- Allows you to set thresholds at the warning and error severities. This helps you identify the deviation of check results from expected values.
- Allows you to set the minimum alert interval and the number of alerts to reduce redundant notifications.
- Provides you with more reliable alert information because raw alerts are Hash de-duplicated. This ensures the idempotence of data during the real-time computing process and avoids repeated notifications.

27.4.6. Data Asset Management

Data Asset Management allows you to view the metadata collected in Data Map. You can also modify the categories for the metadata, add business descriptions to tables, and view the metadata.

Data Asset Management provides you with an overview of your data assets. It requires that data be synchronized by using Data Integration and processed by using DataStudio before you manage your tables and API operations stored in your business system and DataWorks.

27.4.7. Real-time analysis

The real-time analysis feature allows you to query and preview data. This feature is suitable for data analysis and data exploration.

You can create, rename, and delete directories and files.

1. Click the **Run** icon to run the SQL statements.
2. View the running result.

27.4.8. DataService Studio

DataService Studio supports API hosting, authentication, authorization, and management. You can create APIs for tables and publish the APIs by using the API Gateway service.

DataService Studio provides the following features:

- Supports various data stores, including relational databases, AnalyticDB, and NoSQL databases.

  Supported data stores: MySQL, Oracle, SQL Server, PostgreSQL, ApsaraDB for RDS, DRDS, AnalyticDB, Tablestore, MongoDB, and Lightning.
- Provides the codeless UI, which can be used to generate APIs without writing code.
- Provides the code editor, which can be used to create APIs by writing SQL statements.
- Provides accurate access control. You can customize permissions on APIs, table rows, and table columns.
- Supports calling API operations by using API Gateway or HTTP requests.
- Supports a variety of network environments, including the local private network, Virtual Private Cloud (VPC), and classic network.
- Allows you to manage APIs such as managing API groups and APIs, and publishing and removing APIs.
- Supports API isolation by workspace or tenant.
- Allows you to register, manage, and display APIs.
- Supports a variety of API execution environments, including standalone environments and the EAS container service.

- Supports debugging APIs online. You can view the API call information and the performance in real time.

27.4.9. Intelligent Monitor

Intelligent Monitor is a system that monitors and analyzes nodes in DataWorks. Intelligent Monitor sends alerts based on specified rules, times, methods, and alert contacts. It automatically selects the most appropriate alerting time, notification methods, and alert contacts.

Intelligent Monitor has the following benefits:

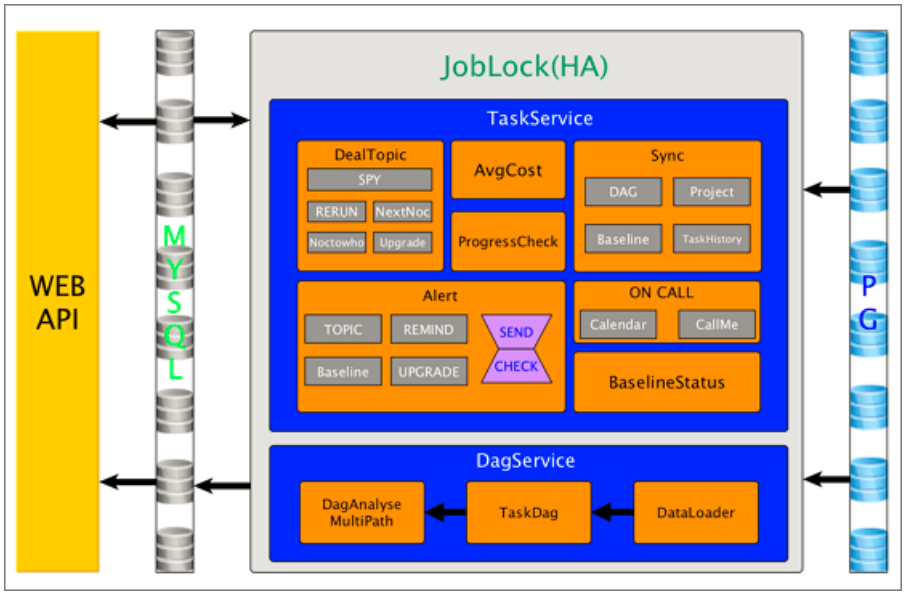
- Improves your efficiency on configuring alert triggers.
- Prevents invalid alerts from bothering you.
- Automatically covers all important nodes for you.

A conventional monitoring system allows you to configure monitoring rules, but cannot meet the requirement of DataWorks due to the following causes:

- DataWorks has numerous nodes, so it is difficult for you to find out the nodes to be monitored. Dependencies between the nodes are complex. Even if you know the most important nodes, it is difficult to find out all ancestor nodes of these nodes and monitor them all. In this case, if you monitor all nodes, a large number of invalid alerts may be generated. In consequence, you may miss those useful alerts.
- The alerting method varies with monitored nodes. For example, some monitoring tasks require the relevant nodes to run for more than 1 hour before alerts are triggered, whereas other monitoring tasks require the relevant nodes to run for more than 2 hours. It is complex to set an alerting method for each node separately, and it is difficult to predict the alert threshold value for each node.
- The alerting time varies with monitored nodes. For example, alerts for unimportant nodes can be reported after the working hours start in the morning but alerts for important nodes must be immediately reported even in off hours. It is hard for a conventional monitoring system to distinguish the importance of nodes.
- Different alerts require different operations to turn off.

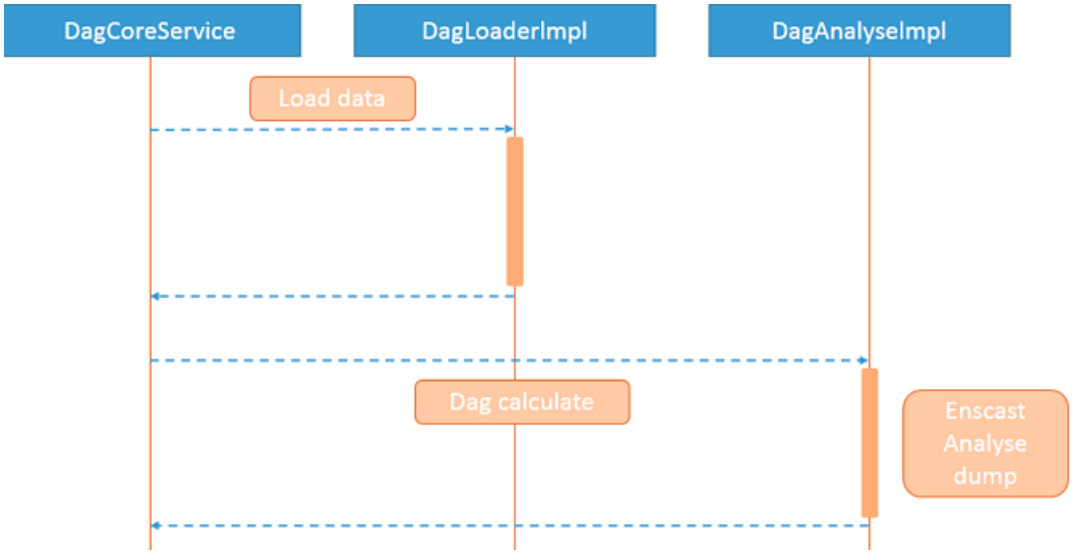
Intelligent Monitor provides comprehensive monitoring and alerting logic. You only need to provide the names of important nodes in your business. Then, Intelligent Monitor automatically monitors the entire running process of your nodes and creates standard alert triggers for them. In addition, you can customize alert triggers by completing basic settings.

Architecture



- DagService: analyzes all nodes in each directed acyclic graph (DAG) based on the baseline settings. DagService then collects information such as the estimated completion time, the key path, the required completion time, and whether to suspend a node. The information collected by DagService provides the basis for TaskService.
- TaskService: runs different nodes based on the information provided by DagService, including estimating the completion time, acquiring and fixing events, and customizing baseline alerts.
- WebService: provides the HTTP API that can be called to send requests. You can call API operations to view the Intelligent Monitor information, such as baseline instances, alert information, events, and gantt charts.

How it works



DagService collects the information of all nodes on each DAG based on the baselines and the average running time of each node. The information contains the estimated completion time, the required completion time, the key path, whether to suspend a node, and whether the node is a child of a suspended node.

TaskService runs nodes based on the node configuration and the information provided by DagService. The database lock ensures that one node is run by only one server. When a server is down, another server takes over the node, which ensures the high availability of the monitoring service.

27.4.10. Scheduling system

27.4.10.1. Overview

The scheduling system is one of the core systems in DataWorks. It is responsible for scheduling all batch sync nodes based on the specified time and the dependencies. The scheduling system provides the following features:

- Schedules millions of nodes.
- Adopts a distributed execution architecture so that the number of concurrent nodes can be linearly expanded.
- Supports different granularities for the scheduling interval, such as minute, hour, day, week, month, and year.
- Supports same-cycle dependency, cross-cycle dependency, and self-dependency between nodes.
- Supports special operations such as dry runs, node suspension, and one-off nodes.
- Allows you to create and run an ad-hoc workflow.
- Displays a workflow in a DAG, which provides you with a clear view for O&M.
- Supports real-time monitoring and alters. Alerts can be sent by text message and email.
- Supports administrative operations such as rerunning a node or multiple nodes at a time, terminating processes, setting the node status to Successful, and suspending nodes.
- Generates retroactive data for multi-cycle instances that are run in sequence.
- Provides an interface that displays the summary of global node details, including the number of scheduled nodes, the number of failed nodes, the number of running nodes, top 10 scheduled nodes by computing resource consumption, top 10 scheduled nodes by execution duration, and node distribution by type.

27.4.10.2. Terms

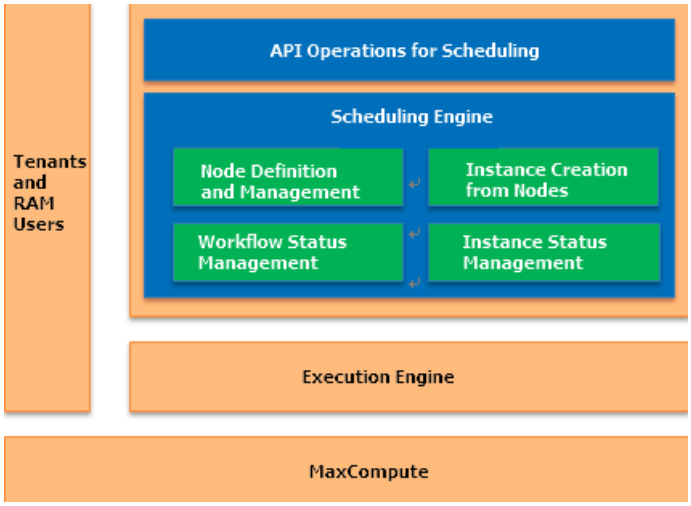
This topic describes the terms of the scheduling system.

- Node: A node represents a batch synchronization task in the scheduling system. Node properties include the node type, the code version, the specified time for running the node, and the dependencies between nodes.
- Instance: An instance is generated each time a node is run in the scheduling system to track the running of the node. An instance contains the runtime information such as the instance status and the time when the status changes.
- Workflow: A workflow is composed of several interdependent instances. The scheduling system consolidates all instances in a day into a workflow for unified management. A workflow has its own status, which is determined by the status of each instance in the workflow.

27.4.10.3. Architecture

This topic describes the architecture of the scheduling system.

The following figure shows the architecture of the scheduling system and its relationship with other systems.



The scheduling engine is the core of the scheduling system. It contains four modules.

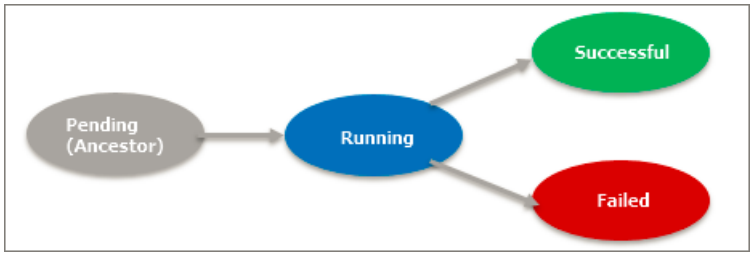
- The node definition and management module maintains node definitions submitted by users, including the code, the specified time for running the node, and the dependencies. An instance is generated from the node configurations at a fixed time every day.
- The instance state management module manages the state changes after an instance runs.
- The workflow state management module maintains the state changes after a workflow runs. A workflow is a set of instances with dependencies.
- The scheduling system allows other systems to add, delete, modify, and query its internal scheduling data by calling API operations.

The resources that are used by the scheduling system are isolated among tenants. Before a node instance runs, the scheduling system schedules the instance to the execution engine.

27.4.10.4. State machines

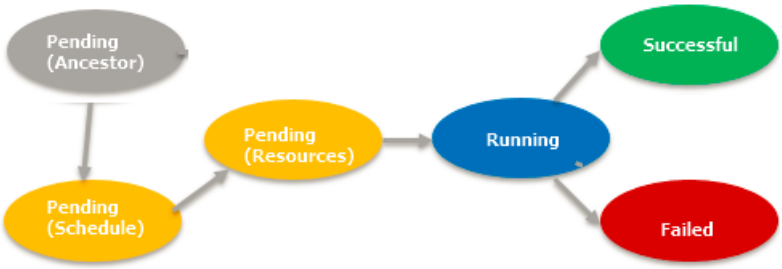
This topic describes the state machines of workflows and node instances.

Workflow state machine



- A workflow has four states: Pending (Ancestor), Running, Successful, and Failed.
- The initial state of a workflow is Pending (Ancestor). At this time, all instances in this workflow are in the Pending (Ancestor) state. When the workflow is called by the scheduling system, its state changes to Running and the root instance of the workflow runs.
- When an instance in the workflow fails, the state of the workflow changes to Failed.
- When all instances in the workflow are in the Successful state, the state of the workflow changes to Successful.

Node instance state machine



- A node instance has six states: Pending (Ancestor), Pending (Schedule), Pending (Resources), Running, Successful, and Failed.
- The initial state of a node instance is Pending (Ancestor). When it is called by the scheduling system, the system checks whether all its parent nodes are in the Successful state. If yes, the state of the instance changes to Pending (Schedule).
- The node instance is called at the time that is specified for running the node. The instance is then sent to the execution engine and its state changes to Pending (Resources).
- The execution engine allocates resources to the instance. The instance runs, and the scheduling system changes the state of the instance to Running. The execution engine sends the result to the scheduling system, and then the scheduling system changes the instance state to Successful or Failed.

27.4.10.5. Node dependencies

You can configure dependencies for nodes based on your business requirements.

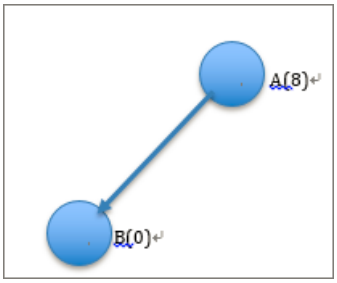
Same-cycle dependency

This is the most common scenario where an instance depends only on its parent instances in the same day. You can configure the following dependencies: A daily-run instance depends on another daily-run instance, a daily-run instance depends on an hourly-run instance, an hourly-run instance depends on a daily-run instance, or an hourly-run instance depends on another hourly-run instance.

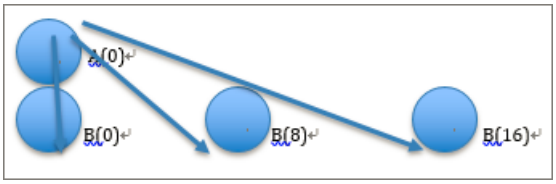
If an hourly-run instance depends on another hourly-run instance, three situations can occur: The number of parent instances is equal to the number of child instances, the number of parent instances is greater than the number of child instances, or the number of parent instances is less than number of child instances. The following examples show all the situations.

> **Note** In the following examples, all A nodes are parent nodes, and all B nodes are child nodes.

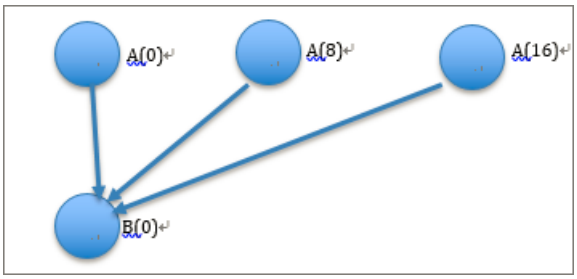
- A daily-run instance depends on a daily-run instance. The B node is specified to run at 08:00. The A node is specified to run at 00:00.



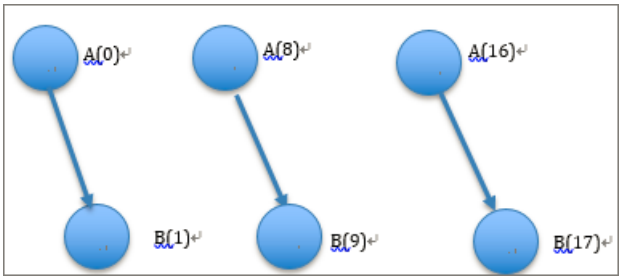
- An hourly-run instance depends on a daily-run instance. The B node is specified to run at 00:00, 08:00, and 16:00. The A node is specified to run at 00:00.



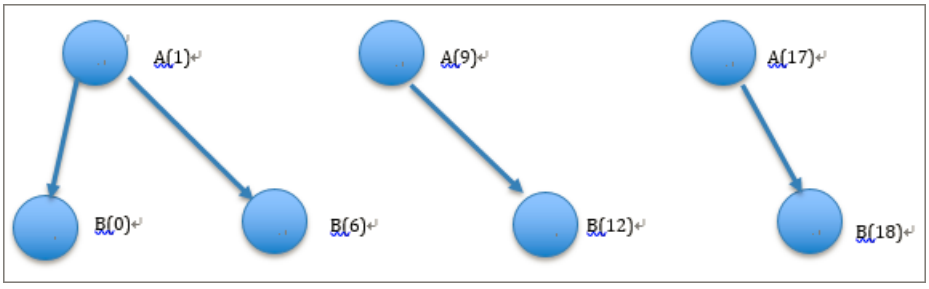
- A daily-run instance depends on an hourly-run instance. The B node is specified to run at 00:00. The A node is specified to run at 00:00, 08:00, and 16:00.



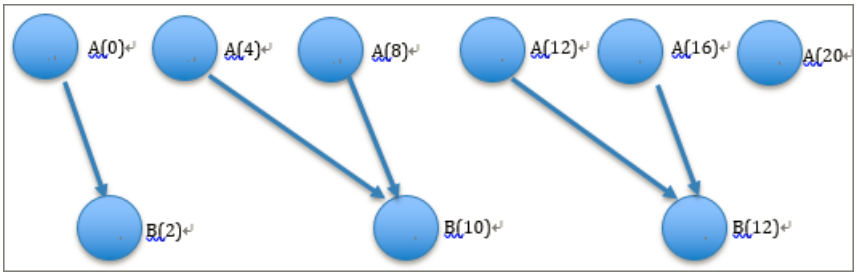
- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is equal to the number of child instances. The B node is specified to run at 01:00, 09:00, and 17:00. The A node is specified to run at 00:00, 08:00, and 16:00.



- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is less than the number of child instances. The B node is specified to run at 00:00, 06:00, 12:00, and 18:00. The A node is specified to run at 01:00, 09:00, and 17:00.



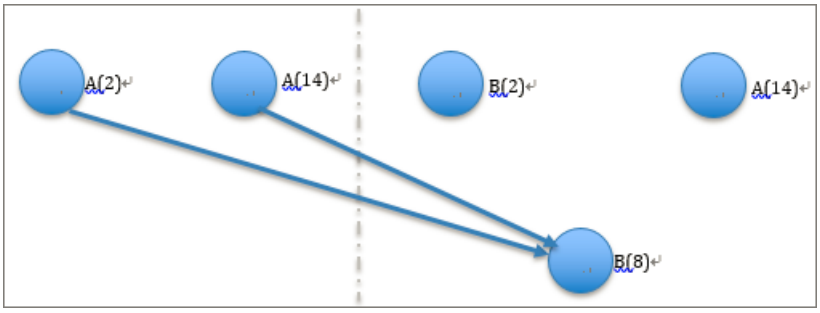
- An hourly-run instance depends on an hourly-run instance, and the number of parent instances is greater than the number of child instances. The B node is specified to run at 02:00, 10:00, and 18:00. The A node is specified to run at 00:00, 04:00, 08:00, 12:00, 16:00 and 20:00.



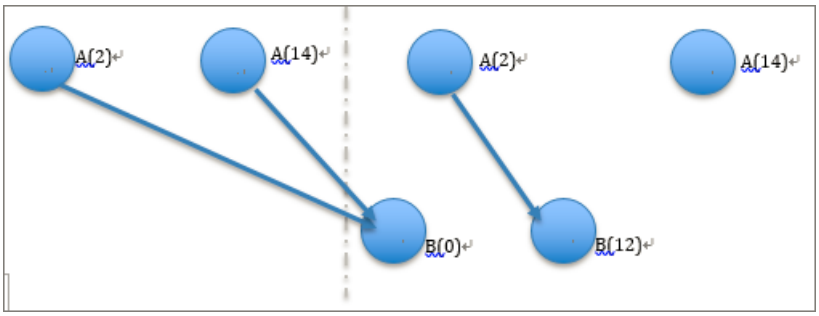
Cross-cycle dependency

You can configure cross-cycle dependency if the data processing operation requires the result of the data processing operation on the previous day.

- In most cases, you only need to configure the dependency between the current instance and the instance in the last day. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 08:00.

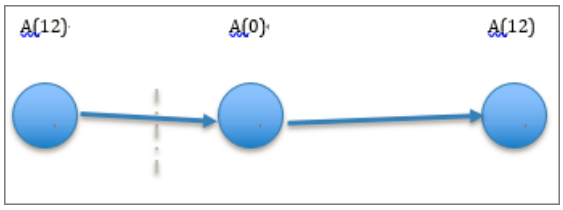


- The same-cycle dependency and the cross-cycle dependency can both exist. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 00:00 and 12:00.



Self-dependency

If a node instance depends on the instance that is generated from the same node in the last cycle, you must configure self-dependency. The following figure shows the dependencies in the situation where the A node is specified to run at 00:00 and 12:00.



28.Realtime Compute

28.1. What is Realtime Compute?

28.1.1. Background

Realtime Compute has its beginnings in the real-time big screen service of Alibaba Group during the Double 11 Shopping Festival. The big screen service allows you to view sales data during the shopping festival in real time on big screens. With five years of experience and development, the small team that once provided the real-time big screen service and limited real-time reporting services has become an independent and reliable cloud computing team. Realtime Compute provides an end-to-end cloud solution for stream processing based on years of experience in real-time computing products, architecture, and business scenarios. We strive to help more enterprises with real-time big data processing.

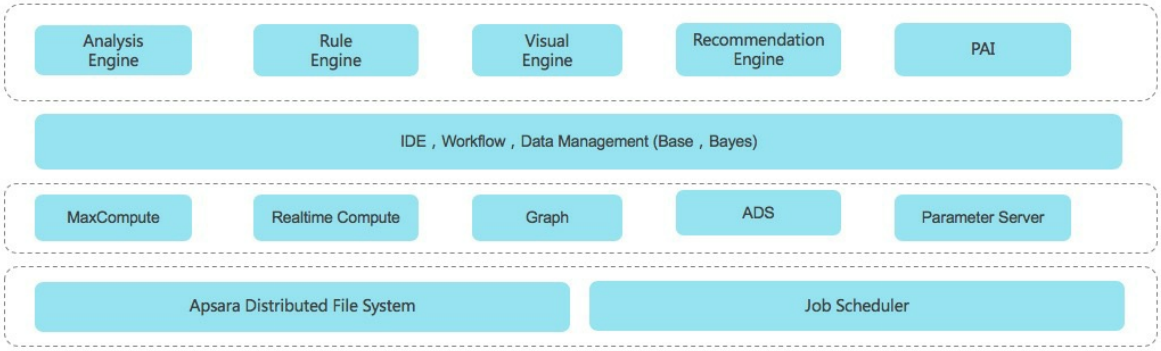
We previously used the open source Storm system to support the big screen service of Alibaba Group during the Double 11 Shopping Festival. We also developed stream processing code based on Storm. In these early stages, the stream processing service was provided on a small scale. Developers used Storm APIs to create jobs for stream processing. In this scenario, developers must have proficient technical skills, handle debugging challenges, and perform large amounts of repetitive work.

To address these challenges, we started working on data encapsulation and abstraction. Before data encapsulation and abstraction, we needed to choose an integrated processing engine for stream and batch processing from the available options: Apache Spark and Flink. The key difference of Apache Spark and Flink lies in the way they process data streams and batches. In Apache Spark, data streams are divided into micro batches, which are then processed by the Spark engine to generate the final stream of results in batches. For this method, the overhead must be increased to achieve a lower delay. Therefore, it is hard to reduce the delay of Spark Streaming to seconds or to sub-second level. In Apache Flink, batches are considered as bounded data streams that have a defined start and end. In this way, most code can be shared for stream and batch processing, which allows you to leverage the advantages of batch processing. Based on a thorough comparison between Apache Spark and Flink, we decided to use Apache Flink as the processing engine for real-time computations over data streams. Stream processing methods can be classified as stateful computations and stateless computations. The introduction of state management allows you to easily implement complex processing logic, which is ground-breaking for stream processing.

Any emerging technology is only adopted by a small group in the beginning. With the growth of this technology and the reduction in adoption costs, it will be widely accepted. Therefore, we are working to enable stream processing technologies to be widely adopted by improving the technology and decreasing adoption costs. Apache Flink has made many improvements to the architecture, but its implementation mechanism needs to be optimized. For example, the tasks of multiple jobs may be executed by the same thread, which greatly reduces the computing performance. To resolve this issue, we introduce the YARN system. Another example is the checkpoint feature of Apache Flink. In Apache Flink, checkpoints are created to ensure data consistency, but checkpoints cannot be created when the state stored for incremental computing is excessively large. To address this challenge, Realtime Compute optimizes the checkpoint feature to efficiently manage large state. Realtime Compute has addressed many performance issues and bottlenecks to ensure the stability and scalability in the production environment. Currently, Realtime Compute is capable of supporting core businesses. We have also improved the SQL of Realtime Compute to support complex business scenarios. We are working to provide excellent user experience through constant exploration and innovation.

28.1.2. Key challenges of Realtime Compute

Realtime Compute runs on a cluster of thousands of nodes within Alibaba Group. It provides services for hundreds of real-time applications for over 20 business units of Alibaba Group, processing hundreds of billions of messages and about 1 petabyte of traffic per day. Realtime Compute has become one of the core distributed computing services of Alibaba Group.



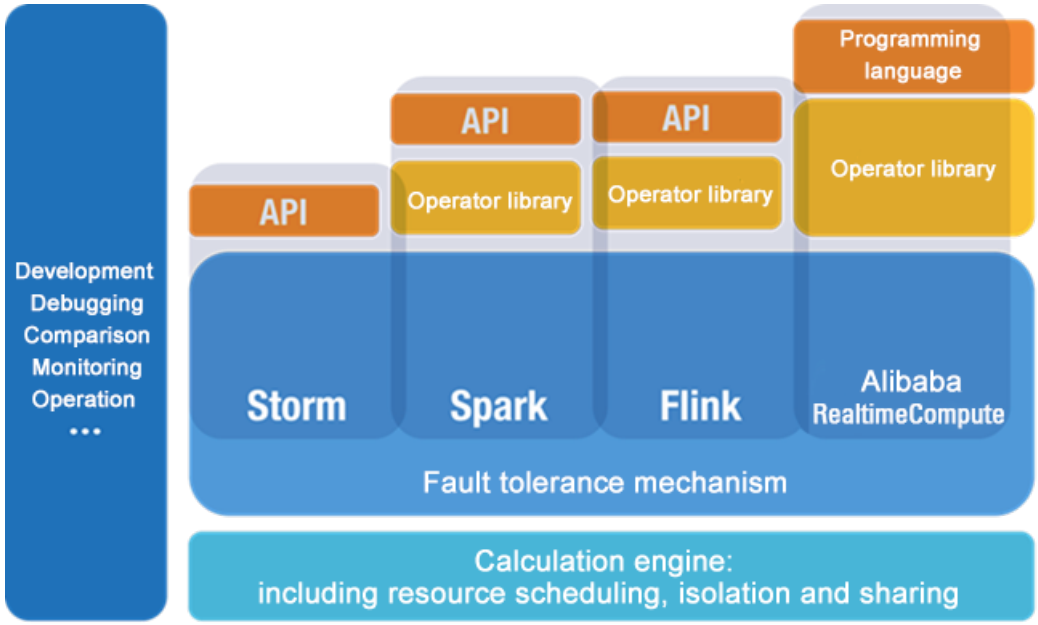
We are working to make the following improvements:

- Computing engine: We are working to improve the engine performance and enable the engine to support multiple semantics of processing messages.
- Programming interfaces: We are working to enable support for more APIs and programming languages. For example, we are working on the compatibility with open source APIs, such as Storm APIs and Beam APIs.
- Programming languages: We are working to enable support for more SQL syntaxes and semantics in stream analysis scenarios, such as temporal tables and complex event processing (CEP). Services: We are working to improve Realtime Compute from the following aspects: debugging, one-click deployment, hot upgrades, and training systems.

28.2. Technical advantages

Realtime Compute uses a compute engine that is developed based on Apache Flink, which allows Realtime Compute to leverage advantages of Apache Flink and optimize the Flink Table API. You can use Flink SQL for batch and stream processing. The application of YARN in Realtime Compute enables full compatibility with Flink API, which enables a large ecosystem of stream processing.

Realtime Compute and other stream processing system



Realtime Compute and other stream processing system shows the differences between the technologies of Realtime Compute and other stream processing systems. Based on the extensive experience of addressing challenging business scenarios, Realtime Compute provides the following benefits:

- Powerful stream processing functions

  Unlike these open source systems, Realtime Compute simplifies the development process by integrating a wide range of functions. These functions are described as follows:

  - A powerful engine is used. This engine offers the following advantages:
    - Provides the standard Flink SQL that enables automatic data recovery from failures. This ensures accurate data processing when failures occur.
    - Supports multiple types of built-in functions, such as text functions, date and time functions, and statistics functions.
    - Enables an accurate control over computing resources. This ensures complete isolation of each tenant's jobs.
  - The key performance metrics of Realtime Compute are three to four times higher than those of Apache Flink. For example, in Realtime Compute, the data processing delay is reduced to seconds or even to sub-second level. The throughput of a job reaches millions of data records per second. A cluster can contain thousands of nodes.
  - Realtime Compute integrates cloud-based data stores such as MaxCompute, DataHub, Log Service, ApsaraDB for RDS, and Table Store. With Realtime Compute, you can read data from and write data to these systems with the least efforts in data integration.

- Managed real-time computing services

  Unlike open source or user-developed stream processing services, Realtime Compute is a fully managed stream processing engine. You can query streaming data without deploying or managing any infrastructure. With Realtime Compute, you can use streaming data processing services with a few clicks. Realtime Compute integrates services such as development, administration, monitoring, and alerting. This allows you to use cost-effective streaming data services for trial and migrate your data for deployment.

Realtime Compute also enables complete isolation between tenants. This isolation and protection extends from the top application layer to the underlying infrastructure layer. This helps to ensure the security and privacy of your data.

- Excellent user experience during development

  Realtime Compute provides a standard SQL engine: Flink SQL. It also provides many built-in functions, such as the text functions, date and time functions, and statistics functions. The application of these functions greatly simplifies and accelerates the Flink-based development. With Flink SQL, even users with limited development knowledge, such as business intelligence (BI) analysts and marketers, can easily perform real-time analysis and processing of big data.

  Realtime Compute provides an end-to-end solution for stream processing, including development, administration, monitoring, and alerting. On the Realtime Compute development platform, only three steps are required to publish a job.

- Low costs

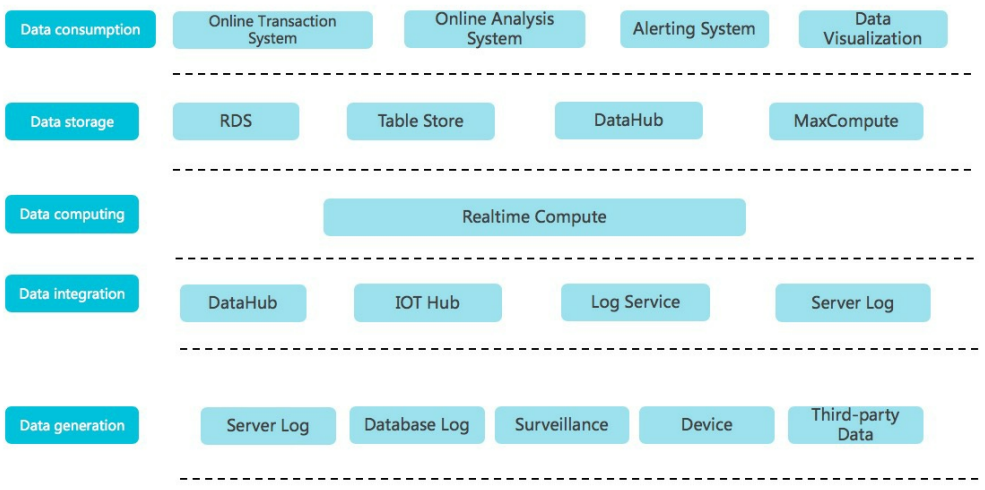
  We have made many improvements to the SQL execution engine, allowing you to create jobs more cost-effectively than to create Flink jobs. Realtime Compute is more cost-effective than open source stream frameworks in both development and production costs.

28.3. Product architecture

28.3.1. Business architecture

Realtime Compute is a lightweight SQL-enabled streaming engine for real-time processing and analysis of data streams.

Business architecture



- Data generation

  In this phase, streaming data is generated from sources such as server logs, database logs, sensors, and third-party systems. The generated streaming data moves on to the next phase for data integration to drive real-time computing.

- Data integration

In this phase, the streaming data is integrated. You can subscribe to and publish the integrated streaming data. The following Alibaba Cloud products can be used in this phase: DataHub for big data computing, IoT Hub for connecting IoT devices, and Log Service for integrating ECS logs.

- Data computing

  In this phase, the streaming data, which has been subscribed to in the data integration phase, acts as inputs to drive real-time computing in Realtime Compute.

- Data storage

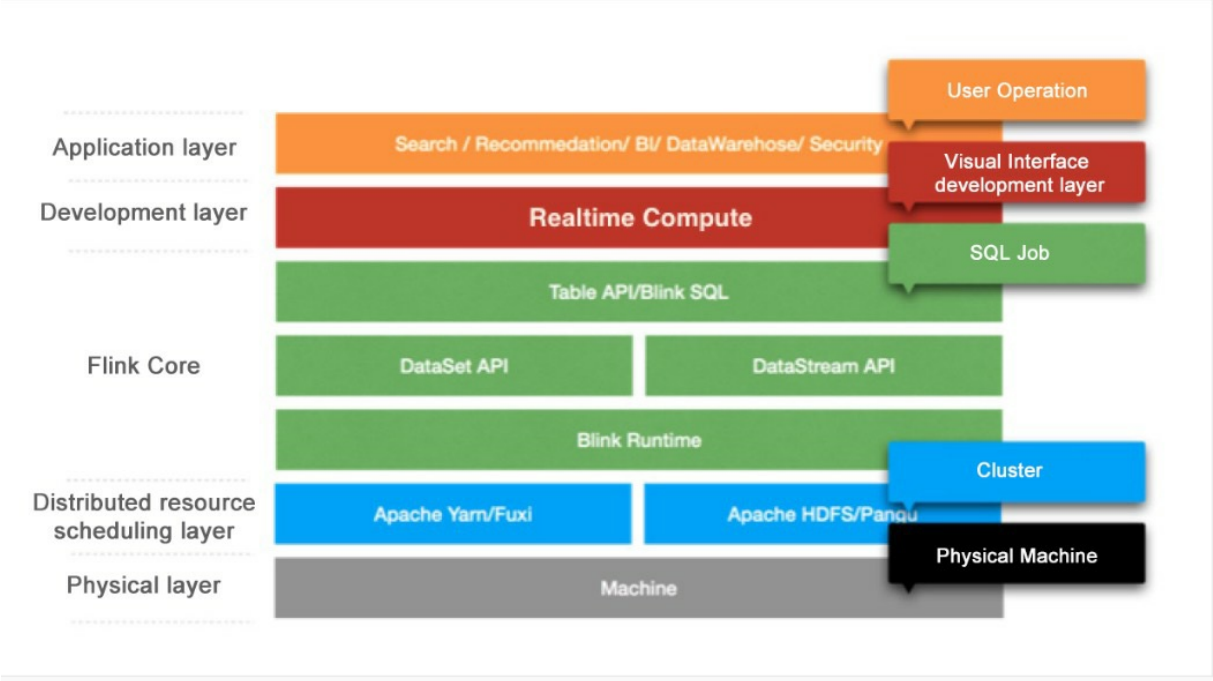
  Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- Data consumption

  Realtime Compute supports multiple data store types, which allows you to consume data in various ways. For example, data stores for message queues can be used to report alerts, and relational databases can be used to provide online support.

28.3.2. Technical architecture

Realtime Compute is a real-time data analysis platform for incremental computing. This platform provides statements that are similar to SQL statements and uses the MapReduceMerge (MRM) computing model for incremental computing. Realtime Compute offers a failover mechanism to ensure data accuracy when errors occur.



The Realtime Compute architecture consists of the following five layers.

- Application layer

  This layer allows you to create SQL files and publish jobs for real-time data processing based on a development platform. With a well-designed monitoring and alerting system, you would be notified of a processing delay for each job in a timely manner. You can also use systems like Flink UI to view the running information of published jobs and analyze performance bottlenecks. This allows you to quickly and effectively improve job performance.

- Development layer

  This layer parses Flink SQL and generates logical and physical execution plans. The execution plans are then conceptualized as executable directed acyclic graphs (DAGs). Based on these DAGs, directed graphs that consist of various models are obtained. Directed graphs are used to implement specific business logic. A model usually contains the following three modules:

  - Map: Operations such as data filtering, distribution (GROUP), and join (MAPJOIN) are performed.
  - Reduce: Realtime Compute processes streaming data by batch, and each batch contains multiple data records.
  - Merge: You can update the state by merging the computing results of the batch, which are produced from the Reduce module, with the previous state. Checkpoints are created after N (configurable) batches have been processed. In this way, the state is stored persistently in a data store, such as Tair and Apache HBase.

- Flink Core

  This layer provides a wide range of computing models, Table API, and Flink SQL. You can use DataStream API and DataSet API at the lower sublayer. At the bottom sublayer is Flink Runtime, which schedules resources to ensure that jobs can run properly.

- Distributed resource scheduling layer

  Realtime Compute clusters run based on the Gallardo scheduling system. This system ensures that Realtime Compute runs effectively and fault tolerance is provided for recovery.

- Physical layer

  This layer provides powerful hardware devices for clusters.

28.4. Functional principles

The Blink engine of Realtime Compute is developed based on Apache Flink. For more information about the functional principles of Realtime Compute, see Discussion on Apache Flink.

29.Machine Learning Platform for AI

29.1. What is machine learning?

Machine learning is a process of using statistical algorithms to learn large amounts of historical data. It allows you to generate an empirical model and make informed business decisions.

Apsara Stack Machine Learning Platform for AI is a set of data mining, modeling, and prediction tools. It is developed based on MaxCompute (also known as ODPS). Machine Learning Platform for AI supports the following features:

- An all-in-one algorithm service that supports algorithm development, sharing, model training, deployment, and monitoring.
- You can manage experiments through the graphic user interface (GUI) or by running commands. Machine Learning Platform for AI is intended for data miners, analysts, algorithm developers, and data explorers.
- In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. After you deploy algorithms in MaxCompute clusters, you can call the algorithms from the Alibaba Cloud Machine Learning Platform for AI console. This decouples algorithm applications from compute engines.
- Multiple algorithms and reliable technical support are provided to resolve issues in various scenarios. In the Data Technology (DT) era, you can develop data-driven business by using Machine Learning Platform for AI.

Machine learning is used in the following scenarios:

- Marketing: commodity recommendation, user profiling, and targeted advertising.
- Finance: credit risk prediction for loans, financial risk management, stock forecast, and gold price forecast.
- Social network: analytics of key opinion leaders and relational networks.
- Text processing: news classification, keyword extraction, document summarization, and text analysis.
- Unstructured data processing: image classification and image text extraction based on Optical Character Recognition (OCR).
- Other scenarios: rainfall forecast and football match result forecast.

Machine learning is classified into the following types:

- Supervised learning: Each sample has an expected value. You can create a model to map input feature vectors to target values. Typical examples of this learning mode include regression and classification.
- Unsupervised learning: Samples do not have target values. This learning mode is used to discover potential regular patterns from data. Simple clustering is a typical example of this learning mode.
- Reinforcement learning: This learning mode is complex. A system constantly interacts with the external environment to obtain external feedback and determines its own behavior to achieve a long-term optimization of targets. Typical examples of this learning mode include AlphaGo and autonomous driving.

29.2. Benefits

Distributed algorithm framework

- Machine Learning Platform for AI mainly supports three engines: deep learning, parameter server, and MPI.
- Deep learning engine with excellent performance.

Improved model and compilation efficiency

Collaborative optimization of models and system compilation is a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports collaborative optimization of models and system compilation.

Heterogeneous resource scheduling

For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

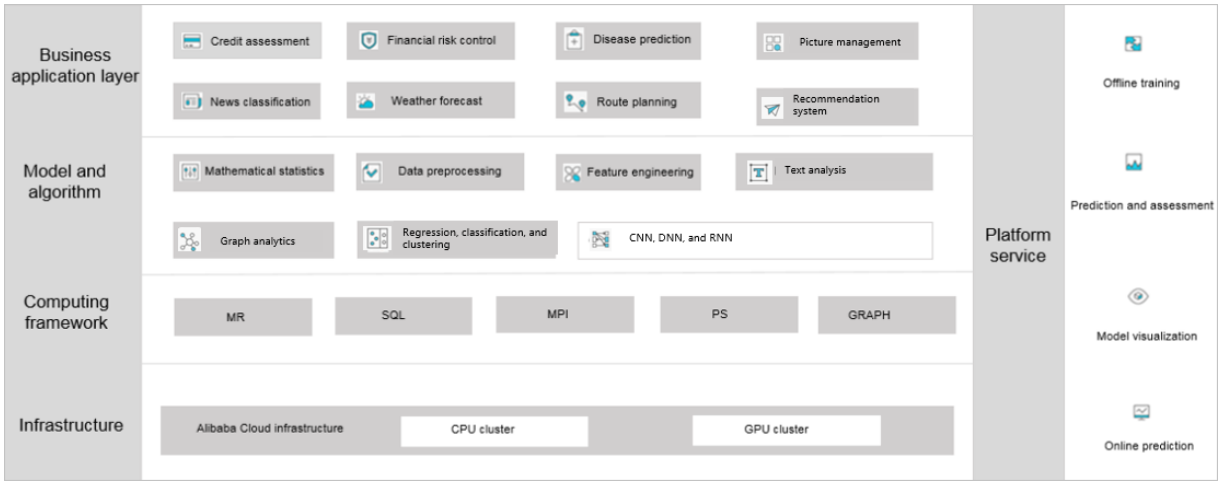
Quality algorithms

All algorithms come from the Alibaba Group algorithm system and have been tested on petabytes of service data and complex business scenarios. This ensures their sophistication and stability.

29.3. Architecture

29.3.1. System architecture

Machine Learning Platform for AI consists of multiple component systems, as shown in the following figure.



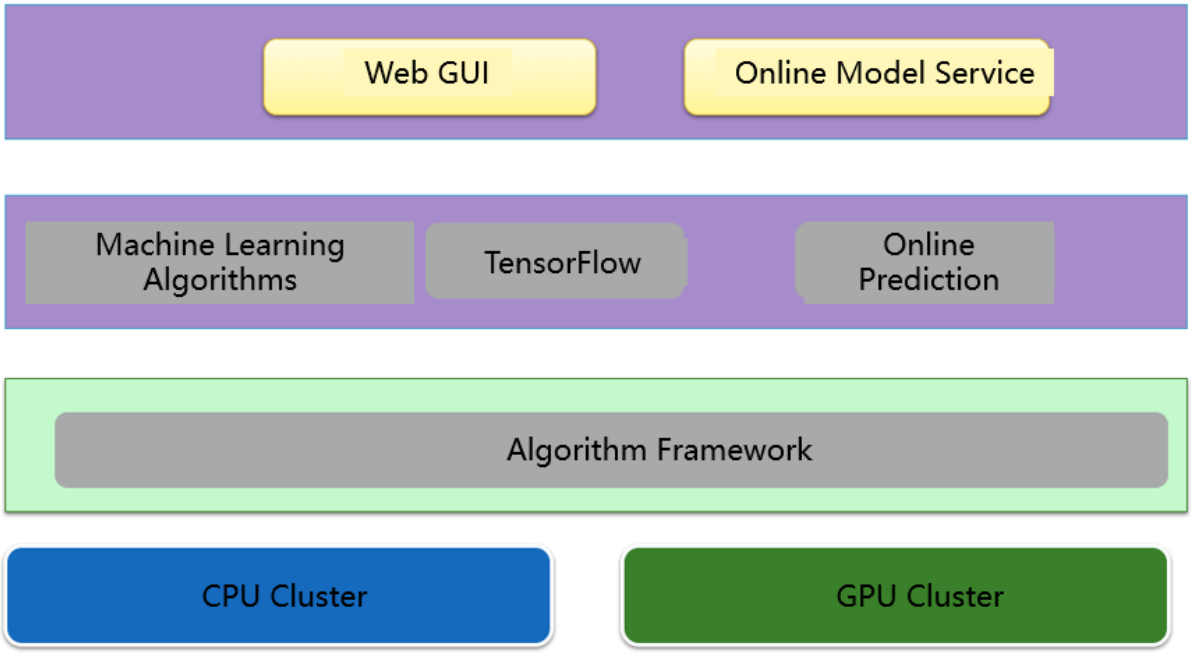
Architecture of Machine Learning Platform for AI:

- Infrastructure layer: includes the CPU and GPU clusters.
- Computing framework layer: provides calculation methods such as MapReduce, SQL, and MPI. The distributed computing architecture is used to perform concurrent execution and distribution of computing tasks.
- Model and algorithm layer: includes basic components, such as data preprocessing, feature engineering, and machine learning algorithms. All of the algorithm components come from the Alibaba Group algorithm system and have been tested on petabytes of service data.
- Service application layer: supports the search system, recommendation system, Ant Financial, and other Alibaba projects in data mining. Machine Learning Platform for AI is applicable in various industries, such as finance, medical care, education, transportation, and security.

If you call models and algorithms in Machine Learning Platform for AI, the system converts the algorithms into compute types. For example, to join two tables, an SQL workflow is automatically generated and then delivered to MaxCompute for calculation and processing. All algorithms are stored in the underlying compute engine as plug-ins for convenient use. This decouples the algorithms from the compute engine.

29.3.2. Functional architecture

Machine Learning Platform for AI consists of the following components:



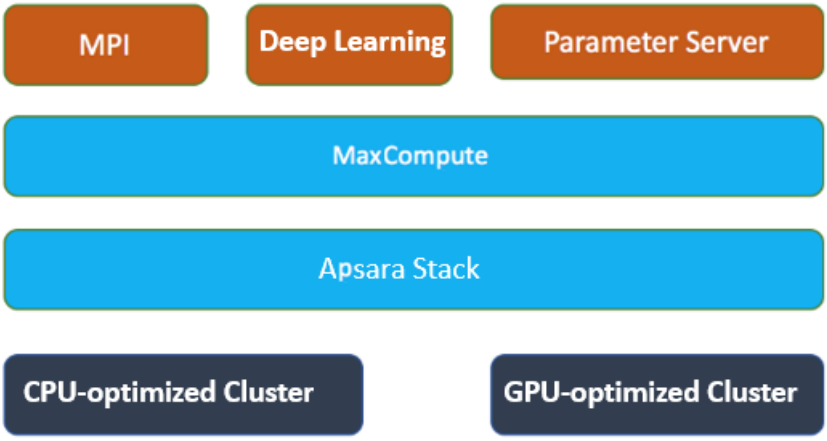
Component	Description
CPU cluster	The CPU cluster runs machine learning algorithms and provides computing resources such as CPU and memory resources. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in a CPU cluster and dispatches jobs to the compute nodes.
GPU cluster	The GPU cluster runs deep learning framework jobs and provides computing resources such as GPU and graphics memory resources. • Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in the CPU cluster and dispatches jobs to the compute nodes. • For a task that requires multiple workers and GPUs, a virtual network is automatically created to dispatch the jobs to the compute nodes in the virtual network.
Algorithm framework	• The algorithm framework manages CPU and GPU computing resources. • The algorithm framework also provides a basic environment for running algorithms, supporting the Message Passing Interface (MPI) library, MapReduce library, Parameter Server (PS) library, algorithm package, and job isolation by user.

Component	Description
Machine learning algorithms	Machine learning algorithms such as data processing, classification, regression, clustering, text analysis, and network analysis are developed in the basic runtime environment of the algorithm framework. The algorithms are provided as components that can be used in experiments.
TensorFlow	• Based on the algorithm framework, the deep learning framework TensorFlow is provided. In addition, the performance and throughput of the TensorFlow open-source edition have been improved. The TensorFlow open-source 1.4 edition is supported. • You can use TensorFlow to read files from and write models to OSS buckets. • When TensorFlow is running, you can start TensorBoard to view the status of parameter convergence during convolution.
Online model service	You can deploy machine learning models and TensorFlow-generated models as online mode services. The online model service supports model version management and blue-green deployment in the rolling upgrade mode.
Web GUI	A visual experiment management console provided by Machine Learning Platform for AI. You can perform the following actions on the Web GUI: • Create experiments, add algorithm components, and run experiments. • You can also deploy models as online model services or publish experiments to the scheduling system in DataWorks.
Call online model services	Models that are deployed as online model services provide APIs for users to call these services through the Internet.

29.4. Functions

29.4.1. Resource allocation and task scheduling

Artificial intelligent (AI) tasks typically consume considerable computing resources. Therefore, a distributed system is indispensable. A task must not occupy all resources or occupy a resource exclusively. Instead, a resource is shared by multiple tenants. Machine Learning Platform for AI balances the efficiency of resource usage between a single task and a cluster.

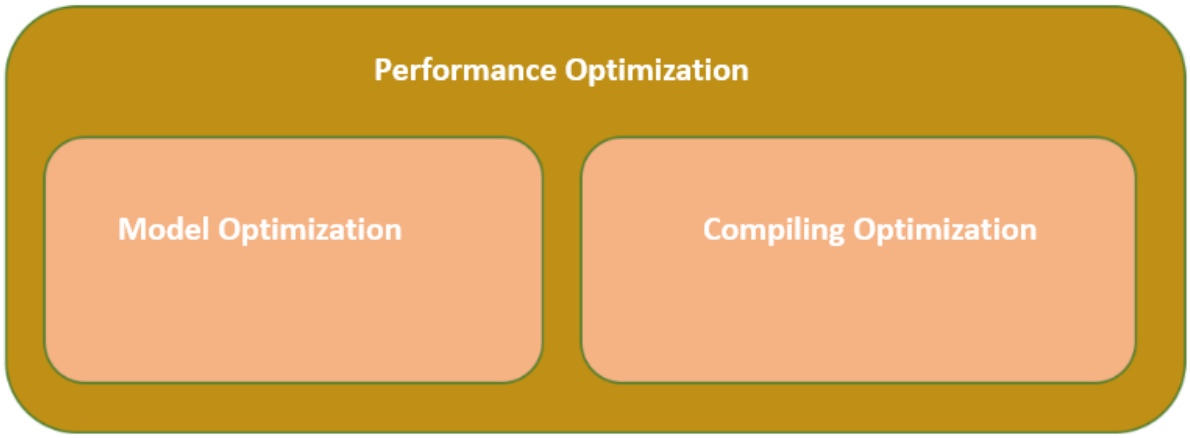


Machine Learning Platform for AI is built on the Apsara operating system and MaxCompute clusters, and is equipped with three types of compute engines: deep learning, parameter server, and MPI. AI tasks and MaxCompute tasks are deployed together to maximize the utilization of resources. For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

To allocate resources to a single task, Machine Learning Platform for AI uses the Tensorflow framework to automatically build a computing chart, allocate CPU and GPU resources, and optimize the task execution efficiency.

29.4.2. Model and compilation optimization

Collaborative optimizations of models and system compilation are a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports the following types of optimization.



Model optimization

Many industrial service models are built based on the statistical learning theory. Model parameters can still be regularized and pruned. Besides, the AI-oriented heterogeneous computing tends to implement mixed precision to maximize the computing efficiency while guaranteeing service precision. As the hardware system develops, many technologies have been integrated in Machine Learning Platform for AI. These technologies include low bit quantization, tensor decomposition, network pruning, distillation compression, gradient compression, and hyperparameter optimization.

Compilation optimization

Model optimization aims to minimize the computing requirements when all service requirements are met. System compilation optimization is used to adapt the specified model to the heterogeneous computing architecture and release the hardware computing resources using end-to-end optimization technologies. Compilation optimization resolves the following issues:

- Computing requirement descriptions for service models. Machine Learning Platform for AI allows you to use advanced abstract languages to describe service models. You need only to describe the computing requirements. The system will translate the descriptions and perform automatic optimization.
- Hardware system independent computing chart optimization. Based on the intermediate expression of computing charts, the system implements optimizations that are independent of the hardware system structure. These optimizations include distributed splitting, mixed precision optimization, redundant computing elimination, computing mixing optimization, constant folding, efficient operator rewriting, and storage optimization of computing charts.
- Optimization and code generation related to the hardware system. The system performs optimization that is related to the hardware system and generates the target code. The optimization includes storage hierarchy optimization, parallel granularity reconstruction, computing and fetch streaming, assembly instruction optimization, and automatic CodeGen space exploration and tuning.

29.4.3. Compute engine

The compute engine provides an advanced programming language for you to compile machine learning models as needed. The engine converts the code into executable tasks at the back end, dissembles or merges the tasks, and submits the tasks to the scheduling system. Machine Learning Platform for AI supports three engines: deep learning, parameter server, and MPI.

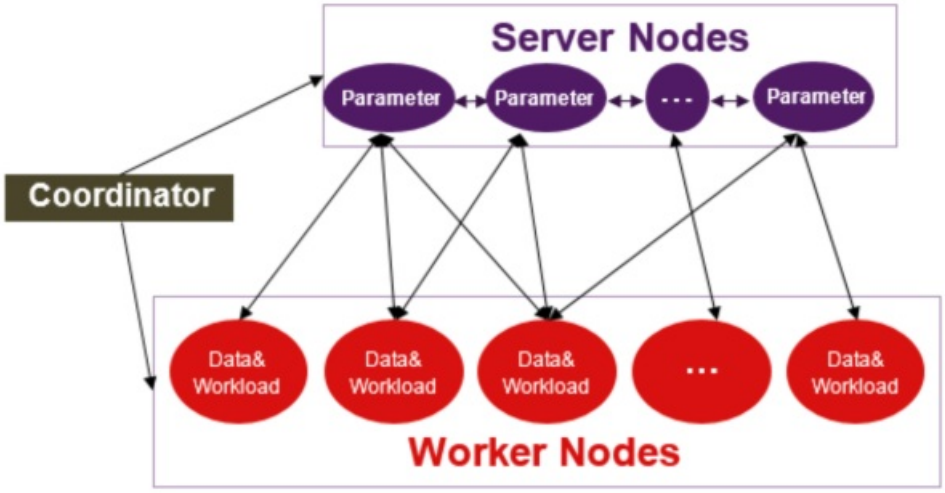
Deep learning

The deep learning engine is developed based on the open-source community TensorFlow. To adapt to the Apsara Stack cluster environment, the following improvements have been made to the deep learning engine:

- Multiple basic functions are supported. These functions include image management, service resuming, permission management, and reading and writing MaxCompute and OSS data.
- The runtime performance of the open-source TensorFlow has been improved.
  - Introduces the allreduce network primitive to improve network utilization.
  - Replaces the native gRPC mode with the RPC framework for better performance.
  - Modifies the synchronization mutex mode to reduce mutex lock competition.
- New optimizers and operators are available.

Parameter server

Parameter servers are a type of compute engine provided by Machine Learning Platform for AI for modeling training based on large models and large amounts of sample data. The engine allows algorithm developers to write distributed machine learning algorithm code in the same manner they write standalone code. Algorithm developers can implement distributed machine learning algorithms on the parameter server framework, and verify the algorithms based on tens of billions of parameter and data dimensions. This shortens the development cycle and allows new algorithms to be released for big data processing.



A parameter server supports the following functions:

- Creates hash indexes for features in real time.
- Allows you to add or delete features.
- Distributed expansion.
- Globally unified checkpoint and exactly once failover.
- Sparse hash feature-based communication.
- Embedding matrix computing based on sparse hash features.

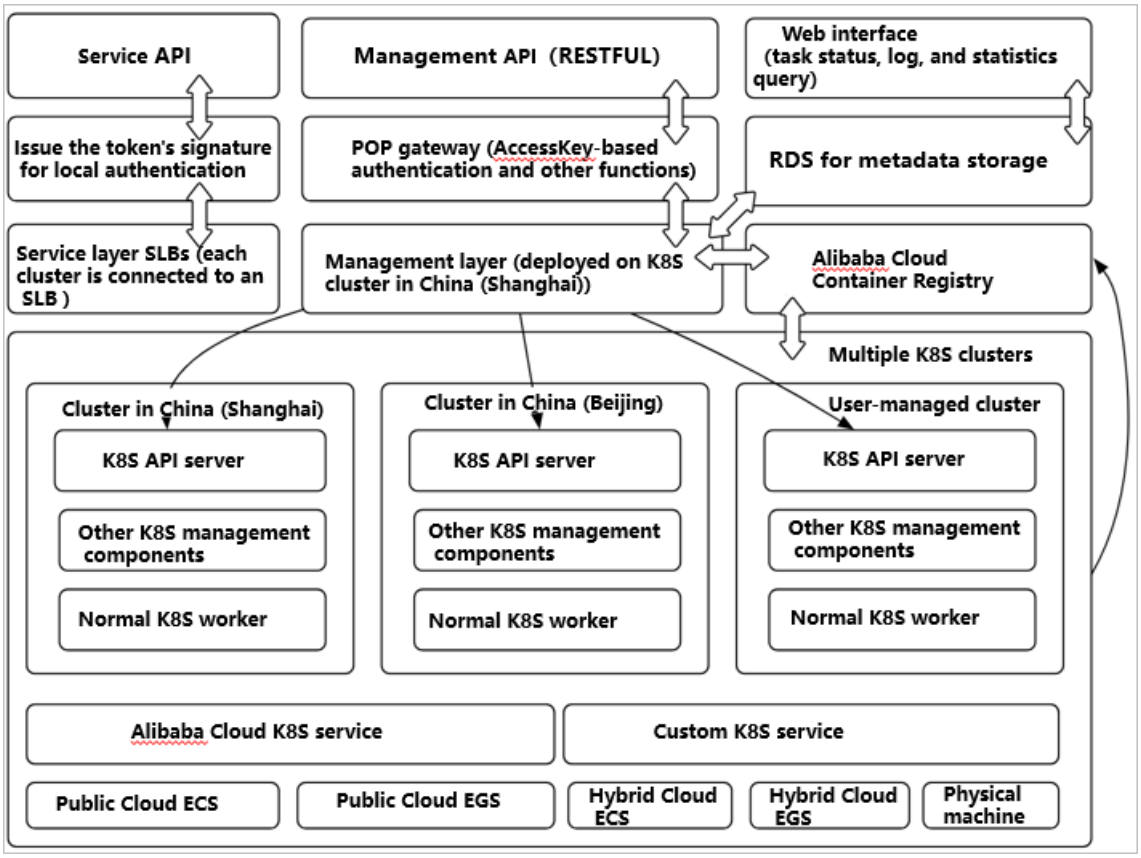
MPI

The MPI engine is a generic distributed framework used in the industry. Machine Learning Platform for AI introduces the MPI engine and integrates the MapReduce feature of MaxCompute so that you can implement classic machine learning algorithms such as logistic regression, GBDT, FM, and K-means.

29.4.4. Online prediction system

The online prediction system performs predictions tasks in the cloud using multiple types of CPUs and GPUs. The online prediction system is built based on Apsara Stack services, such as ECS, EGS, SLB, and RDS. It uses Docker to manage resources and isolate resources. It uses open-source Kubernetes (K8s) to schedule tasks.

The overall architecture of the online prediction system is as follows:



The preceding figure shows the architecture of the online prediction system.

API layer

The online prediction service APIs are classified into two types:

- Prediction service APIs
- Prediction request APIs

The two API types are designed with different features to meet different requirements.

- Prediction service APIs are used to create, deploy, delete, and modify prediction services.
- Prediction request APIs are used to process prediction requests sent by clients and return prediction results.

Computing layer

- Computing resources

  All computing resources are managed by Kubernetes (K8s). Each node in a K8s cluster is an ECS instance, EGS instance, or physical server.

- Failover

  Failover depends on the failover solution provided by K8s. K8s allows you to configure a listening service for each container. For example, when the listening port is set to port 80:

- If an IRP error occurs:
  a. Port 80 has failed the health check. K8s sets the status of the pod (container group) to Unavailable. Traffic is forwarded to another pod.
  b. When a pod is restarted, the framework initializes the pod and loads the model. Port 80 is not enabled until the model has been loaded.
  c. Port 80 is enabled after the initialization is completed. After port 80 passes the health check at the scheduling layer, the pod is added to the traffic pool again.
- If a node in a K8s cluster fails: The keepalive message exchange between the K8s primary node and the failed node fails. K8s sets the status of the failed node to Not Ready. The pod (container group) running on the node is migrated to another node. The traffic is also forwarded to another node.

- Rolling update

  Rolling updates indicate application updates with zero downtime. The updates are classified into two types:

  - User data update: User data about the model and processor is updated using an API provided by the online prediction service. The back end creates a new image version based on the current image version and updates the deployment.
  - IRP framework code update: The framework code is updated by creating a procedure to update all user tasks in the back end. Framework code update follows the rolling update procedure. Users are not aware of the update procedure.

  User data and framework code are decoupled during cluster scheduling and they can be updated separately. The system packages user data and framework code into images of later versions separately and then modifies the description file of the existing application deployment. K8s performs rolling updates for running pods and dynamically switches the traffic to ensure that users are unaware of ongoing updates.

29.4.5. List of functions by module

Machine Learning Platform for AI provides a complete workflow of machine learning, such as data uploading, data processing, data visualization, model training, model deployment, model evaluation, and model utilization.

The following table describes the modules and corresponding functions.

Module	Function	Description
Data control	Data uploading	You can upload data through Machine Learning Platform for AI. When you upload data, the data is parsed, verified, and any errors are recorded and reported.
	Data table displaying	On Apsara Stack Machine Learning Platform for AI, click **Data Source** in the left-side navigation pane to view the uploaded data tables. You can enter a data table name in the search box and click the search icon to search for a data table. Fuzzy search is also supported.

Module	Function	Description
	Data visualization	Right-click a component and choose **View Data** from the shortcut menu to view data in histograms, pie charts, or line charts.
Model control	Model training	On Machine Learning Platform for AI, click **Run** in the upper section of the canvas to train and generate a model.
	Model visualization	On Machine Learning Platform for AI, click **Models** in the left-side navigation pane. Right-click a model and choose **Show Model** from the shortcut menu to view model parameters. Tree models and linear models can be displayed in tables.
	Model downloading	Right-click a model and choose **Export PMML** from the shortcut menu to generate and download a PMML file. A PMML is a standard model description file which can be parsed by a variety of open-source software.
	Model-based prediction	You can connect model generation components and prediction components. The system will automatically use the generated model for prediction.
	Model addition, deletion, modification, and query	Right-click a model and choose to add, delete, modify, or query a model.
	Online model service	You can use the online model service to deploy a model and call the corresponding RESTful API for online prediction.
	DataWorks task scheduling	You can deploy experiments to DataStudio as DataWorks tasks and configure the system to periodically run the tasks.
	Model evaluation	You can evaluate models using confusion matrix, binary classification evaluation, clustering model evaluation, and regression model evaluation. Models are evaluated based on metrics such as F1 score, AUC, and KS. All evaluation results can be viewed in tables or charts.
Experiment control	Whole experiment lifecycle control	You can add, delete, modify, query, and copy experiments.
	Experiment visualization	Animated visualizations are used to display the entire procedure by which an experiment runs.
	Notifications	The status of a running experiment is displayed in a prompt in the upper-right corner of the canvas, such as success and error messages.

Module	Function	Description
Deep learning	Multiple deep learning frameworks	Three mainstream deep learning frameworks are supported: TensorFlow, Caffe, and MXNet. With many underlying optimizations, TensorFlow delivers better performance than other open-source frameworks.
	TensorBoard	You can view the training status of each layer in a TensorBoard job in real time and display the results visually.
	Automatic authorization	When the data source of a TensorFlow project is set to OSS, you must obtain permissions on OSS before you can run an experiment. Machine Learning Platform for AI supports automatic authorization, allowing you to obtain the read and write permissions on OSS with a single click.
	Visualized TensorFlow execution settings	The TensorFlow component is added to provide related data source settings, allowing you to run the component visually. On the Tuning tab, you can specify the number of GPUs to run with and implement parallel training with multiple GPUs easily.
	Scheduling	Deep learning jobs can be deployed and periodically executed in DataWorks.
Dashboard	Experiment history chart	You can view the experiment history on the dashboard page.
	Running experiments	You can view running experiments or delete a running experiment to save resources.
	Scheduled tasks	You can view scheduled tasks that have been deployed and add, delete, modify, and query tasks through DataStudio.
Templates on the homepage	Machine Learning Platform for AI provides many built-in experiment templates	The experiment templates can be used for a wide range of scenarios such as product recommendations, news classification, financial risk control, haze prediction, heart disease prediction, agricultural loan delivery, and census. All these cases contain complete data sets and instructions about their use. You can also create your own experiments by using these templates.
	Model version management	You can upload multiple versions of a model, configure them to share the same resources, and switch between those versions.

Module	Function	Description
Online prediction	Blue-green model deployment	The blue-green model deployment function allows you to dynamically change the proportions of the traffic forwarded between different versions of a model.
	Online model debugging	The online debugging function of Machine Learning Platform for AI allows you to debug deployed models online and view the debugging results in real time.

29.5. System metrics

Metric	Requirement
Core metrics	Provides typical machine learning algorithms, such as the data preprocessing, feature engineering, statistical analysis, classification, regression, and clustering: • Provides model evaluation algorithms. • Provides time series, text analysis, and network analysis algorithms. • Provides deep learning frameworks such as TensorFlow. • Provides the GPU job scheduling capability. • Provides the online model service and allows you to directly deploy models to the online model service. • Provides a visual console to help you use visual components to create experiments.
	Supports reading structured and unstructured data.
	• Supports data sampling and filtering algorithms, such as the random sampling, weighted sampling, and stratified sampling. • Supports data merging algorithms, such as JOIN, UNION, and MERGE. • Supports data preprocessing algorithms, such as splitting, normalization, standardization, KV to Table, Table to KV, and adding ID columns to tables.
	• Supports the principal component analysis (PCA) algorithm. • Supports feature importance evaluation for linear and random forest models.
	Supports the following statistical analysis algorithms: the covariance, empirical probability density chart, whole table statistics, chi-square goodness of fit test, chi-square test of independence, scatter plot, correlation coefficient matrix, two sample T test, single sample T test, normality test, percentile, Pearson coefficient, and histogram.

Metric	Requirement
Function metrics	• Supports the following binary classification algorithms: the Gradient Boosting Decision Tree (GBDT), Linear Support Vector Machine (SVM), and logistic regression. • Supports the following multiclass classification algorithms: K-nearest neighbors (KNN), multiclass classification for logistic regression, random forest, and naive Bayes. • Supports the GBDT, linear regression, PS-SMART regression, and PS linear regression algorithms. • Supports K-means clustering. • Supports the following evaluation algorithms: the binary classification model, regression model, clustering model, multiclass classification, and confusion matrix.
	• Supports the deep learning framework TensorFlow. • Supports TensorBoard. • Supports scheduling a deep-learning job to a GPU server.
	Supports time series algorithms such as x13_arima and x13_auto_arima.
	Supports the following text algorithms: the word frequency statistics, TF-IDF, parallel latent dirichlet allocation (PLDA), Word2Vec, word splitting, converting rows, columns, and values to KV pairs, string similarity, deprecated word filtering, text summarization, document similarity, sentence splitting, keyword extraction, ngram-count, semantic vector distance, and pointwise mutual information (PMI).
	Supports the following network analysis algorithms: the K-Core, single-source shortest path, page rank, label propagation clustering, label propagation classification, modularity, maximum connected subgraph, vertex clustering coefficient, edge clustering coefficient, counting triangle, and tree depth.
	• Supports the online model service and allows you to deploy machine learning algorithm models or deep-learning models to the service. • Provides an HTTP-based API.
	• Provides the Web-based visual editor, which allows you to create an experiment by dragging and dropping components. • Supports releasing experiments to DataWorks for task scheduling. • Supports experiment and model management.
Compatibility/openness	• Supports the open-source deep learning framework TensorFlow 1.4. • Supports exporting the PMML file from machine learning models. • Supports the online service model and allows you to deploy the model as an API.

30.E-MapReduce (EMR)

30.1. What is E-MapReduce?

E-MapReduce (EMR) is short for Elastic MapReduce. EMR is an end-to-end big data processing and analytics system. It leverages open source big data ecosystems such as Hadoop, Spark, Kafka, and Storm to manage clusters, jobs, and data.

30.2. Benefits

This topic describes the technical benefits of EMR in terms of deep integration and security.

EMR provides an integrated solution to manage clusters, which frees you up from the complex management of user-created clusters. EMR also offers the following benefits:

- Deep integration

  EMR is integrated with other Alibaba Cloud services such as Object Storage Service (OSS), Message Service (MNS), ApsaraDB for RDS, and MaxCompute. This enables these services to act as the input source or output destination of the Hadoop or Spark compute engine of EMR.

- Security

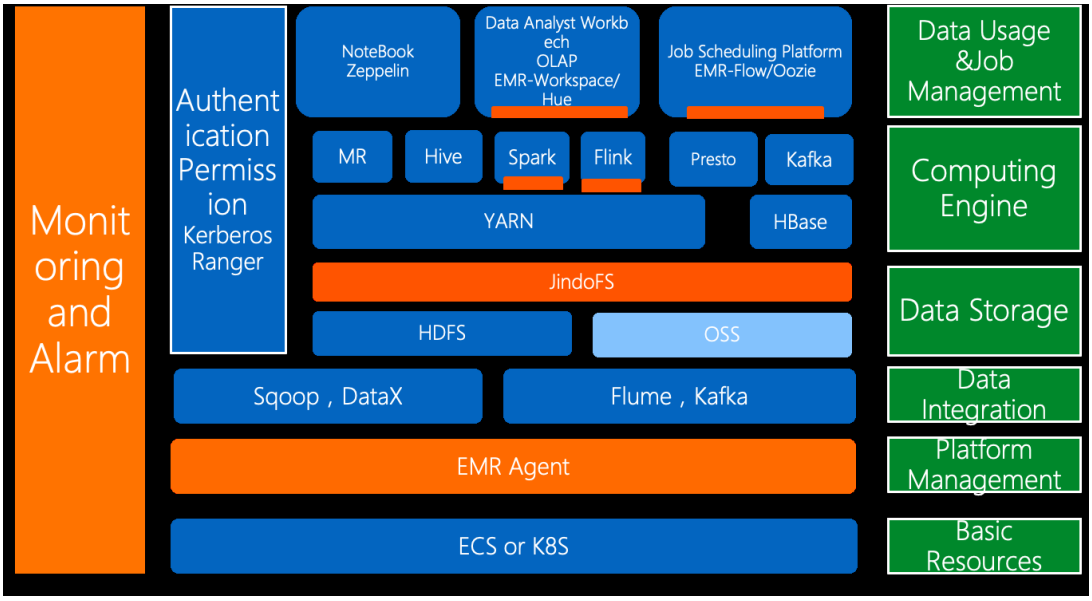
  EMR is integrated with Resource Access Management (RAM), which allows you to use Apsara Stack tenant accounts and RAM users to isolate permissions on services.

30.3. Architecture

This topic describes the architecture of EMR.

The following figure shows the architecture.

EMR architecture



30.4. Features

30.4.1. Clusters

An EMR cluster consists of Hadoop and Spark clusters that are deployed on Elastic Compute Service (ECS) instances.

EMR provides an integrated cluster management environment. It helps you select hosts, deploy environments, create, configure, and run clusters, configure and run jobs, and monitor performance. It frees you up from complex cluster building activities, such as procurement, preparation, and maintenance. You only need to focus on the processing logic of your applications.

EMR also offers different combinations of cluster services to meet requirements specific to your business. For example, to perform daily data measurement and batch computing, you only need to run the Hadoop service for EMR. To also perform streaming computing and real-time computing, you need to add the Spark service.

30.4.2. Jobs

To execute a computing task in EMR, you must create a job.

EMR supports multiple job types, such as Spark, Hadoop, Hive, Pig, Sqoop, Spark SQL, and Shell. After you select a job type, you can specify the commands to execute as well as the actions to follow if the job fails.

30.4.3. Execution plans

An execution plan is a set of jobs. You can run an execution plan on an existing cluster or a temporary cluster that is dynamically created. You can configure scheduling policies to determine whether to run an execution plan only once or on a schedule. An execution plan consumes as many resources as each job requires. This maximizes resource utilization and reduces costs.

You can run an execution plan periodically or manually:

- Periodic: You must specify the execution interval and start time. Execution plans run based on the settings.
- Manual: You must manually run an execution plan.

30.4.4. Alerts

EMR supports alerts. You can associate execution plans with alert groups.

If you enable Alert Notification on the Execution Plan page, contacts in the associated alert group receive an SMS message after each execution plan is complete. The SMS message contains the name of the execution plan, job execution results (the numbers of successes and failures), the cluster name, and the duration of execution.

31.DataHub

31.1. What is DataHub?

31.1.1. Overview

DataHub collects, stores, and processes streaming data, allowing you to analyze streaming data and build applications based on the streaming data.

DataHub is a platform designed to process streaming data. You can publish and subscribe to streaming data in DataHub and distribute the data to other platforms. DataHub allows you to analyze streaming data and build applications based on the streaming data.

DataHub collects, stores, and processes streaming data from mobile devices, applications, website services, and sensors. You can use your own applications or Apsara Stack Realtime Compute to process streaming data in DataHub, such as real-time website access logs, application logs, and events. The processing results such as alerts and statistics presented in graphs and tables are updated in real time.

Based on the Apsara system of Alibaba Cloud, DataHub features high availability, low latency, high scalability, and high throughput. DataHub is seamlessly integrated with Realtime Compute, allowing you to use SQL to analyze streaming data.

DataHub can also distribute streaming data to Apsara Stack services such as MaxCompute and Object Storage Service (OSS).

DataHub supports the following features:

- **Data queue**: DataHub automatically generates a cursor for each record in a shard, which can be considered as a logical data queue. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number of shards in the topic.
- **Offset-based data consumption**: DataHub saves consumption offsets for applications. You can resume data consumption from a saved consumption offset when your application fails.
- **Data synchronization**: Data in DataHub can be automatically synchronized to other Apsara Stack services, including MaxCompute, OSS, AnalyticDB, ApsaraDB RDS for MySQL, Tablestore, and Elasticsearch.
- **Scalable topics**: DataHub allows you to scale in or out topics by splitting or merging shards.

31.1.2. Benefits

High throughput

You can write terabytes (TB) of data into a topic and up to 80 million records into a shard every day.

Real-time processing

DataHub makes it easy to collect and process various types of streaming data in real time so you can react quickly to new information.

Ease of use

- DataHub provides a variety of SDKs for C++, Java, Python, Ruby, and Go.
- In addition to SDKs, DataHub provides RESTful APIs so that you can manage DataHub by using existing protocols.

- You can use collection tools such as Fluentd, Logstash, and Oracle GoldenGate to write streaming data into DataHub.
- DataHub supports structured and unstructured data. You can write unstructured data to DataHub, or create a schema for the data before it is written into the system.

High availability

- The processing capacity of DataHub is automatically scaled out without affecting your services.
- DataHub automatically stores multiple copies of data.

Scalability

You can dynamically adjust the throughput of each topic. The maximum throughput of a topic is 256,000 records per second.

Data security

- DataHub provides enterprise-level security measures and isolates resources between users.
- It also provides several authentication and authorization methods, including whitelist configuration and RAM user management.

31.1.3. Highlights

The highlights of DataHub features are described as follows:

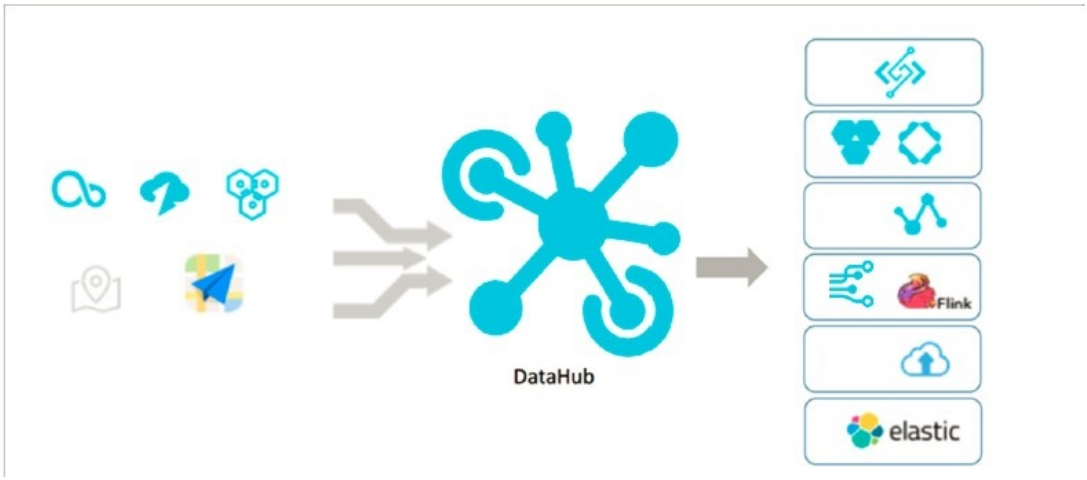
Highlights

Highlight	Description
Data security	DataHub ensures data security based on the Alibaba Cloud RAM system.
Simple O&M	DataHub automatically deactivates and recovers problematic nodes before reactivating the nodes.
Resource isolation	DataHub isolates resources between tenants.
Connection with various Alibaba Cloud services	DataHub can be used with a variety of other Alibaba Cloud services.
Scalability	The processing capacity of DataHub is automatically expanded without affecting your services. The scalability is verified during the service peak of Double 11.
Read/write performance	Records written into DataHub can be consumed repeatedly within the time-to-live of the records.
High availability	DataHub offers various high availability solutions.
Seamless integration with Alibaba Cloud services	DataHub is seamlessly integrated with various Alibaba Cloud services.

31.1.4. Scenarios

Data uploading

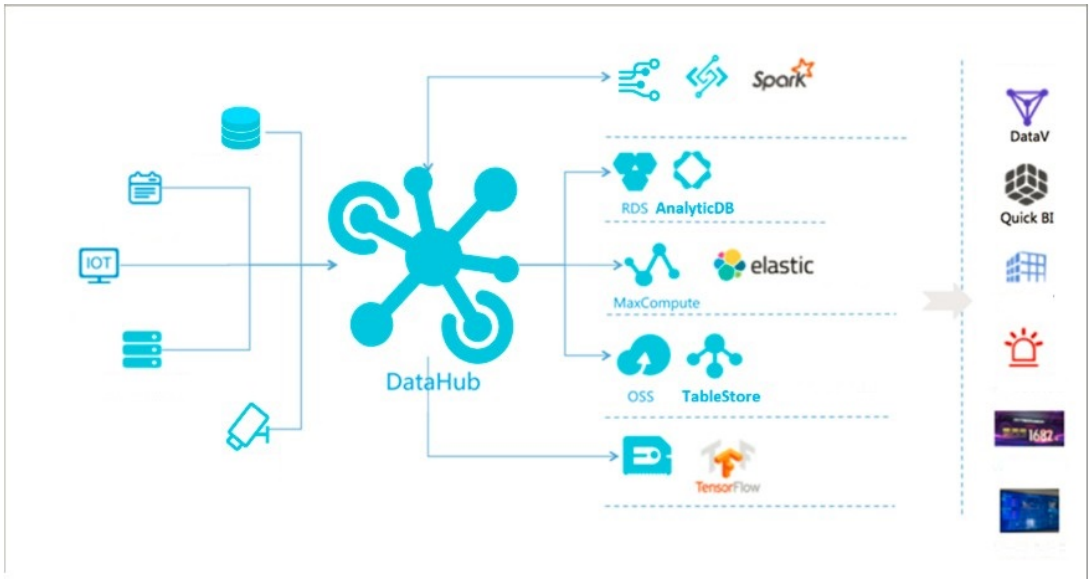
Data uploading



DataHub is connected to other Alibaba Cloud services, saving you the trouble of uploading the same data to different platforms.

Data collection

Data collection



DataHub provides several types of data collection tools for you to write your data into DataHub. DataHub supports log collection from Logstash and Fluentd, and binary log collection from Data Transmission Service (DTS) and Oracle GoldenGate (OGG). DataHub also supports the collection of surveillance videos through GB28181.

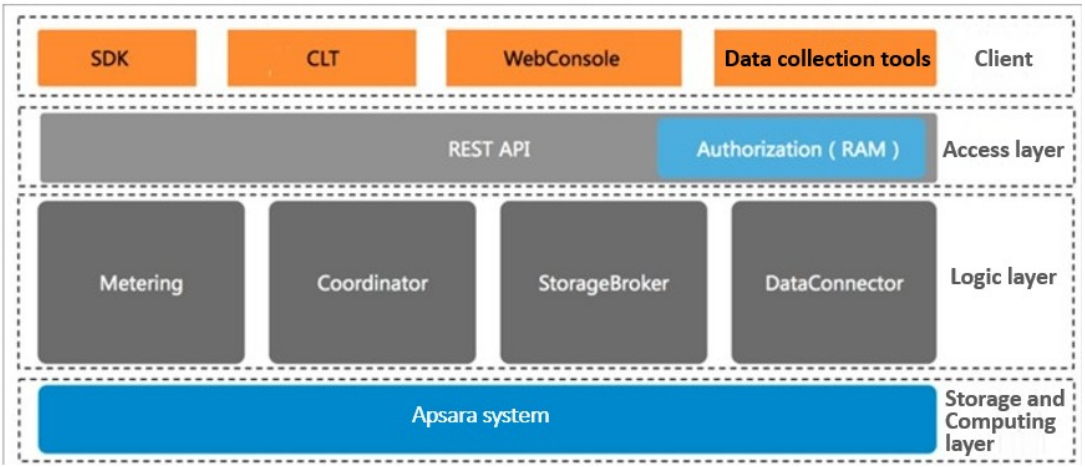
31.2. Architecture

31.2.1. Feature oriented architecture

Feature oriented architecture of DataHub shows the feature oriented architecture of DataHub.

Feature oriented architecture of DataHub

Feature oriented architecture of DataHub



The architecture of DataHub consists of four layers: **clients**, **access layer**, **logic layer**, and **storage and scheduling layer**.

Clients

DataHub supports the following types of clients:

- SDKs: DataHub provides SDKs in a variety of languages such as C++, Java, Python, Ruby, and Go.
- Command-line tools (CLTs): You can run commands in Windows, Linux, or Mac operating systems to manage projects and topics.
- Console: In the console, you can manage projects and topics, create subscriptions, view the shard status, monitor topic performance, and manage DataConnectors.
- Data collection tools: You can use Logstash, Fluentd, and Oracle GoldenGate (OGG) to collect data to DataHub.

Access layer

You can access DataHub by using HTTP and HTTPS. DataHub supports Resource Access Management (RAM) authorization and horizontal scaling of topic performance.

Logic layer

The logic layer handles the key features of DataHub, including project and topic management, data read and write, offset-based data consumption, traffic statistics, and data synchronization. Based on these key features, the logic layer is composed of the following modules: StorageBroker, Metering, Coordinator, and DataConnector.

- StorageBroker: provides data reads and writes in DataHub. This module adopts the log file storage model of Apsara Distributed File System, halving the read/write volume compared with the conventional write-ahead logging (WAL) model. This module stores three copies of data to ensure that no data is lost if a server fault occurs, and supports disaster recovery between data centers. It supports real-time data caching to ensure efficient consumption of real-time data and supports an independent read cache of historical data to enable concurrent consumption of historical data.
- Metering: supports shard-level billing based on the consumption period.
- Coordinator: supports offset-based data consumption and horizontal scaling of the processing capacity. It supports up to 150,000 QPS on a single node.
- DataConnector: supports automatic data synchronization from DataHub to other Apsara Stack

services, including MaxCompute, OSS, AnalyticDB, ApsaraDB RDS for MySQL, Tablestore, and Elasticsearch.

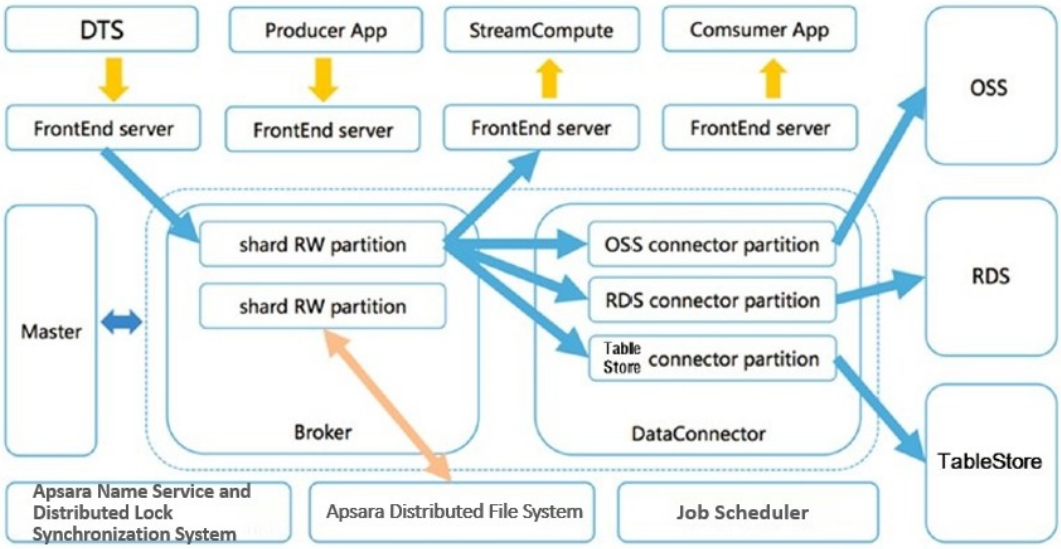
Storage and scheduling layer

- Storage: Based on the log file storage model of Apsara Distributed File System, DataHub supports append operations and solid state drive (SSD) storage. Data in each shard is stored in a separate file based on the timestamp of the data.
- Scheduling: Based on Job Scheduler, DataHub assigns shards to nodes based on the traffic on each shard. This ensures that the shards do not occupy the CPU or memory of Job Scheduler. The number of partitions on a single node has no upper limit. DataHub supports failovers within milliseconds and hot upgrades.

31.2.2. Technical architecture

Technical architecture of DataHub shows the technical architecture of DataHub.

Technical architecture of DataHub



The figure shows the process from data ingestion to consumption.

1. A shard is the smallest unit of data management in DataHub, and is a first-in, first-out (FIFO) collection of records.
2. Data in each shard is stored in a set of log files in Apsara Distributed File System.
3. The master distributes each shard to a StorageBroker. Each StorageBroker is responsible for the read and write operations on multiple shards.
4. The frontend server finds a StorageBroker based on the project, topic, and shard information specified in the request and forwards the request to the StorageBroker.
5. DataConnectors read data from the StorageBroker and forward the data to other Apsara Stack services.

Data collection

You can write data to DataHub from applications developed by using SDKs and from data collection tools such as Logstash, Fluentd, and OGG. You can also write data by using Data Transmission Service (DTS) and Realtime Compute.

Frontend server

Frontend servers constitute the access layer and support horizontal scaling. You can call RESTful API operations to access DataHub. RAM authorization is supported.

Master

The master handles metadata management and shard scheduling. It supports create, read, update, and delete operations on projects and topics. The master also supports split and merge operations on shards.

StorageBroker

StorageBrokers handle read and write operations on each shard including data indexing, caching, and file organization and management.

DataConnector

DataConnectors forward data in DataHub to other Apsara Stack services. DataConnectors provide different features for various destination services. These features include automatically creating partitions in MaxCompute and converting data streams into files stored in OSS.

31.3. Features

31.3.1. Data queue

DataHub automatically generates a cursor for each record in a shard. The cursor is a unique sequence of numbers. You can improve the performance of a topic by increasing the number shards in the topic.

31.3.2. Checkpoint-based data restoration

DataHub supports saving checkpoints for subscribed applications in the system. You can restore data from any checkpoint you saved if your subscribed application fails.

31.3.3. Data synchronization

Data in DataHub is automatically synchronized to other Alibaba Cloud services.

DataConnector

You can create a DataConnector to synchronize DataHub data in real time or near real time to other Alibaba Cloud services, including MaxCompute, OSS, Elasticsearch, ApsaraDB RDS for MySQL, AnalyticDB, and Table Store.

You can configure the DataConnector so that the data you write to DataHub can be used in other cloud platforms. At-least-once semantics is applied in data synchronization. This ensures that no data is lost, but may result in duplicated records in the destination platform if an error occurs during the synchronization process.

Destination platforms

The following table describes the platforms to which DataHub records can be synchronized.

Destination platforms

Destination platform	Timeliness	Description
MaxCompute	Near real-time. Latency: 5 minutes.	The column names and data types in the source topic must be the same as those in MaxCompute. The MaxCompute table must have one or more corresponding partition columns.
OSS	Real-time	Records are synchronized to the specified bucket in OSS and are saved as CSV files.
Elasticsearch	Real-time	Records are synchronized to the specified index in Elasticsearch. Records may not be synchronized in the order of the recording time. If you want to synchronize data in the order of the recording time, you must write the records with the same partition key into the same shard.
ApsaraDB RDS for MySQL	Real-time	Records are synchronized to the specified table in ApsaraDB RDS for MySQL.
AnalyticDB	Real-time	Records are synchronized to the specified table in AnalyticDB.
Table Store	Real-time	Records are synchronized to the specified table in Table Store.

31.3.4. Scalability

The throughput of each topic can be scaled by splitting or merging shards.

You can adjust the number of shards in a topic according to the service load.

For example, if the topic throughput cannot handle a surge in the service load during Double 11, you can split existing shards to up to 256 to increase the throughput to 256 MB/s.

As the service load decreases after Double 11, you can reduce the number of shards as needed by performing the merge operation.

32.Quick BI

32.1. What is Quick BI?

Alibaba Cloud Quick BI is a cloud computing-based service that supports big data analysis and data visualization. It is a lightweight, easy-to-use BI tool. You create data sources and datasets to perform ad hoc queries and analysis on data. You can use workbooks and dashboards to visualize data.

Background

As IoT develops at a high speed, the volume of data increases sharply. It has become increasingly important to analyze and use data to generate commercial value. Quick BI achieves efficient data analysis and makes everyone a data analyst. Quick BI supports online ad hoc analysis, drag-and-drop operations, and data visualization, which helps you easily analyze data and monitor your business. Quick BI is a tool for business personnel to view data and a booster for data-driven operations, which helps solve *the last mile problem* of big data applications.

Status quo

Today, more and more enterprises migrate their data to the cloud, so enterprise data is in distributed storage. However, governments and financial institutions build on-premises databases for security concerns. Quick BI connects to various data sources to meet different requirements of on-cloud and on-premises deployment and creates datasets based on the data sources for centralized scheduling. Quick BI can connect to the following data sources:

- MaxCompute (formerly ODPS) and ApsaraDB for RDS
- User-created MySQL and SQL Server databases that are hosted on ECS
- VPC data sources

Challenges

Quick BI faces the following challenges:

- Provide better service and improve customer experience.
- Accelerate the construction of traditional BI and big data systems with minimal costs.
- Improve business insights with effective business monitoring and related business data analytics.

32.2. Benefits

The benefits of Quick BI can be summarized as quick response, powerful capabilities, high security, visualization, and user-friendliness.

Rapid data modeling

You can create a dataset with a few clicks. This reduces your reliance on professional staff.

Powerful data analysis

Quick BI generates professional workbooks that allow you to jointly analyze data online and generate reports, such as daily, weekly, and monthly reports. The more than 300 regular data analysis functions allow you to easily acquire business analysis results.

Reliable data access control

Quick BI adopts an access control list (ACL) system. An access object is used as a control unit for permission approval and authorization. Quick BI also provides row-level permission control to implement fine-grained access control.

Diversified data visualization

Quick BI provides more than 30 chart types to achieve effective data visualization. In addition, Quick BI adapts to various terminals, so that the works created on Quick BI can be accessed from various terminals. This significantly improves the efficiency of data analysis.

User collaboration

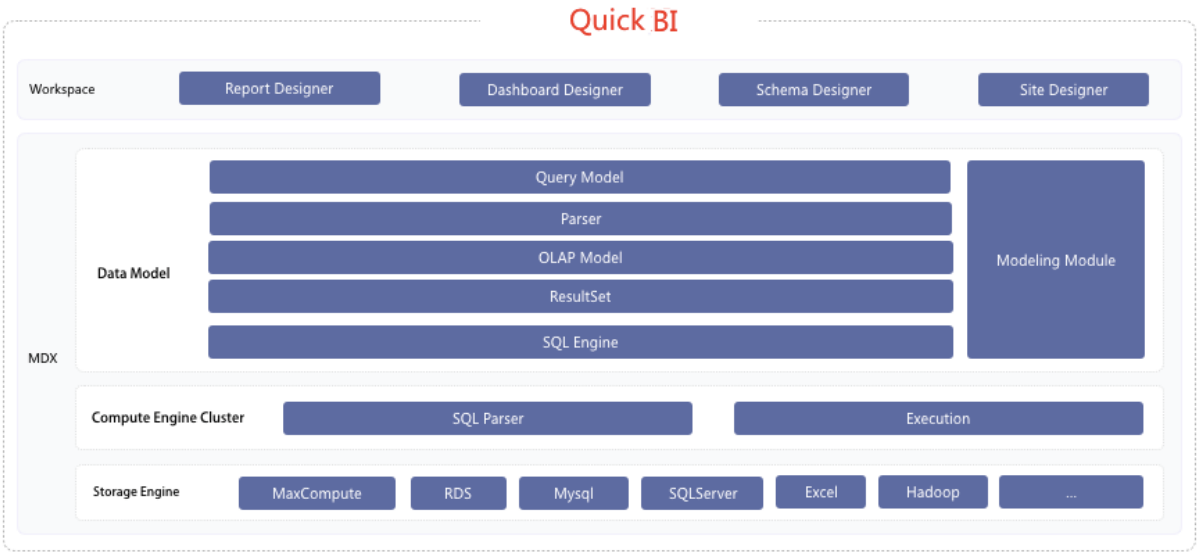
All analysis objects are online. Users of an enterprise can be organized in a workspace. Quick BI allows the members in a workspace to collaboratively analyze data.

32.3. Product architecture

32.3.1. System architecture

This topic describes the system architecture of Quick BI.

The following figure shows the system architecture of Quick BI.



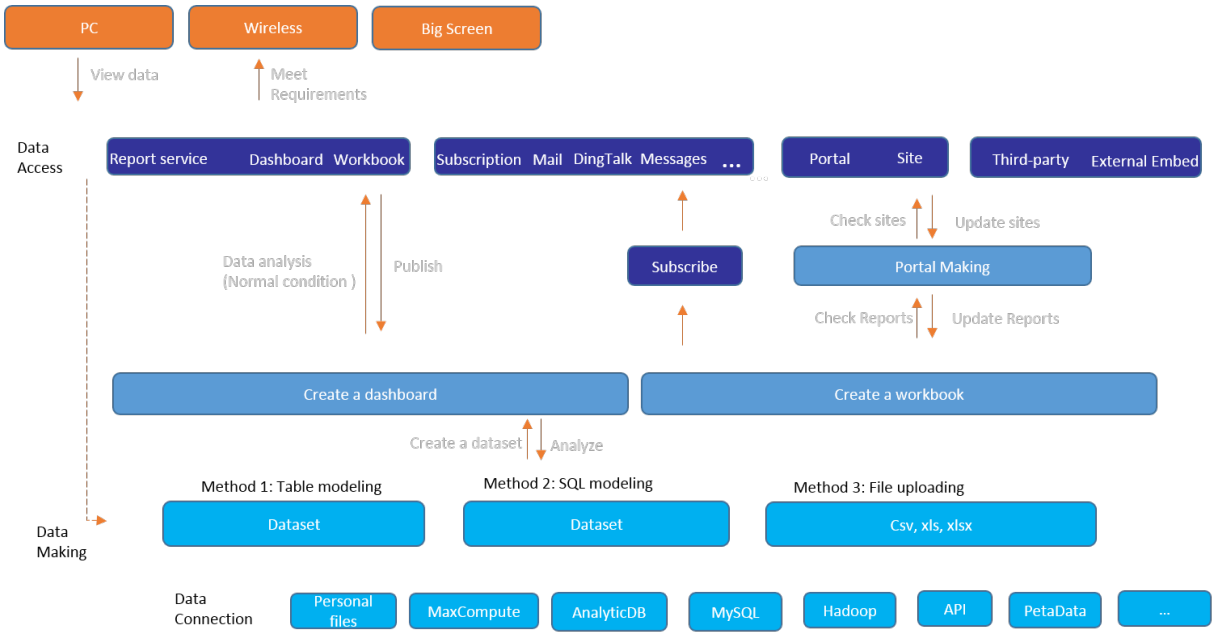
The system architecture of Quick BI consists of four function modules, including data connection, computing engine, data management, and data analysis.

- **Data connection**: establishes a connection with a database to achieve ad hoc data analysis.
- **Computing engine**: efficiently generates SQL statements based on constructed datasets and user-defined conditions and executes them in a database.
- **Data management**: creates datasets that support multi-dimensional analysis based on tables, SQL statements, and local files, providing dataset management and join operations.
- **Data analysis**: provides dashboards and workbooks to support enterprise data analysis and ad hoc query.

32.3.2. Components

This topic describes the components of Quick BI.

The following figure shows the topology of Quick BI.



Data source

Quick BI can read data from various data sources, such as relational databases, MaxCompute, and local files. All data is stored in data sources, and Quick BI does not copy data from the data sources.

Dataset

A dataset is the smallest object for data analysis in Quick BI. It can be data of a specific theme or data from some scenario components.

Datasets support join operations to increase the number of dimensions or measures. For example, you can associate a dataset with another to analyze transactions. Join operations require no SQL statements and help you easily build analysis objects with powerful functionalities.

The date dimension can be constructed by multiple granularities, including day, week, month, quarter, and year. In addition, you can obtain the value accumulated in a month, quarter, or year.

Dashboard

Quick BI dashboards provide data analysis components, support filter interactions among components, and offer a wide range of component settings.

The dashboards provide mainstream data analysis components, such as vertical bar charts, maps, and funnel charts. In addition, the dashboards provide rich functional components, such as the query control, iFrame, and tab. Quick BI allows you to design beautiful dashboards with flexible components. This reduces your business operation costs and your reliance on professional staff.

Workbook

Workbook is a unique feature of Quick BI that offers online data analysis similar to spreadsheets. A cell in a workbook is a data unit. Workbooks allow you to copy data from local data sources and to retrieve data from datasets. All cells are linked and support the following features:

- Built-in formulas, data aggregation, more than 300 functions, and cross-sheet references

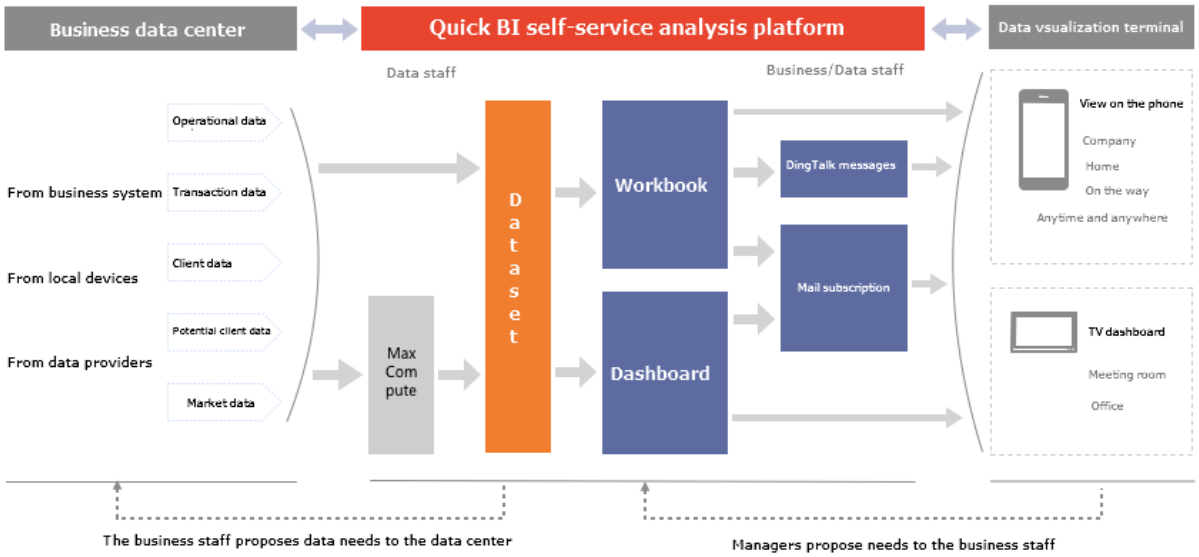
- Interactive drag-and-drop operations, and row and column freezing
- Visualization of analyzed data
- View, edit, and preview modes

BI portal

A BI portal is a set of dashboards and workbooks organized in the form of menus based on the requirements of a scenario, department, or marketing plan. BI portals allow you to check the status of all businesses.

Monitoring dashboard

The monitoring dashboard enables you to view data. You can use the single-screen dashboard of Quick BI to create reports and display them in full-screen mode on your TV monitor and other terminals. This allows you to view the status of your business.



32.3.3. Deployment

Quick BI is automatically deployed in the Apsara Infrastructure Management Framework console.

The following table lists resources required by Quick BI deployment.

Item	Specification	Remarks	Scalability
Quick BI server	Two Docker containers. Each container has 16 GB of memory and 8 cores.	Console page	Scale-out
QuicK BI agent	Two Docker containers. Each container has 16 GB of memory and 8 cores.	Computing server	Scale-out

Item	Specification	Remarks	Scalability
Redis primary node	One Docker container. It has 16 GB of memory and 8 cores.	Cache	Scale-out
Redis secondary node	Two Docker containers. Each container has 16 GB of memory and 8 cores.	Cache	Scale-out
dbinit	One Docker container. It has 1 GB of memory and one core.	Metadata initialization	None
test	One Docker container. It has 1 GB of memory and one core.	Periodic monitoring system	None

32.3.4. Server roles

This topic describes the server roles of Quick BI.

Quick BI consists of the following server roles:

- base-biz-yunbi-dbinit: a database initialization component that initializes the basic metadata of Quick BI. It is a prerequisite for Quick BI to run properly and must be at desired state.
- quickbi-redis-slave: the secondary node of Redis. It provides the data caching function for Quick BI to improve system query performance.
- quickbi-redis-master: the primary node of Redis. It provides the data caching function for Quick BI to improve system query performance.
- base-biz-yunbi-executor: an executor component that queries the data and metadata of the table in the connected data source.
- base-biz-yunbi: a web service component that provides web services and allows you to visit Quick BI web pages.
- ServiceTest: an automated testing component that executes test cases to test the service availability of Quick BI.

32.4. Features

This topic describes the features of Quick BI.

Quick BI provides the following features:

- Supports a wide range of data sources, such as MaxCompute, relational databases, and local files.
- Quickly analyzes offline data sources. For example, Quick BI can analyze a 100 GB file in 10 seconds.
- Provides complete workbooks to enable you to easily make complex reports.
- Enables everyone to easily learn and use.
- Provides you with diverse options for data visualization, and automatically identifies data properties to generate the most appropriate charts.
- Provides strict permission control and adopts multi-layer verification to ensure data security.

- Provides an OLAP analysis engine that has comprehensive functions and is easy to use.
- Supports collaborative operations, allowing multiple users to analyze data cooperatively.

33.Graph Analytics

33.1. What is Graph Analytics?

Graph Analytics is an intelligent visual analysis platform for relationship networks.

Graph Analytics is designed to facilitate **multi-source data integration, computing applications, visual analytics, and intelligent businesses**. Based on relationship networks, Graph Analytics can visualize the properties of objects and reveal the relationship among objects.

Development of Graph Analytics

Based on years of practice in multiple industries, Graph Analytics has made impressive progresses and evolved with the development of visual analysis and intelligent network analysis. Featuring the OLEP model, Graph Analytics helps users analyze data and relationship networks with ease. Graph Analytics supports data source integration and major compute engines, and can be applied to multiple business scenarios. Graph Analytics aims to build a highly efficient and intelligent platform for network analysis.

Graph Analytics provides multiple features, including relationship networks, , search networks, information cubes, intelligent analysis, collaboration and sharing, dynamic modeling, and pattern recognition. With its visual interfaces, Graph Analytics integrates machine computing capabilities with human cognition to provide users with data insight and help users obtain information and knowledge more efficiently.

Graph Analytics is oriented to intelligence analysis in the public security, industry and commerce, taxation, customs, banking, insurance, and Internet finance fields. Graph Analytics provides strong support for case analysis, investigations in money-laundering, fraud, and corruption, and correlated transaction cases. This platform helps analysts gain the key information from massive data and find out case-solving clues and valuable intelligence with ease.

Graph Analytics has been widely used in Alibaba Group and Ant Financial for risk control, such as anti-fraud, anti-theft, and anti-money laundering solutions.

Background and challenges

Background: Due to the drastic increase in available information and data, Graph Analytics faces the challenge of obtaining useful intelligence from massive data.

Challenges: Receiving massive amounts of complex data from multiple sources, the key challenge for Graph Analytics is to obtain useful information in a highly efficient and intelligent manner.

33.2. Benefits

This topic describes the technical advantages of Graph Analytics.

OLEP model

Graph Analytics developed the OLEP metadata model based on ontological theories. Unlike a conventional physical data warehousing model that consumes both time and effort, the OLEP model is a model based on the **object-link-property** logic and built for detailed data from multiple data sources. After you understand and sort your business data, you can quickly configure and define the business logical model, the mapping relationship between the logical model and the physical data, and the application scenario parameters in Administration Console. After you complete these configurations, the business analysis is modeled and defined.

Powered by the OLEP model, Graph Analytics supports real-time deployment and allows you to add, delete, or modify the graph pattern dynamically without cleansing data. Graph Analytics also supports a full range of features such as data integration across multiple data sources, relationship analysis, graph algorithm mining, and offline data integration. Graph Analytics supports cross-database and cross-engine data integration based on the logical mapping between the OLEP model and data, so that you can use the same data multiple times with no need to cleanse the data for relationship analysis.

Visual interface

Graph Analytics uses its visual interactive interfaces to present large amounts of data in multiple forms, including relationship networks, map analysis, information cubes, and a behavior chronology pane. This enables users to perform analyses and investigations based on the network, time, and space. Graph Analytics also supports various commonly used tools to fit with the habits of users and improve user experience.

Support for multiple data engines

Graph Analytics supports multiple data engines. It can handle several exabytes of data and process tens of billions of nodes or links.

Relationship network models and algorithms

To optimize data analysis, Graph Analytics provides multiple relationship network models and algorithms for data mining. Graph Analytics combines classic network analysis methods with machine learning and business algorithms to perform intelligent analyses, including following relationship analysis, backbone analysis, path analysis, intimacy analysis, and key location analysis. This makes it easy for you to perform data mining on relationship networks.

Intelligent network

Graph Analytics recognizes various intelligent network patterns. It extracts a user-defined graph pattern from large amounts of data and uses optimized graph algorithms to match subgraphs that have the same pattern.

API

Graph Analytics integrates data from multiple sources by using the OLEP model. It provides a multitude of features such as relationship network analysis, graph algorithm mining, and offline data integration. Graph Analytics provides an open API for custom development based on years of project experience and proven business algorithms. Graph Analytics provides an API for you to call specific operations as needed. The operations cover the following features:

- Object and link search
- Relationship network powered by years of business experience
- GIS service
- Label system and intelligent network

You can integrate the relationship network analysis capability of Graph Analytics with your project system, and customize and define the analysis features to suit your business needs. This API also supports the following features:

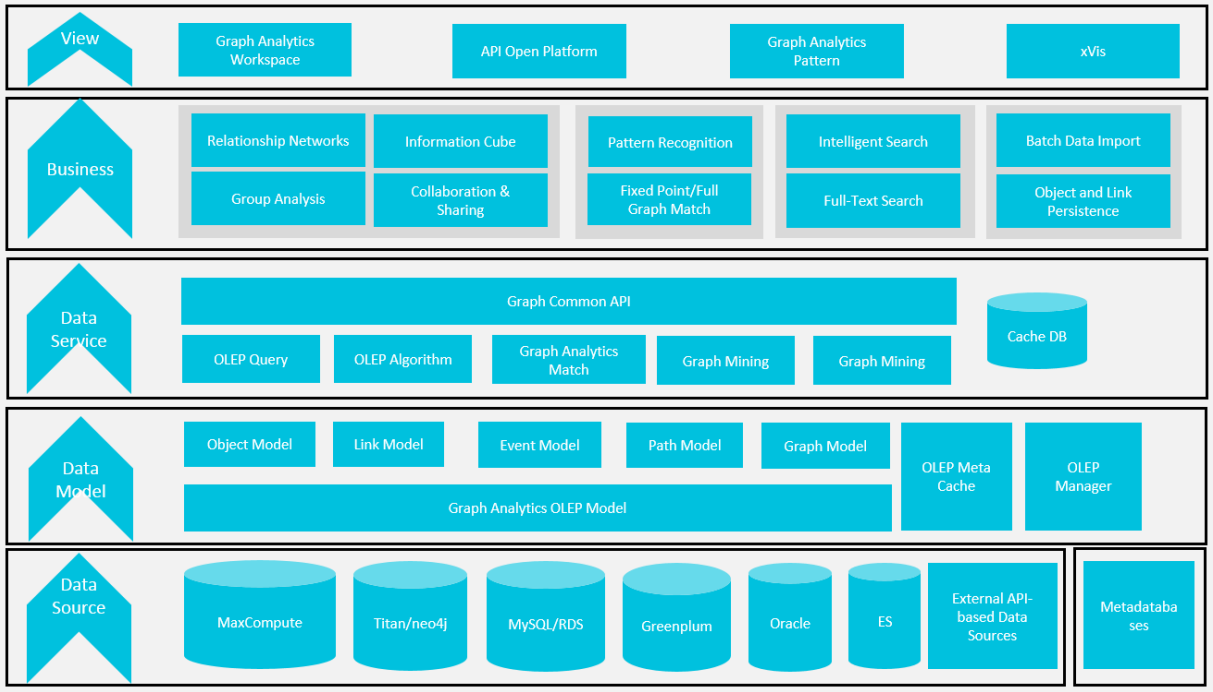
- Allows you to customize API variables to define your own project and run the project efficiently.
- Provides multiple entries for you to switch between systems seamlessly.

33.3. Product architecture

33.3.1. System architecture

This topic describes the system architecture of Graph Analytics.

Graph Analytics provides multiple components and a multi-layer architecture, including the data source layer, data model layer, data service layer, business layer, and the view layer.



Data source layer

Based on the Alibaba Cloud Big Data platform, the data source layer can store and handle petabytes or exabytes of data. It provides powerful data integration, processing, analysis, and computing capabilities. The data source layer provides the following features:

- Supports open source graph databases, such as Titan and Neo4j.
- Supports open source relational databases, such as MySQL, RDS, Oracle, and Greenplum.
- Supports NoSQL databases, including Elasticsearch and KV HBase, a database where each row is a key/value pair.
- Supports external API-based data sources.
- Supports the integration, processing, and online calculation of data from multiple sources.

Data model layer

The data model layer supports the following features:

- Established based on ontological theories, the OLEP model studies the objects, relationships between natural objects, relationships between social objects, and event information.
- Various types of data are converted into nodes and links in the graph. Based on these nodes and links, Graph Analytics builds paths and graph models to lay the foundation for a subgraph model, providing a standardized data model for data mining and graph algorithm calculation.

Data service layer

The data service layer provides link queries, relationship mining, and graph algorithms for you to analyze relationship networks. This layer supports pattern recognition and extracts graph structure data that is matched with the user-defined graph pattern.

Business layer

The business layer supports the following features:

- Graph Analytics provides an API to call application components at the analysis layer. These application components include relationship networks, search networks, information cubes, intelligent judgement, collaboration and sharing, and dynamic modeling.
- Supports intelligent networks, including pattern definition and pattern matching features.

View layer

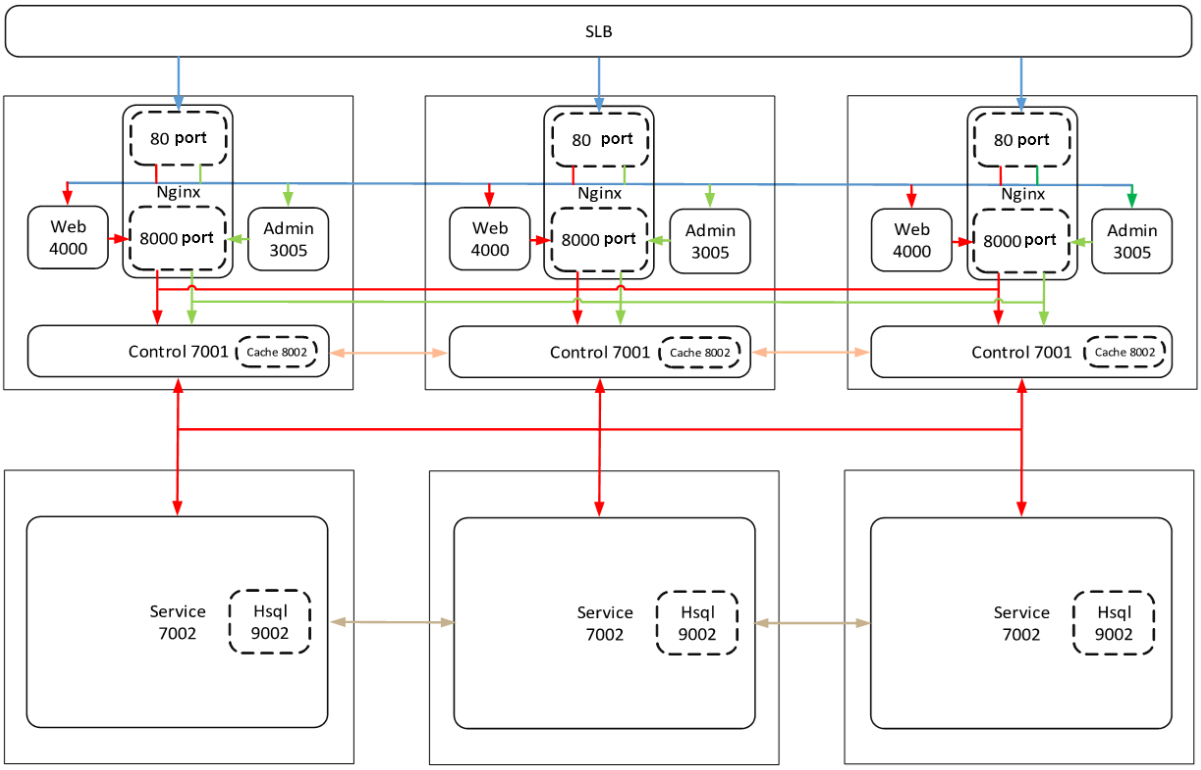
The view layer refers to the Web layer of Graph Analytics. This layer displays the entire graph, and its features are as follows:

- Supports multiple layouts of relationship networks to fit with different business scenarios.
- Graph Analytics provides a diversified, visual, and interactive analysis interface and supports various terminals.
- Graph Analytics provides visual components and external APIs and supports third-party system integration.

33.3.2. Architecture

This topic describes the architecture of Graph Analytics.

In Graph Analytics, control nodes are separated from service nodes. These nodes support both single server deployment and clustered server deployment. The following figure shows the architecture of Graph Analytics:



Reliability and security

Graph Analytics helps you build a reliable and secure system with the following features:

- The web and the control nodes support stateless load balancing to transfer the load stress from the server where a Node.js failover occurred to another server, without affecting other features.
- Multiple control nodes are deployed. The sessions of these control nodes are synchronized in real time. The servers are stateless. If one server is down, the other servers can still function, and users are not affected.
- Control nodes are used to restrict connection requests to the service nodes, and allocate servers based on the number of users and the usage of resources. If some users query large amounts of data, the operation occupies most of the resources of the current server, but does not affect users on other servers.

Performance and stability

Graph Analytics uses the following features to improve performance and stability of the system:

- The underlying layer selects data sources based on the business scenario. Graph Analytics supports graph database (GDB) and high-performance databases and allows you to perform a quick analysis on large amounts of data.
- The compute layer supports in-memory databases and allows you to perform the second queries and statistics quickly based on the results of the first query stored in the in-memory database.
- Service nodes are allocated based on the available resources and the available number of users to improve system performance and maximize resource utilization.
- To ensure a stable system, Graph Analytics uses control nodes to throttle traffic and avoid frequent API calls.

Scalability

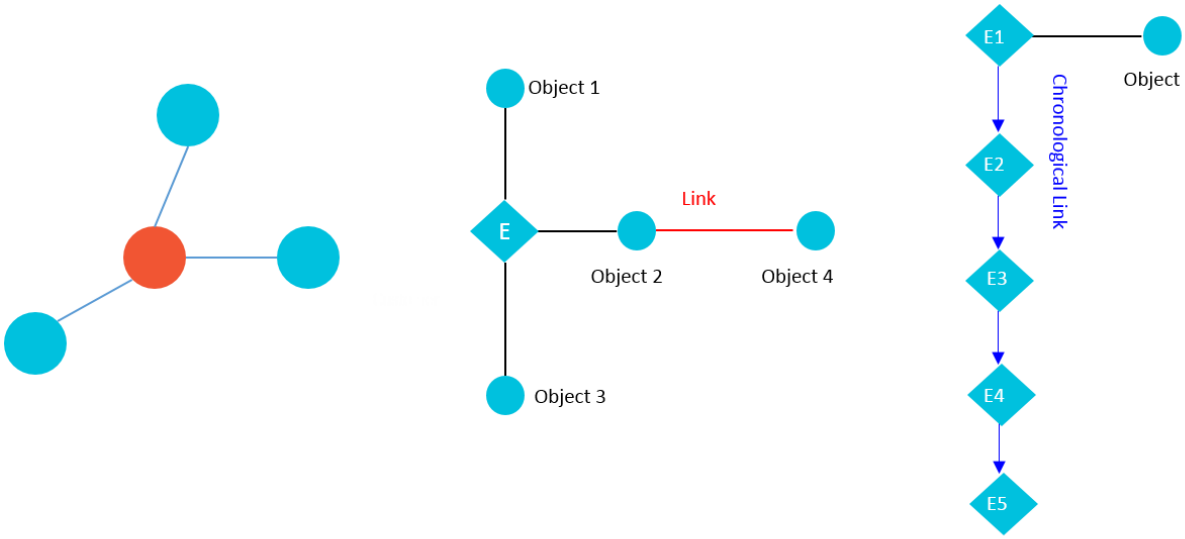
Graph Analytics supports clustered deployment and clustered scale-out.

33.4. Features and principles

33.4.1. OLEP model

Graph Analytics uses the OLEP model instead of the conventional data warehouse model that requires a large amount of time and effort from the user.

The following figure shows how the OLEP model works.



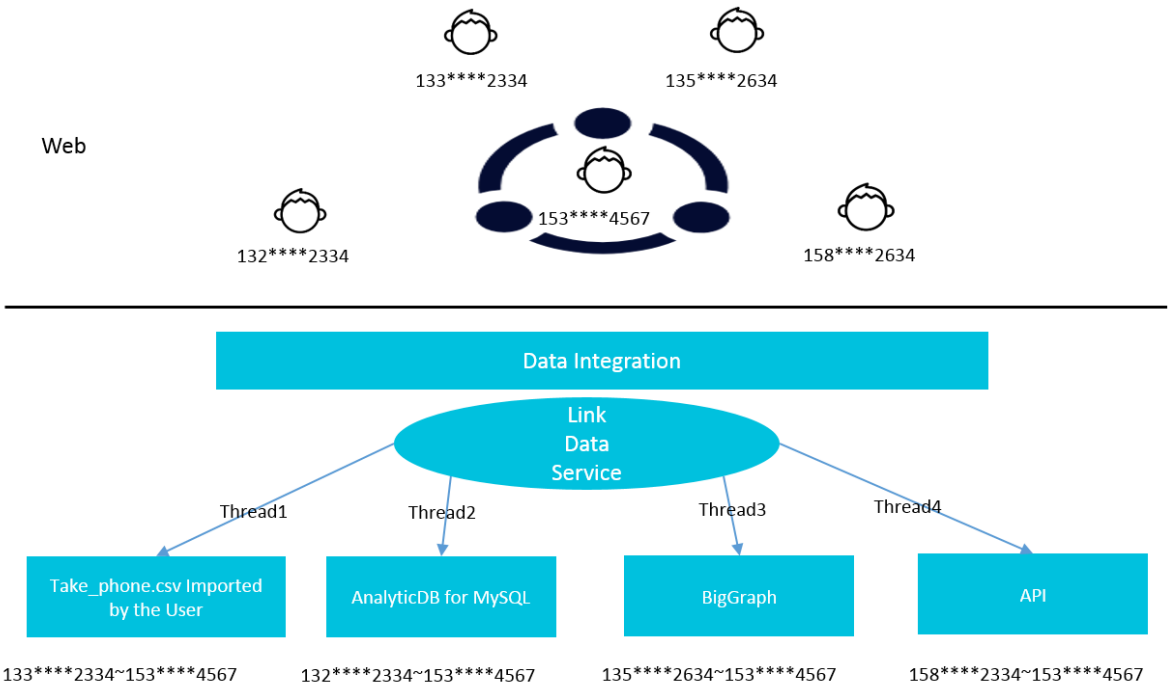
OLEP Model Event-Link Model Chronological Model

In Graph Analytics, a table can be mapped to multiple objects, links, and events. Columns in the table will be mapped to the properties of objects, links, or events. Based on the OLEP model, every detail record will be mapped to the node, link, and property model of the graph.

33.4.2. Data integration

Graph Analytics uses the OLEP model to integrate data retrieved by major search engines.

The following figure shows how Graph Analytics integrates data.



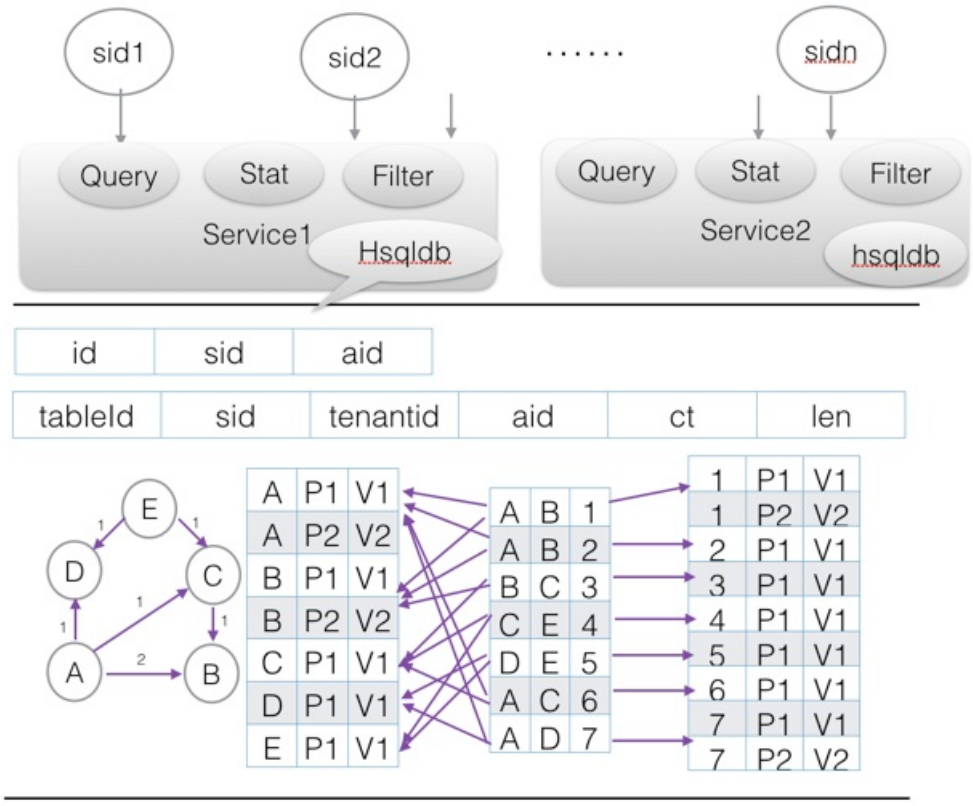
Powered by the OLEP model, Graph Analytics supports major data engines, common SQL syntax, the gremlin language, and API calls.

Graph Analytics supports concurrent processing I/O requests by multiple threads, and simultaneously queries data retrieved by multiple engines that is fit into the OLEP model. Graph Analytics can model and integrate the queried data, and perform visual data analysis based on the graph structure.

33.4.3. Separate the graph structure logic from graph details

Graph Analytics separates the graph structure logic and graph details to facilitate large-scale graph analysis.

To separate the graph structure logic and graph details is to store the graph structure in the user's browser while storing the object and link properties in a remote server, as shown in the following figure.



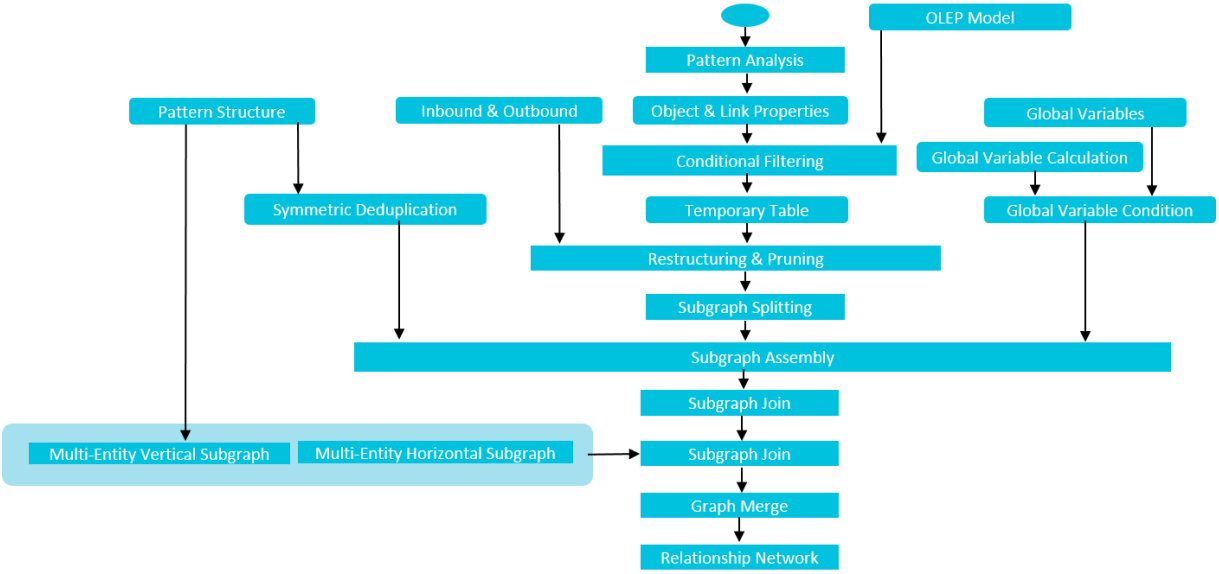
The separation is described as follows:

- The graph structure is stored in the user's browser, so the user can arrange the graph layouts and analyze the graph with ease.
- The object and link properties are stored in a remote server. Graph Analytics also has introduced HyperSQL DataBase (HSQLDB) and designed a columnar storage structure to ensure highly-efficient detail queries, statistics, and data filtering. The detailed data is split by analysis. Each user has an independent analysis space with clear structures and can perform queries efficiently.

33.4.4. Intelligent network

The intelligent network feature enables Graph Analytics to easily extract graph pattern data that matches the user-defined graph pattern from large amounts of data.

The following figure shows how the intelligent network feature works based on the OLEP model:



The intelligent network feature matches correlated data with a pattern and stores the matched data as intermediate results in a temporary table. Then, Graph Analytics extracts the features of the user-defined pattern, and uses these features as conditions to filter other intermediate results. Then, Graph Analytics uses graph algorithms to calculate the amount of relationship data.

Graph Analytics analyzes the overall graph pattern, splits the non-fixed patterns, including linear relationship patterns and intricate relationship patterns, and decomposes a graph into multiple subgraphs starting from the least-related data. Graph Analytics analyzes these subgraphs and locates duplicate data in the subgraphs by using global variables, and merges the data by using the SQL JOIN statement multiple times. Then, the subgraphs are merged with the subgraphs that have non-fixed patterns.

Graph Analytics queries the key nodes in the pattern, and merges subgraphs that have the same key nodes. In addition, Graph Analytics groups the results, queries relationship network data, and merges the results with the features of the same pattern.

34.Apsara Big Data Manager (ABM)

34.1. What is Apsara Big Data Manager?

Apsara Big Data Manager (ABM) is an operations and maintenance (O&M) platform tailored for big data services.

Background information

In the daily use of the Alibaba Cloud big data platform, O&M engineers and data service developers need to manage various big data services. These services include offline computing engines, real-time computing engines, analytic and query engines, AI platforms, and big data applications. These big data services are sometimes closely related to each other. To improve O&M and development efficiency, an end-to-end O&M platform is required to integrate these services. ABM is developed against this background.

Supported services

ABM supports the following services:

- MaxCompute
- DataWorks
- StreamCompute
- Quick BI
- DataHub
- Machine Learning Platform for AI
- E-MapReduce

ABM supports O&M on big data services from the perspectives of business, services, clusters, and hosts. You can also update big data services, customize alert configurations, and view the O&M history in ABM.

On-site Apsara Stack engineers can use ABM to easily manage big data services. For example, they can view resource usage, check and handle alerts, and modify configurations.

Challenges

The stability of a big data service may be adversely affected by an unstable platform and poor service implementations, such as slow or bad SQL queries, data skews, or long tail queries. O&M engineers alone are not able to ensure service stability. Instead, service developers and O&M engineers must work together to troubleshoot issues and improve stability.

In addition to an O&M platform for O&M engineers, ABM aims to provide data-based and intelligent O&M capabilities. It is exploring ways to implement cross-computing engine management and add O&M support for more services, such as big data applications, computing engines, scheduling systems, storage, operating systems, and networks.

34.2. Benefits

Based on a mature O&M mid-end, ABM can quickly connect to big data products and provide comprehensive O&M capabilities for each product.

O&M mid-end

With the O&M mid-end, ABM provides various built-in services and SDKs to enable all-around operations and maintenance capabilities. Each product can easily connect to ABM and has an exclusive site to implement operations and maintenance. Compared with the traditional development process, ABM provides a more visualized, configuration-based, and function-based alternative and minimizes the development costs of business customization.

The O&M mid-end provides the following services in Apsara Stack:

- Job platform: supports visualized job management, execution, and scheduling. This satisfies various needs of visualized O&M.
- Knowledge graph: supports data storage, integration, and query in different scenarios. This resolves the difficulties in integrating and querying dispersed data.
- Function as a service (FaaS): supports low-cost trial and error, fast code development, and function-based business logic management. This relieves users from complex project organization, dependency management, deployment, and scaling and allows them to focus on business.
- Application management: stores business logic and configurations in a hierarchical way, and supports highly flexible extensions. This allows users to create complex application structures with simple configurations by using JSON.
- Inspection service: provides a universal solution for checker management and scheduling, and supports disparate alert data sources. The inspection service can be embedded into any page of an application site.
- Third-party system adaptation: allows users to use one SDK to call APIs of all connected third-party systems.
- Authorization proxy: adapts to AAS and OAM in Apsara Stack, provides capabilities such as visualized user management and authorization management, and satisfies the authorization and authentication requirements of third-party systems.
- Gateway service: integrates all service APIs so that external systems can call these APIs uniformly. In addition, isolation, decoupling, and scaffold capabilities are provided for authenticating and processing all requests centrally.
- Apsara Infrastructure Management Framework synchronization: adapts to the Apsara Infrastructure Management Framework base in Apsara Stack, and provides encapsulated interfaces for querying and managing all host data.
- Tunnel service: uses StarAgent to shield the differences of underlying command execution tunnels and provides a universal interface. This allows users to deliver commands and files to a large number of hosts, and aggregate and query the statuses of these hosts.

Quick service development

Based on the O&M mid-end, ABM now supports multiple products and provides stable and reliable operations and maintenance capabilities for them. Supported products include MaxCompute and DataWorks.

34.3. Architecture

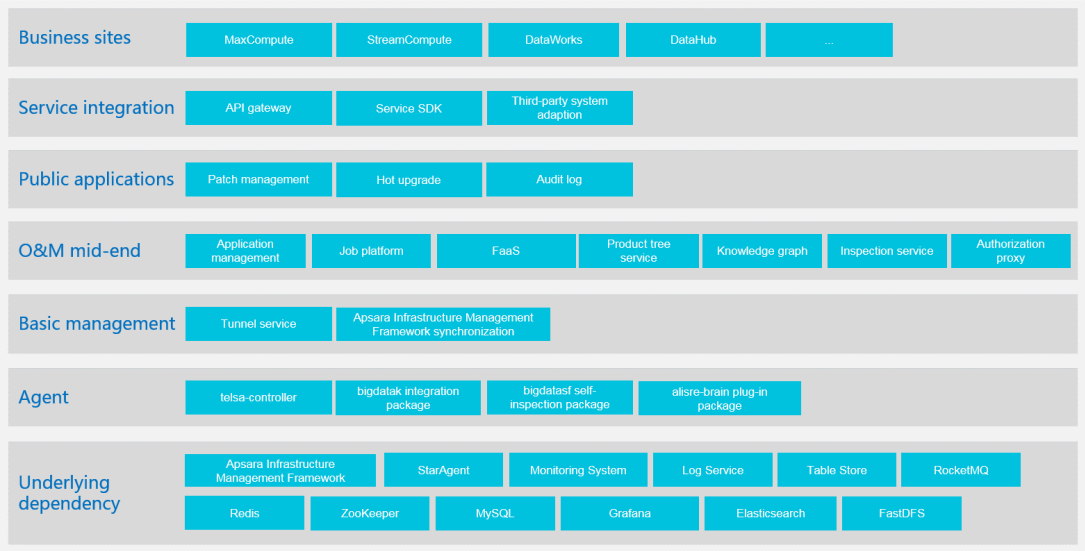
34.3.1. System architecture

This topic describes the system architecture of Apsara Big Data Manager (ABM) and the functions of each component.

ABM uses a microservice architecture that supports data integration, interface integration, and feature integration through a unified platform, and provides standard service interfaces. This architecture enables a consistent user interface, which means that O&M operations are the same for all products. This reduces training costs and lowers O&M risks.

The ABM system consists of the following components: underlying dependency, agent, basic management, O&M mid-end, public applications, service integration, and business sites.

Architecture



Underlying dependency

ABM depends on open source systems from Alibaba and third parties.

- Uses StarAgent and Monitoring System of Alibaba to run remote commands and remote data collection instructions.
- Uses ZooKeeper to coordinate primary and secondary services. This guarantees high availability of services.
- Uses RDS to store metadata, Redis to store cache data, and Table Store to store large amounts of self-test data. This improves service throughput.

Agent

The agent provides client SDKs, scripts, and monitoring packages to be deployed on managed servers.

O&M mid-end and basic management

The O&M mid-end and basic management components form the base of the ABM system. Each service in these two components provides different capabilities for business sites. This enables quick construction of business sites and makes the capabilities of each business site complete.

Public applications

Public applications are developed based on the O&M mid-end and designed with special purposes. These applications are adaptive to all big data products supported by ABM.

Service integration

Service integration links business sites with underlying components. It integrates interfaces of all internal services, adapts to various third-party systems, and provides a unified SDK for users.

Business sites

Business sites are built based on the O&M mid-end and cover all big data products, including MaxCompute, StreamCompute, DataWorks, and DataHub. Each business site provides comprehensive O&M capabilities for one product.

34.4. Features

34.4.1. Small file merging

This topic describes the small file merging feature of ABM for MaxCompute.

What are small files

Apsara Distributed File System stores data in blocks. The size of each block is 64 MB. Small files in this topic refer to files whose size is less than 64 MB. Reduce computing or real-time data collection through tunnels will generate a large number of small files.

Impacts of small files

- More small files consume more instance resources. In MaxCompute, a single task instance can handle up to 120 small files. Therefore, too many small files cause a resource waste and deteriorate system performance.
- Too many small files cause high pressure on Apsara Distributed File System, and decrease the utilization rate of disk space.
- Too many small files occupy a large amount of memories of Master servers and Chunkservers in Apsara Distributed File System. When the memory usage exceeds 50% of the safety limit on a Master server of Apsara Distributed File System, the cluster stability is affected.

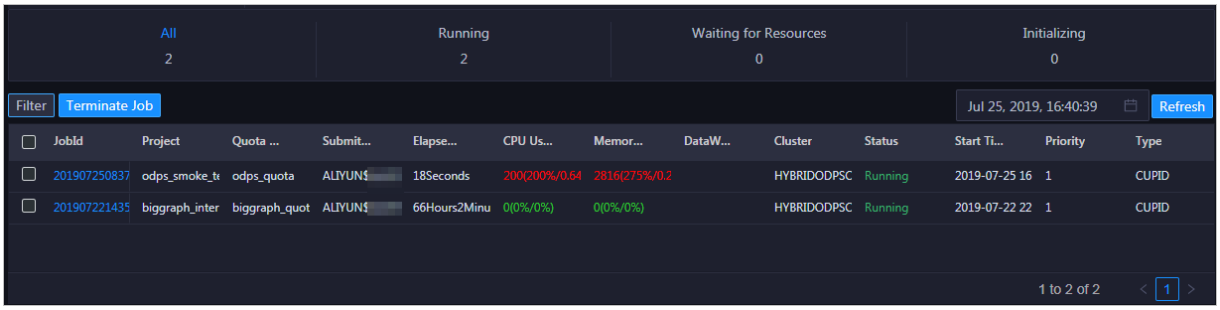
Method of merging small files

ABM uses the MaxCompute SDK to generate merge tasks for merging small files. This method increases merging concurrency to the maximum extent. Currently, you can create merge tasks by cluster or project. You can configure whether to allow merge tasks to run concurrently and specify the start and end time for each merge task.

34.4.2. Job snapshot

This topic describes the job snapshot feature of ABM for MaxCompute.

In this topic, all jobs refer to MaxCompute jobs. When a job is executed, ABM saves detailed job logs. These logs are used to generate a job snapshot. The following figure shows an example of the job snapshot page.



The job snapshot feature supports the following functions:

- Displays information about current and historical jobs, including the resource usage and queuing status.
- Supports aggregating jobs from different dimensions, such as the quota group, submitter, and job status. This allows you to clearly understand the status of current jobs.
- Supports generating a detailed Logview page for a single job.
- Supports terminating jobs.

35.Dataphin

35.1. What is Dataphin?

35.1.1. About Dataphin

Dataphin is an intelligent engine for building big data platforms. It is designed to meet the requirements of big data development, management, and application across multiple industries.

Dataphin combines technologies and methodologies based on the OneData, OneEntity, and OneService systems that Alibaba Group has developed and applied for years. Dataphin provides all-in-one intelligent data development and management services, including data ingestion, data standardization, data modeling, data asset management, and data services. These features help government agencies and enterprises build an asset-oriented, service-oriented, closed-loop, and self-optimizing intelligent data system with unified standards to stimulate and drive innovation.

Dataphin applies to different computing and storage environments. This enables you to use a single console to process data from various data sources. Dataphin allows you to import data, standardize data production, develop data by data modeling, and create a tag system by extracting tags from entities. You can also generate and manage data assets by using your business data and knowledge. Dataphin also provides multiple types of data services such as table query and intelligent voice search.

35.1.2. Features

This topic describes the modules and features of Dataphin.

Platform

This module helps you obtain information about the entire system and global settings, and understand the system features to get started with ease. It also implements system management and control to ensure that all the other modules are running as expected.

Global design

You can design a data architecture based on a global view of your business and data. During the design, you can define namespaces, theme domains, and terms. You can also create projects as management units and add data sources and computing engines.

Data ingestion

Based on the projects and physical data sources defined during global design, the data ingestion module can extract data of various business systems and types and load it to the target big data storage. During this process, data is synchronized and integrated, and the source data layer is built based on the integrated data of vertical business. Data ingestion provides a solid foundation for further data processing.

Data standardization

Based on the architecture defined in global design and the source data layer built by data ingestion, you can create data elements such as statistical metrics. You can use these data elements to ensure that clear and standardized data will be produced.

Data modeling

You can use the data elements created for data standardization to design data models. After the data models are submitted and published, Dataphin automatically generates code and scheduling tasks to complete data production at the common dimensional model layer in a fully managed manner.

Coding

Dataphin provides a code editor for you to configure and submit code tasks.

Resource and function management

Dataphin allows you to manage resource packages such as JAR packages and other type of files to meet data processing requirements. You can search for and use built-in functions and create user-defined functions to meet specific requirements for functional processing.

Scheduling and management

Dataphin supports policy-based scheduling and management of tasks generated by modeling, coding, and data distilling. You can deploy, run, and manage data production tasks, and view and manage task dependencies. This module ensures that all tasks can run as expected without interruption.

Metadata center

Dataphin allows you to collect, parse, and manage metadata of the source data layer, common dimensional model layer, and data distilling center.

Data asset management

Based on the metadata center, this module supports deep metadata analysis and data asset management. It shows asset distribution and metadata details. This makes it easy for you to search for data assets and obtain information about data assets in more detail.

Ad hoc query

This module allows you to execute custom SQL statements to query data. It uses the search and analysis engine to find data in physical tables and theme-based logical tables. Theme-based logical tables are also known as data models or logical models.

35.1.3. Benefits

This topic describes the benefits of Dataphin.

Dataphin provides the following benefits.

- Standard data: The definitions of dimensions, dimension attributes, business processes, and metrics are standardized based on dimensional modeling. This ensures the quality of data and accuracy of metrics.
- Efficient and automatic coding: Dataphin defines logical components for common data computing based on functional programming. It allows you to customize statistical metrics. You can create data models as required. Then, Dataphin automatically generates code to produce data.
- Optimal intelligent computing: You can create logical models from business perspectives. After you publish your logical models, Dataphin automatically generates the physical representations and code of the logical models. This simplifies data modeling and coding.
- All-in-one development: Dataphin integrates data ingestion, data modeling, scheduling and management, data search, and data exploration to help you develop data in a centralized and efficient manner.

- Systematic data catalog: Based on standardized modeling and efficient and automatic metadata extraction, Dataphin provides a standardized and readable business data catalog. The data catalog forms a data asset map and allows you to spend less time finding and using the data you require.
- Efficient data search: An overview of data assets is provided based on your metadata and the data from the Dataphin system database. You can search for tables and query data in a fast and intelligent manner.
- Visualized data assets: A business data asset map is built to represent your business system from different data perspectives, extract business data knowledge, and obtain more information about key business stages and data.
- Easy and reliable data utilization: Data elements can be used for data production after they are created. You can search and access logical tables that are created based on business themes with ease. This simplifies about 80% of query code.
- High efficiency: Dataphin provides end-to-end and intelligent data development and management tools to improve the data development efficiency. Developers can independently run the ETL procedure to meet data requirements. The patent pending OneData, OneEntity, and OneService methodology allows you to abstract and customize models and metrics. Dataphin can also automatically generate code, aggregate data by theme, and provide data aggregation results.
- Low costs: Dataphin is based on metadata and driven by intelligent algorithms. Data can be automatically produced on the physical platform and logical plane in an intelligent manner. In addition to comprehensive analysis for data assets, Dataphin ensures the optimal allocation of computing and storage resources. This reduces the cost of data production and consumption.

35.2. Technical advantages

This topic describes the technical advantages of Dataphin.

- **Data standardization and automatic coding**
  - Data standardization: You can use functions to define dimensions, business processes, and metrics based on dimensional modeling. You can combine these computing logic components to create data models. Then, Dataphin automatically generates code to produce data.
  - Efficient and automatic coding: Dataphin optimizes the combination of computing logic components by physically or logically partitioning physical tables referenced by a logical model and generates code for producing data. In this way, optimal computing and storage performance can be achieved.
  - Automatic scheduling: By analyzing code and following scheduling configuration, Dataphin designs a scheduling directed acyclic graph (DAG) to run tasks in the optimal sequence. This ensures the optimal performance of data production.
  - Various models: Dataphin allows you to create logical tables in seconds. After you submit logical tables, Dataphin generates and optimizes code in minutes. Dataphin supports common, hierarchy, enumeration, and virtual dimension-based models, transaction and periodic snapshot fact-based models, and native atomic and composite metrics. Dataphin allows you to use business filters and metrics to create logical tables in the snowflake schema or star schema.
- **Systematic data directories**
  - Metadata extraction: Dataphin automatically extracts metadata from logical tables and code.
  - Metadata parsing: Dataphin automatically parses metadata to accumulate data assets based on the data asset model.
  - Data asset presentation: Dataphin displays the overview, flow, and structure of data assets so that you can obtain data in relevant business scenarios.

- **Easy and reliable use of data**
  - Logical table-based query: Dataphin allows you to query required data based on *[Logical model. Dimension-associated field. Dimension-associated field. ···. Attribute]*, such as Order. Buyer. Membership type. Type. This can shorten the length of SQL statements by 60%.
  - Code optimization: After you submit logical tables, Dataphin optimizes the code to achieve optimal computing performance. This improves the computing efficiency and saves resources if compared with physical table-based query.
- **All-in-one development**
  - Dataphin integrates data ingestion, data modeling, data development, scheduling and management, data search, and data exploration to help you develop data in a centralized and efficient way.
  - The Dataphin code editor provides syntax prompts and integrates with metadata for you to track table details with one click.
  - Up to 100 members can collaborate in Dataphin at the same time. You can grant permissions on fields and tables to members. Different roles have different permissions. Members can apply for permissions as required.
- **Efficient scheduling**: Dataphin can schedule millions of tasks at hourly intervals. After you configure resource settings for tasks, Dataphin can allocate resources to the tasks based on your settings.
- **Disparate data sources**: Dataphin allows you to read data from or write data to various heterogeneous data sources and provides features such as dirty data filtering and traffic shaping.
- **Multiple computing engines**: Dataphin supports two computing engines: MaxCompute and Hadoop 2.6.0-cdh5.11.2.
- **Deep data distilling**: Dataphin focuses on people-related IDs and tags. You can define objects and set computing parameters to identify IDs related to objects, map IDs of different types, and create tags for objects. This allows you to build a DMP for marketing with ease.

35.3. Product architecture

35.3.1. System architecture

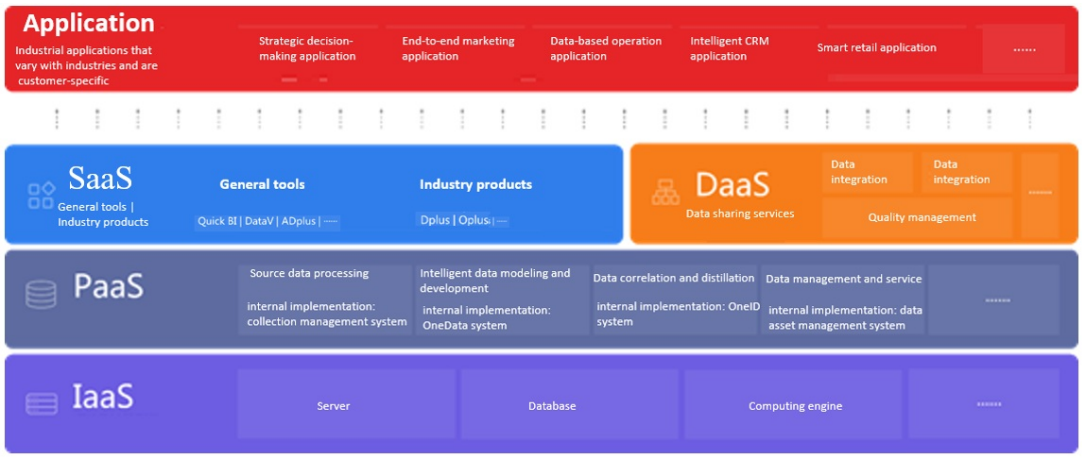
This topic describes the system architecture of Dataphin.

System architecture shows where Dataphin is deployed in a business system.

System architecture

As the Platform as a Service (PaaS) source data layer of the data platform for business systems, Dataphin can provide easy-to-use data services to support various data services at the upper layer. In this way, Dataphin helps you implement data-based business operations. Dataphin is compatible with different underlying hardware facilities and can connect to a wide range of applications. It functions as data pathway from Infrastructure as a Service (IaaS) to Software as a Service (SaaS) and can generate business-oriented standards, combined connections, and manageable and queryable data, as shown in Business system.

Business system

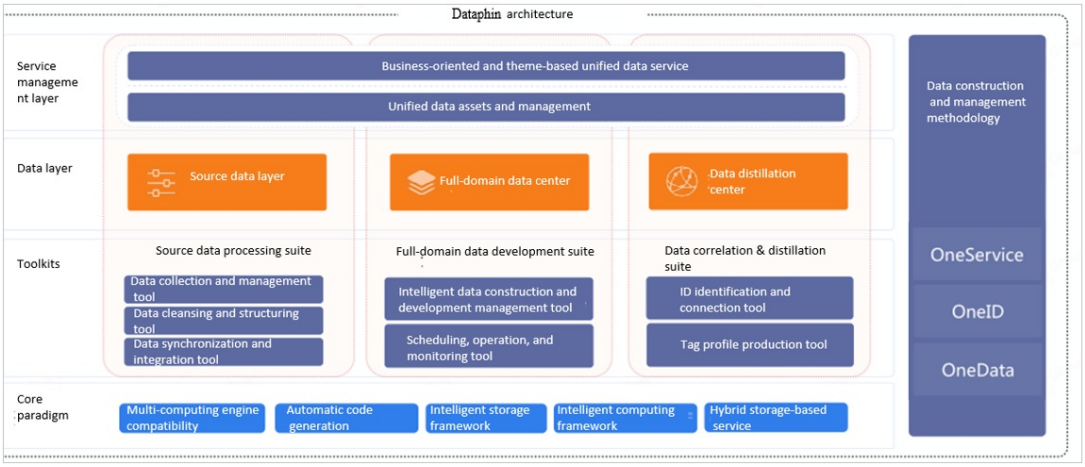


35.3.2. Technical architecture

This topic describes the technical architecture of Dataphin.

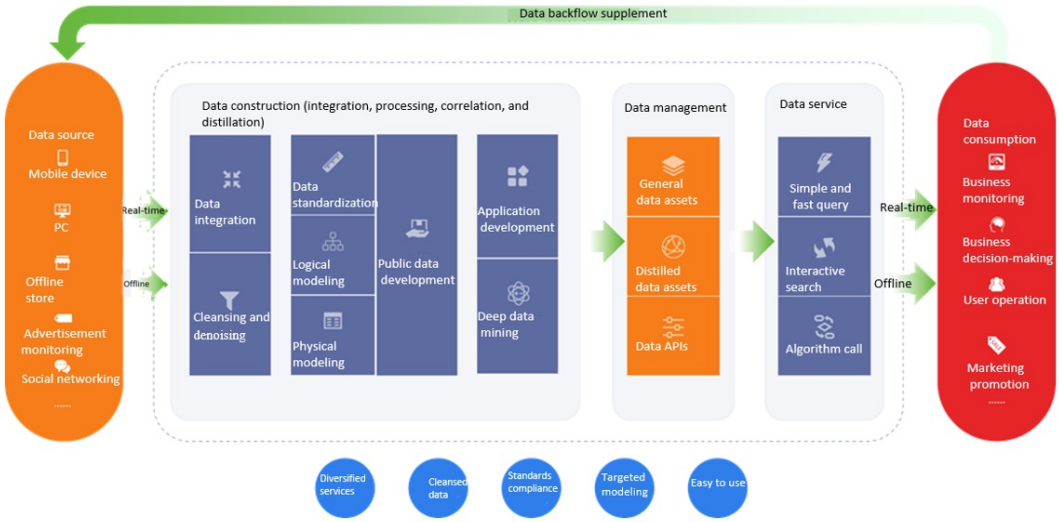
Technical architecture shows the technical architecture of Dataphin.

Technical architecture



Based on the data development and management methodology of OneData, OneID, and OneService, Dataphin consists of four layers: **core technology layer**, **tool layer**, **data layer**, and **management and service layer**. Collaboration between these four layers enables controllable data flows, as shown in Data flow.

Data flow



- **Core technology layer**: This layer masks the differences of underlying computing, storage, and software systems. It allows you to use multiple computing engines to complete data development more efficiently. It supports automatic coding, intelligent storage and computing, and hybrid storage.
- **Tool layer**: This layer provides data development and management tools for developers. Developers can use these tools to standardize, integrate, and ingest source data, standardize common data, build models, schedule and manage tasks, implement machine learning, identify and map object IDs, and define tags.
- **Data layer**: This layer uses core technologies and tools to generate three types of structured data: source data, common data, and distilled data. These data types help build a business-oriented source data layer that provides precise business data, a theme-based common data modeling layer that provides modeling data, and an entity-centered data distilling center that provides deeply processed behavior data and tags.
- **Management and service layer**: This layer provides insights into data and data service management, allowing both developers and business staff to obtain high-quality and unified data

assets. From a business perspective, the existing data is packaged and processed into theme-based data services to ensure that business data can be searched for and utilized in a unified manner.

35.4. Features

35.4.1. Console

As the basis of Dataphin, the Dataphin console guarantees that all Dataphin members can develop data in a controllable, orderly, and smooth manner. In this console, you can configure global settings, such as account management and computing management. The Dataphin console supports both Chinese and English, and provides introductions and entrances to various modules on its homepage. This helps the super administrator get the whole picture of Dataphin and members of other roles quickly access modules.

Account management

The Dataphin console allows you to manage member accounts to guarantee secure use of Dataphin. You can connect your enterprise account system to Dataphin. Then, the users who need to use Dataphin can be added to Dataphin as members. Users with the highest privileges can manage the accounts and permissions of other users.

Computing management

- As a Platform as a Service (PaaS), Dataphin enables you to select a computing engine type and configure connection settings for your data sources. This makes Dataphin compatible with various environments at the Infrastructure as a Service (IaaS) layer. In this way, Dataphin can develop and compute data in a uniform and stable manner.
- Dataphin supports two major types of computing engines: MaxCompute and Hadoop. Dataphin can automatically collect and parse the metadata of these two types of engines. For more information about how to collect and deploy metadata, see Metadata center.

Homepage

- The Dataphin console provides shortcuts to functional modules, projects, and the scheduling center on the homepage. You can also find an overview of the scheduling center and projects on the homepage.
- The homepage classifies modules based on the workflow in Dataphin, which consists of data warehouse planning, data R&D, data asset management, and theme-based data services. The workflow helps you learn about Dataphin features before you get started, and enables you to quickly access specific modules.

Language

To help users from different countries and regions use Dataphin, the Dataphin console selects Chinese or English based on the language of your operating system.

35.4.2. Global design

You can design a data architecture based on a global view of your business and data. This a fundamental step in data development. The architectural design ensures that data is manageable and controllable. The data systems defined and designed during data development, distilling, and management meet mid- and long-term business requirements. The produced business data is service-oriented, theme-based, and easy to use.

This module provides the following features:

- Data architecture definition based on business characteristics, including business unit management and access control, data domain management and access control, and management of the global objects that are defined
- Project definition based on requirements for independent data management and collaborative development, including member management and the management of basic project information and computing resources
- Data source configuration based on computing resources for projects and requirements for business data, including data source management

Data architecture

The data architecture defines logical namespaces, theme domains, and terms based on business characteristics. This standardizes data definitions during architectural design management and data development control.

Project

A project is a physical namespace used to isolate users from resources. Projects are created to meet the requirements for independent management of data development projects and efficient management of data resource quality. Data development constraints can be configured for each project.

Physical data source

Dataphin allows you to manage data sources, for example, add and modify data sources. You can also register and cancel the registration of databases. Dataphin supports various types of data sources such as MaxCompute, MySQL, SQL Server, and PostgreSQL. Data sources can be used as the source storage or destination storage for data synchronization. Some special types of data sources such as MaxCompute can serve as the computing engine for projects. Such a data source functions as both the computing and storage base of the target project.

35.4.3. Data ingestion

This topic describes the features provided by the data ingestion module.

Data ingestion helps you build the source data layer. Before data ingestion, you must select a business data storage system as the data source. Then, you must formulate data synchronization, cleansing, and structuring policies based on your data needs in terms of storage, timelines, and quality.

Data ingestion is the initial stage in data development. The data synchronization suite is developed based on years of practice of Alibaba in synchronization and exchange of various types of data, such as business data and log data. This helps achieve efficient ingestion of raw business data. Pipelines can be used to collect and analyze metadata and check the amount and content of data that has been transmitted. Simple rule-based check and flexible management of custom error tolerance mechanisms are supported. This helps achieve high-quality data synchronization.

Data source configuration

You can add and manage multiple data sources. The data source list allows you to manage added data sources and add diversified data sources. Dataphin supports the following types of data sources: MaxCompute, MySQL, SQL Server, PostgreSQL, and Hive.

Data synchronization

You can select the source and destination data sources for data synchronization, set parameters for incremental or full synchronization, and specify mappings between fields in the source and destination data sources. You can also configure the data transmission rate and the number of concurrent sync tasks. Dataphin generates and schedules synchronization tasks based on your settings.

35.4.4. Data standardization

This topic describes how to standardize dimensions, business processes, atomic metrics, business filters, and derived metrics.

Overview

During traditional development, specific and important data construction and development, such as data modeling and metric definition, depend on the professional capabilities of developers in most cases. Without a uniform naming convention, standards for development and design are transferred based on individual and changing documents. This may cause a series of issues such as metric name conflicts or repeated computing.

Based on the OneData methodology, Dataphin standardizes the definition of important data elements such as dimensions, business processes, and metrics. This ensures unique computing logic and names, and eliminates metric ambiguities during the initial stages of architectural design. In addition, Dataphin provides form-based interfaces for you to create multiple metrics at a time. This lowers the requirements of data development and increases overall development efficiency. This also allows business staff with limited data analysis expertise to carry out development work by using Dataphin.

The data items to be defined include dimensions, business processes, atomic metrics, business filters, and derived metrics.

Dimension

- A dimension is unique within a business unit and exclusively belongs to a data domain. This standardizes naming and theme classification.
- You can create dimensions by adding additional attributes to an existing dimension, which is used as a parent dimension.
- Dataphin allows you to create various types of dimensions, including common dimensions, common dimensions by hierarchy, enumeration dimensions, and virtual dimensions.
- Dataphin allows you to view and manage the list of dimensions created in a specific business unit or a specific project. You can also view and modify each dimension.

Business process

A business process is a collection of the smallest unit of behaviors or events that occur in a business activity. For example, creating an order or browsing a web page can be the smallest unit of behavior. The behaviors occurring in a business process, such as paying for an order and browsing a web page, are recorded in a fact table. In most cases, a fact table models a particular business process.

Similar to dimension, business process is a key concept in the OneData methodology that is used to design the data architecture. To design a data architecture, you must define both business processes and dimensions. Dataphin allows you to define standards for business processes. You can check the overall business data of your organization and classify fact tables by business process.

To ensure that a fact-based model is built in a unified and standard manner, each business process is unique within a business unit and exclusively belongs to a data domain. This standardizes naming and theme classification.

Dataphin allows you to view and manage the list of business processes created in a specific business unit or a specific project. You can also view and modify each business process.

Atomic metric

An atomic metric is an abstraction of computing logic. To eliminate definition and development inconsistency, Dataphin introduces the concept of **Design to Code**. When a metric is defined, the statistical criteria (computing logic) is also defined. Re-engineering of the ETL process is not required, which increases development efficiency and ensures the consistency of statistical results.

Based on the complexity of computing logic, Dataphin classifies atomic metrics into the following types:

- Native atomic metrics, for example, payment amount
- Composite metrics: Composite metrics are created based on the combination of atomic metrics. For example, the average sales per customer is calculated by dividing the total sales by the number of customers.

Each atomic metric is unique within a business unit and has only one source logical table. The computing logic of an atomic metric is defined based on the fields of the source logical table model. This ensures that all statistical metrics are created in a unified and standard manner. The data domain of each logical table that is related to the source logical table model is retrieved to trace the data domains to which the atomic metric belongs. An atomic metric may belong to multiple data domains. This ensures that names and logic are normalized and themes are classified in a standard manner.

Business filter

An atomic metric is the standardized definition of computing logic, whereas a business filter is the standardized definition of a query condition. Similar to an atomic metric, a business filter is unique within a business unit and belongs to only one source logical table. The computing logic of a business filter is defined based on the fields of the source logical table model. This ensures that all metrics are created in a unified and standard manner. The data domain of each logical table that is related to the source logical table model is retrieved to trace the data domains to which the business filter belongs. A business filter may belong to multiple data domains. This ensures that names and logic are normalized and themes are classified in a standard manner.

Derived metric

Derived metrics are commonly used statistical metrics. To ensure that statistical metrics are generated in a standard and unambiguous manner, the OneData methodology defines each derived metric based on the following elements:

- Atomic metric: the statistical criteria, which is the computing logic.
- Business filter: the scope of business to be measured. It is used to filter the records that comply with specific business rules.
- Statistical period: the time range during which statistics are collected, for example, the last day or

the last 30 days.

- Statistic granularity: the statistical object or perspective that defines the level of data aggregation. It can be considered as a grouping condition for aggregation. A statistic granularity is similar to the object specified by the GROUP BY clause in SQL statements. A statistic granularity is a combination of dimensions. For example, if a derived metric measures the turnover of a seller in a province, the statistic granularity is the combination of the seller and region dimensions.

Based on the combination of the preceding elements, Dataphin allows you to create multiple derived metrics and ensures clear and non-duplicate definitions and computing logic. You can also customize derived metrics as required. A derived metric is a concept at the same level as a field. Each derived metric is unique and defined at the specified granularity level.

35.4.5. Modeling

Dataphin provides systematic modeling and development functions to deeply implement the data warehouse theory. You can create business dimensions and business processes by using a top-down approach, and then enrich dimension tables, fact tables, aggregate tables, and the application data store layer. This process allows you to produce standardized data assets, which provides you with layered business data. The data standardization process can also optimize computation and storage.

Logical dimension tables

A logical dimension table contains details about a dimension. Dataphin allows you to view and manage the list of created logical dimension tables, and to view and modify a specific logical dimension table.

Logical fact tables

Dataphin supports using logical fact tables to model a specific business process (such as placing an order and paying for a commodity) or a state measure (such as account balance and inventory). A logical fact table is created in an optimized schema that is similar to a snowflake schema. Apart from measures and dimension-associated fields, this type of schema allows a fact table to also contain fact attributes. This reduces the complexity of the model design and makes it more user-friendly.

Logical aggregate tables

The logical aggregate table model is an important data warehouse model. It contains two types of elements. The first type of element refers to various statistical values used to describe statistic granularity. The statistical values form a derived metric, for example, the sales in the last seven days. Granularity is a combination of several dimensions, such as the province and the product line dimensions. The second type of element refers to the attributes of the dimensions that constitute granularity. Examples of attributes are province name, product line name, product line level.

Coding automation

After a logical dimension table, logical fact table, or logical aggregate table is published, Dataphin automatically designs the corresponding physical model, generates code and tasks to produce required data. Multiple tasks are usually generated to convert a logical table to a physical model. If you want to view the task running logic, go to the Scheduling page.

35.4.6. Coding

Coding is an important data development method that can be used to achieve the same goal as data modeling. Dataphin allows you to edit scripts by using the coding method supported by your computing engine. You can submit the scripts to the scheduling system, which then schedules the code tasks to produce data. You can also view historical versions of each code task.

Dataphin supports multiple types of scripts, such as SQL, Shell, and MapReduce scripts. Different types of scripts have different coding and configuration requirements. The requirements include syntax requirements and requirements for configuring scheduling policies. After you submit and publish a script, Dataphin generates a code task and runs the task to produce data. In a DAG, a task is also called a node. Dataphin allows you to manage code tasks. You can create, view, modify, and delete code tasks, edit scripts, configure task scheduling policies, publish tasks, and manage task versions.

Code editor

The code editor provides an online code editing interface for you to complete data development. It supports SQL, MapReduce, Spark, and Shell programming.

Scheduling policy configuration and task publishing

- Configure scheduling policies

  You can configure scheduling policies for one-time and recurring tasks. You can publish tasks after you configure scheduling policies for them. The system can check the integrity of task scheduling policies. Only tasks with a complete scheduling policy can be published. All published tasks are recurring tasks. You can go to the **Scheduling** page and click **Recurring Tasks** in the left-side navigation sub-menu to view the published recurring tasks.

- Publish tasks

  Members of a project can submit and publish tasks if they have required permissions. Only tasks with complete scheduling policy configurations, including parameter settings, valid dependencies, and no circular dependencies, can be submitted and published for scheduling. This ensures stable and orderly data production on schedule.

Code management

Dataphin supports various code operations to facilitate code file management and use. You can create, delete, update, rename, and view code files, and place code files in specific folders to categorize the code files.

- Manage files

  Dataphin allows you to edit, delete, unpublish, and rename each code file. You can also view the publishing status, creator, and creation time of each code file. This facilitates easy creation, clear display, and systematic management of code files.

- Manage folders

  Dataphin allows you to sort code files in different folders. It can save and display code files in an orderly manner. You can create, rename, and delete folders, and move historical and new code files to specified folders for better management. Dataphin also supports hierarchical folder structures.

Collaborative programming

- Manage node versions

Dataphin allows you to view historical versions of task nodes. You can view the version number, submitter, submission time, and description of each version. You can also view the code of each version to identify differences in code. Dataphin supports multiple node types, including MaxCompute_SQL, MaxCompute MR, and Shell.

- Develop scripts collaboratively

  To achieve more efficient development by allowing collaboration between multiple developers, Dataphin provides a script locking mechanism, which prevents conflicts during collaborative development. This mechanism ensures that a script can be edited by only one user at a time. A user can steal the lock of another user to obtain the script editing permission. The user whose lock is stolen can obtain editing permission again by stealing the lock.

35.4.7. Resource and function management

Resource and function management assists code development. Data developers can upload local resources and configure task nodes to call these resources, to meet specific data processing requirements. Developers can also complete common data processing by using the built-in functions in the programming language supported by the computing engine. If specific data logic, such as data conversion in compliance with specific business logic, needs to be processed at a high frequency and this cannot be achieved with the built-in functions of the system, developers can define custom functions based on the resources that they have uploaded.

Resource management

As a data developer of a project, you can manage resources in the project. For example, you can add and modify resources in the project. You can upload and name resource files, and copy the resource file names to reference the resource files in code. You can also delete unnecessary resource files.

- Create and upload resource files

  By default, you can upload the following types of local resource files: XLS, DOC, TXT, CSV, JAR, Python, and other types such as ZIP packages. New file types that are different from these types can be added in three days by using the standard interface. Each resource file name is unique within a project. After you submit a resource file, the file name and resource package cannot be changed. You can upload only one resource file each time. The type of the file to upload must be the same as the file type you selected.

- Reference resources

  You can copy and paste a resource file name to a specific position in the code editor, and write a statement to call this resource.

- Update resources

  You can update the description of managed resources and delete existing resources to save storage space.

Function management

You can search for, use, and manage functions. Functions are classified into two types: built-in functions of the system and user-defined functions based on uploaded resources such as JAR and Python packages. You can extend user-defined functions by referencing standard functions.

- Create user-defined functions

  Each user-defined function must have a unique name in a project. After a user-defined function is registered, you cannot change its name.

- Reference functions

  You can copy the name of a built-in function or a user-defined function, and then paste the name to a specific position in the code editor. Then, you can write a statement to call the function for data processing.

- Update functions

  You can update information except the name of a user-defined function and delete unnecessary user-defined functions.

35.4.8. Scheduling and management

This topic describes the task and instance lists in the scheduling center and the management operations that you can perform on the tasks and instances.

The scheduling center allows you to perform management operations during the later stages of data development. It provides the list of all data processing tasks and task instances. Data processing tasks include recurring and one-time tasks. Task instances include instances of the data processing tasks and retroactive data generation tasks. The scheduling center also provides the DAGs that show task dependencies, task instance dependencies, and instance status. You can set the task running sequence, schedule specific nodes in a DAG, optimize resource allocation, and discover abnormal tasks. This ensures that all the tasks are stably and reliably run on schedule. The scheduling center also reports alerts during task running to ensure that errors can be handled in time. The scheduling center allows you to view and manage tasks and task instances.

Task list

You can view the lists of recurring and one-time tasks that are created in a specific project and the DAGs that show task dependencies.

Recurring tasks

You can view the list of recurring tasks, search for specific tasks, and view the dependencies of each task. You can switch between different projects to view and search for tasks in a specific project. You can search for tasks by task name or task ID in fuzzy match mode. You can also filter the tasks that you own and the tasks that are published on the current day. This helps narrow down the scope of tasks or find specific tasks that you want to manage.

One-time tasks

You can view the list of one-time tasks, search for specific tasks, and view details of each task. You can switch between different projects to view and search for tasks in a specific project. You can search for tasks by task name or task ID in fuzzy match mode. You can also filter the tasks that you own and the tasks that are published on the current day. This helps narrow down the scope of tasks or find specific tasks that you want to manage.

Task instance list

You can view the lists of recurring, one-time, and retroactive data generation task instances that are created in a specific project and view the details of each task instance.

Recurring task instances

You can view the list of recurring task instances, search for specific instances, and view the details of each instance. You can view the running status and details of each recurring task instance. The details include the task ID, task name, task owner, task start time, end time, and running duration. You can switch between different projects to view and search for task instances in a specific project. You can search for task instances by task name or task ID in fuzzy match mode. You can also filter the task instances that you own, instances with errors, and incomplete instances. This helps narrow down the scope of instances or find specific instances that you want to manage.

One-time task instances

You can view the list of one-time task instances, search for specific instances, and view the details of each instance. You can view the running status and details of each one-time task instance. The details include the task ID, task name, task owner, task start time, end time, and running duration. You can switch between different projects to view and search for task instances in a specific project. You can search for task instances by task name or task ID in fuzzy match mode. You can also filter the task instances that you own and the instances that are published on the current day. This helps narrow down the scope of instances or find specific instances that you want to manage.

Retroactive data generation task instances

You can view the list of retroactive data generation task instances and details of each instance. The details include the data timestamp, status, and running duration. You can also view the ID, name, and owner of the task for which retroactive data is generated. You can also search for and filter retroactive data generation task instances.

Logical tables

You can search for and view logical tables and their conversion tasks. You can also view the details of each logical table. You can switch between a logical table task and a logical table task instance to view their details. By default, the DAG on the right of the logical table task list shows all nodes of the current logical table and the dependencies between the nodes, including indirect dependencies. By default, the DAG on the right of the logical table task instance list shows all task instances of the current logical table and their status. The status may be running, success, or failed.

35.4.9. Metadata center

Dataphin provides powerful metadata management capabilities. It can collect and extract metadata from MaxCompute, Hadoop, Hive, MySQL, PostgreSQL, and Oracle data sources. It supports real-time tracing of metadata in the preceding computing and storage engines, and can build a unified metadata model by extracting metadata from different types of storage engines.

Dataphin supports rapid enrichment of multiple types of metadata and provides diverse metadata that complies with unified standards. In this way, Dataphin can provide powerful and stable metadata support for data maps and data governance.

The metadata center is the core foundation of data asset management. We recommend that you consider the following points when you build the metadata center:

- Metadata collection standard: A unified data development standard is required to ensure the consistency of metadata for modeling, data table creation, and data lineage. This improves the availability of metadata for data retrieval and data services.
- Metadata timeliness and quality: The metadata output time and quality must be ensured to improve the accuracy of the data in the data asset module and the efficiency of data retrieval performed by developers.

- Metadata model system: A unified public metadata model is used to ensure compatibility with various types of data and deliver a comprehensive data map service.

35.4.10. Data asset management

After development works such as data collection, integration, and processing are complete, you can manage data assets in a systematic way.

Based on OneData and data asset management methodologies, Dataphin designs data application principles and provides core technologies, including metadata collection, extraction, and processing technologies. You can classify and manage data in the form of assets, monitor data quality, and optimize resources. This allows you to minimize the cost of data, maximize the value of data, and apply this value to benefit your business.

Data assets are managed by using a series of core technologies. The real-time event and subscription services enable real-time updates of metadata for tables and tasks. The rules engine ensures efficient and accurate judgment of data governance rules and creation of health scoring models. Dynamic log analysis analyzes numerous production task execution logs and machine operations logs every day. Graph computing supports the analysis and establishment of data lineages. Onelog tracing analysis interlinks end-to-end metadata during data production, service, and consumption. The plug-in metadata access and processing architecture ensures compatibility with multiple computing and storage engines. Data asset management uses a methodology that integrates a set of procedures, such as data analysis, governance, application, and operation. Alibaba Group developed this methodology based on its extensive experience in managing large amounts of data. Data asset management covers the entire data lifecycle, including data creation, management, application, and destruction.

Data asset management involves two keywords: universal and fusion. Universal refers to a process of checking all data and creating a data asset dashboard based on factors such as dimensions, business processes, and correlations in the OneData system. This process describes data assets by using modeling. Fusion is a process of analyzing the cost and value of data assets during production. This process describes the functions of different datasets on the data asset dashboard based on the connection and contribution models.

Based on the data asset catalog that is established based on an analysis of enterprise data assets, the data map module provides a search engine and metadata profiling, both of which are derived from user behavioral data. This allows you to retrieve the data assets of an enterprise with improved efficiency.

Asset overview

Dataphin can display the structure of enterprise data assets that are created based on OneData. Components in different shapes represent business entities, whereas lines of different styles represent relationships between these entities. This helps visualize the structure of the data for a business unit.

Asset map

An asset map summarizes the relationships between dimensions and business processes in a data domain of a business unit to show the composition of your enterprise data. In addition, the asset map provides efficient, fast, and accurate data search and exploration based on your self-initiated behaviors, such as searching for, accessing, and bookmarking data assets.

35.4.11. Security management

Dataphin focuses on intelligent development and management of data and attaches great importance to data security management. It provides comprehensive data security protection throughout the entire lifecycle from data production to destruction. The protection measures include data access control, data isolation, data classification by security level, privacy compliance, data de-identification, and data security auditing.

The wide use of big data services makes data security an important issue. In China, the Cyber Security Law of the People's Republic of China took effect on June 1, 2017. It encourages the development of network data protection and application technologies. EU General Data Protection Regulation (GDPR) was enacted on May 25, 2018. It aims to enhance the protection of data such as personal information.

Data access control and data isolation require the highest priority in data security management. Dataphin supports management of data access permission requests, approvals, and lifecycle. It can isolate data by tenant and control data access by field. It offers a data access authorization model based on access control lists (ACLs).

Dataphin establishes a comprehensive data security guarantee system that ensures data security in the entire lifecycle of data. This system provides technologies and management measures to protect data from the perspectives of data access behaviors, data content, and data environment. During big data development and management, Dataphin works with the Alibaba Cloud data security management system to provide an available but invisible environment for secure big data exchange. In addition, Dataphin can control data access by field, control permission request and approval processes, and trace and audit data use behaviors. All these features help ensure data security during the storage, transmission, and use of big data.

Dataphin offers a hierarchical permission control system and a full range of management, covering the request, approval, assignment, handover, and authentication of data access permissions.

Permission types

Dataphin provides data access control based on user roles and resources. This allows you to use Dataphin and access data in a secure and controllable manner.

- Role permissions

  To manage user operations on the platform in a centralized manner, Dataphin provides an account management mechanism that controls access of the super administrator and system members on the platform. Dataphin can also control user access to resources in specific projects based on roles. It can grant a set of data resource permissions to specific roles. Users with specific roles have the permissions that are granted to the roles.

- Resource permissions

  Dataphin provides a data access control mechanism to centrally manage user operations on project data resources. When each project is independently managed, and system members are logically isolated from resources, Dataphin can control cross-project resource access. This helps achieve data sharing by allowing users in one project to use data of another project without data migration.

Permission management

- Permission requests

  Data developers can find the required data table on the data map and view the metadata details of this table. However, if they want to query data in the table, they must apply for permissions.

  In a permission request process, Dataphin displays information about the target data table by default, including the table type and the business unit to which the table belongs. Field metadata of the table is also displayed.

Dataphin supports permission requests that follow the principle of least privilege.

- You can request permissions on specific fields.
- Dataphin provides multiple options of permission validity period. You can customize a date range or select 30 days, 90 days, 180 days, or one year as the validity period.
- You can describe the purposes for which you intend to use the requested permissions. The approver can determine whether to grant you the permissions based on the description.

- Request management

  Dataphin allows you to view your permission requests and the status of each request on the **Initiated Tasks** page in **Task Center**. You can also view the list of approved permissions and the fields in detail. After your request is approved, you can view your permission details, including the accessible fields.

- Permission approval

  After you submit a permission request, the system randomly assigns the ticket to an administrator of the project to which the target data table belongs. The administrator is the approver of your permission request. The approver can view the details of your request on the **Unprocessed Tasks** tab and decide whether to approve or reject the request.

- Permission handover

  Users must hand over their permissions before they shift to another position or leave the company. This ensures that related data and data production tasks can be handed over to appropriate staff. On the **Unprocessed Tasks** page, you can click **Remove Permissions** to hand over your permissions to the project administrator. Then, Dataphin reclaims the permission.

35.4.12. Ad hoc query

Dataphin supports high-performance ad hoc queries based on the OneService engine. You can use both traditional simple query and theme-based query methods to achieve fast query with simple code.

Syntax

- Dataphin supports offline queries on all modeled logical tables. The intelligent query engine selects the optimal physical table based on factors such as the output time and query performance.
- Dataphin supports join queries based on snowflake schemas. This makes it simpler to write SQL queries.
- Dataphin supports queries on physical tables, logical tables, and combinations of physical tables and logical tables.
- Dataphin supports the syntax of multiple computing engines, such as MaxCompute SQL and Hive SQL.
- Dataphin provides intelligent code completion, precompilation, and beautification for SQL statements.
- Dataphin can manage permissions and authenticate users for access to fields in a logical or physical table.

Query execution

You can enter query statements in a query script as required. The script editor provides intelligent prompts based on the input content, locates the required data table or field, and verifies the validity of the script syntax.

36.Elasticsearch (on ECS)

36.1. What is Apsara Stack Elasticsearch?

Elasticsearch is a Lucene-based data search service. It provides a distributed, multi-tenancy, full-text search engine that uses a RESTful web interface. Elasticsearch, developed in Java, is a popular search engine for enterprises. Elasticsearch is designed to serve cloud computing for real-time search. It is stable, reliable, fast, and easy to install and use.

Apsara Stack Elasticsearch is released in 5.5.3 and 6.3.2 versions. Apsara Stack Elasticsearch is ideal for data analytics and searches. Based on open source Elasticsearch, Apsara Stack Elasticsearch provides enterprise-grade access control. Each Apsara Stack Elasticsearch cluster is composed of multiple nodes. The management nodes of Apsara Stack Elasticsearch support high availability. Management node failures do not affect the running of Apsara Stack Elasticsearch. In addition, Apsara Stack Elasticsearch can store audit logs and automatically dump them to the specified directory on a specific node for long-term storage and management.

Built-in plug-ins provided by Apsara Stack Elasticsearch include but are not limited to:

- **IK analyzer**: an open source, lightweight Chinese tokenizer kit developed in Java. It is a popular plug-in for Chinese tokenization in open source communities.
- **Smart Chinese analysis plug-in**: the default Lucene Chinese tokenizer. It enables an Apsara Stack Elasticsearch cluster to be automatically scaled to hundreds of nodes and process petabytes of structured or unstructured data.
- **ICU analysis plug-in**: a Lucene ICU tokenizer. ICU is a set of stable, tested, powerful, easy-to-use libraries and provides Unicode support for applications on different platforms.
- **Japanese (Kuromoji) analysis plug-in**: a Japanese tokenizer.
- **Stempel (Polish) analysis plug-in**: a French tokenizer.
- **Mapper attachments type plug-in**: an attachment plug-in that can use Apache Tika to parse content in different types of files into strings.

Core features

Apsara Stack Elasticsearch provides an easy-to-use data analytics service.

- High availability: Apsara Stack Elasticsearch provides high availability based on the underlying Alibaba Cloud IaaS architecture.
- Cost-effectiveness: The TCO of using Apsara Stack Elasticsearch comes from the fees of 100 virtual machines with medium configuration. The cost is 25% lower than that of open source Elasticsearch.
- Enterprise edition: Apsara Stack Elasticsearch supports access control, security monitoring and alerting, and auto scaling.
- Cloud product ecosystem: Apsara Stack Elasticsearch allows you to analyze and index data without the need to migrate data.

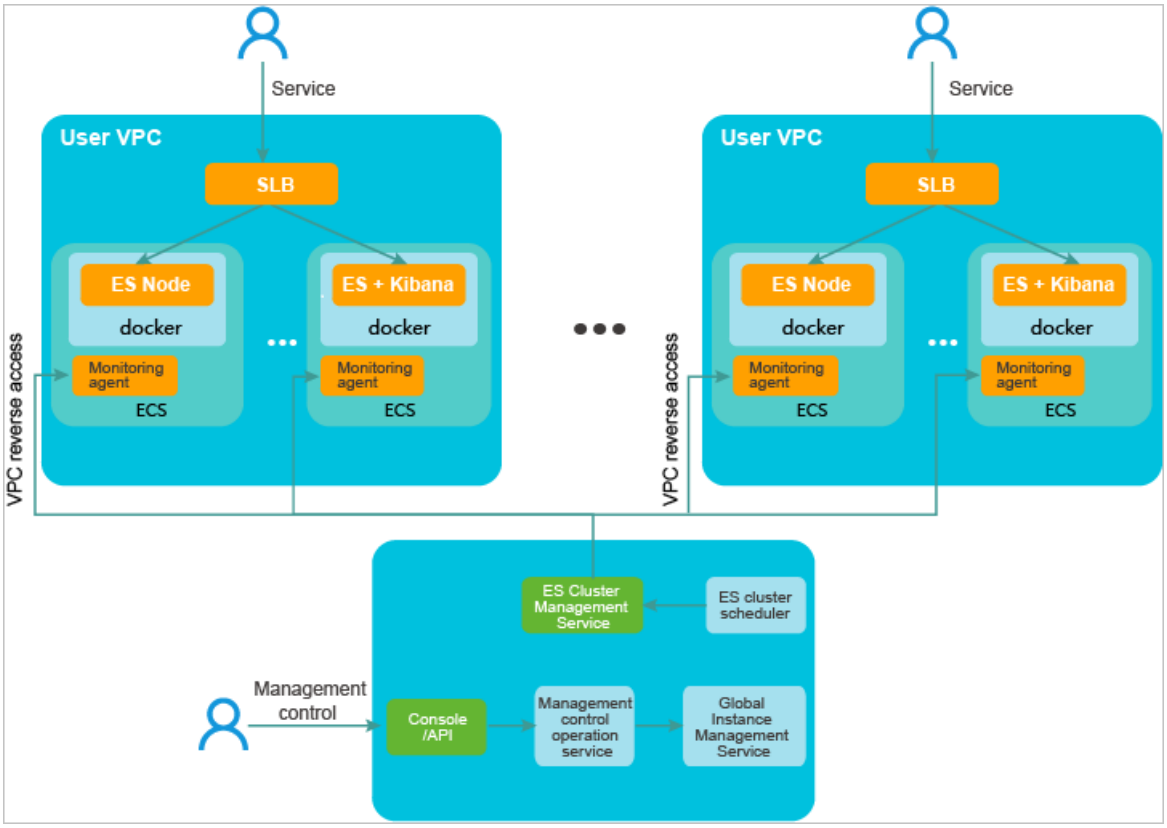
36.2. Benefits

This topic describes the benefits provided by Apsara Stack Elasticsearch.

36.3. Architecture

This topic describes the architecture of Apsara Stack Elasticsearch.

The following figure shows the architecture. In this figure, the procedure of creating an Apsara Stack Elasticsearch cluster is used as an example.



You can submit the configuration of the Elasticsearch cluster that you want to create from the Apsara Stack Cloud Management (ASCM) console or by calling the Elasticsearch API.

1. Select an Elastic Compute Service (ECS) instance. This ECS instance is used as an Elasticsearch node and provides storage space.
2. The governance service retrieves the instance and storage space information from ECS, saves your request to the database, and then submits the request to the global instance management service.
3. The global instance management service creates a configuration file for the Elasticsearch cluster based on the request type and submits the file to the Elasticsearch cluster management service.
4. The Elasticsearch cluster management service is an offline processing system that runs a task state machine based on the request type. The task state machine runs until the task reaches its desired state.

   When you create an ECS instance, the Elasticsearch cluster management service labels the ECS instance, connects it to a Virtual Private Cloud (VPC), and configures load balancing. Then, the service designates the cluster scheduler to manage the ECS instance. The cluster scheduler creates Elasticsearch and Kibana processes on the ECS instance.

   The Elasticsearch and Kibana processes run in containers on the ECS instance. The monitor agent, an independent process, collects monitoring metrics and sends them to Cloud Monitor by using Log Service. Elasticsearch clusters are isolated by VPCs. The governance service uses port mapping to establish reverse connections to your clusters for cluster management.

36.4. Features

This topic describes the features of Apsara Stack Elasticsearch.

Feature	Description
Kibana console	The Kibana console is a part of the Elastic ecosystem and is seamlessly integrated into Elasticsearch. The Kibana console allows you to monitor the status of your Elasticsearch cluster and manage the cluster. You can obtain the URL of the Kibana console for an Elasticsearch cluster from the Basic Information page of the cluster. Then, you can log on to the Kibana console from a server that resides in the same Virtual Private Cloud (VPC) as the cluster.
Cluster restart	This feature allows you to perform a restart or forced restart for your Elasticsearch cluster.
Cluster information refresh	This feature allows you to manually refresh the information of your Elasticsearch cluster. For example, if the Elasticsearch console fails to display the status of a newly created cluster, you can use this feature to update the status.
Basic Information page	On this page, you can view the basic and configuration information of an Elasticsearch cluster. The information includes the internal endpoint, URL of the Kibana console, and status.
Cluster configuration upgrade	You can upgrade the configuration of an Elasticsearch cluster from the perspectives of node specifications, storage space per node, and the number of nodes. You cannot downgrade the configuration of an Elasticsearch cluster.
Cluster configuration	This feature allows you to configure: • Synonyms You can upload a synonym dictionary file and use this file to configure synonyms. You can also reference synonyms. • YML file You can configure the Auto Indexing and Index Deletion features in the Elasticsearch console. You can also configure cross-origin resource sharing (CORS) and a remote index whitelist in the YML file.
Plug-in configuration	Elasticsearch provides built-in plug-ins and allows you to upload custom plug-ins. The IK analyzer plug-in supports two update modes for IK dictionaries: standard update and rolling update.
Security configuration	This feature allows you to reset the password, modify the Kibana whitelist, and modify the VPC whitelist for an Elasticsearch cluster.
Data backup	Elasticsearch provides the Auto Snapshot feature. This feature allows you to customize the backup time. You can view the statuses of snapshots and data in these snapshots. You can also restore data to the original Elasticsearch cluster from the snapshots.

37.Elasticsearch (on k8s)

37.1. What is Apsara Stack Elasticsearch?

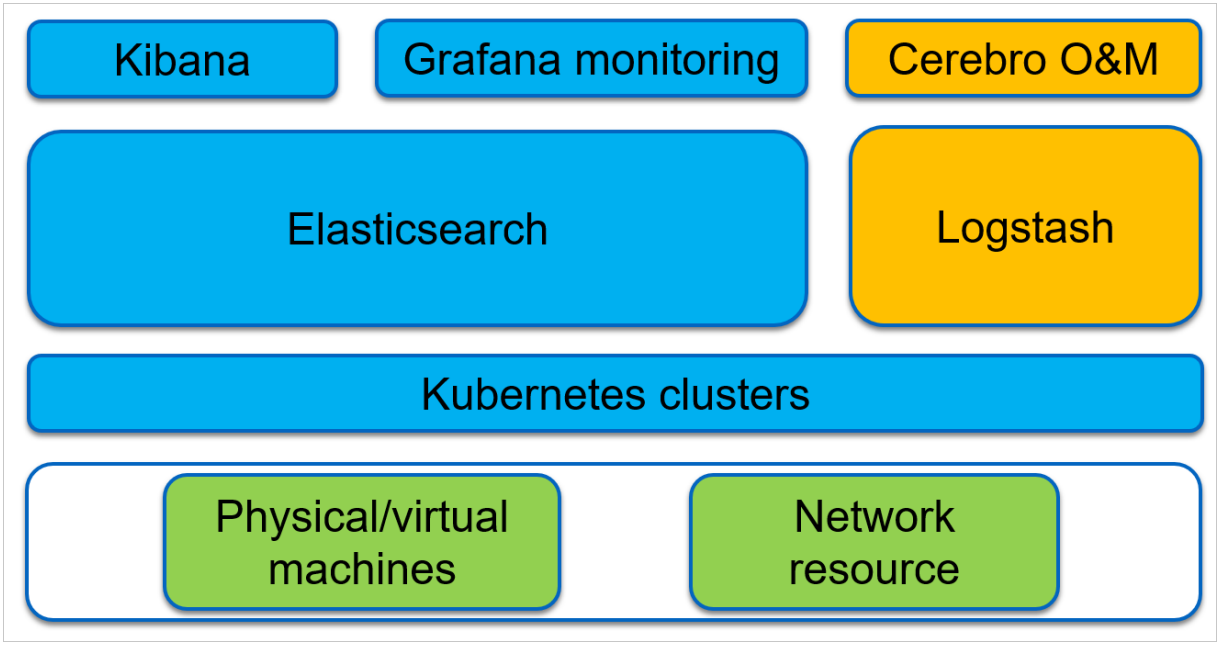
Open source Elasticsearch is a Lucene-based, distributed, real-time search and analytics engine. It is a product released under the Apache License. Elasticsearch is a popular search engine for enterprises. It provides distributed services, allowing you to store, query, and analyze large amounts of datasets in near real time. Elasticsearch is typically used as a basic engine or technology to support complex queries and high-performance applications.

Apsara Stack Elasticsearch provides fully-managed Elasticsearch services. It supports multiple versions of open source Elasticsearch and is compatible with all open source Elasticsearch features. Apsara Stack Elasticsearch offers an optimized kernel and provides the multi-tenancy, high availability, and auto scaling features. In addition to the features of open source Elasticsearch, Apsara Stack Elasticsearch allows you to create a cluster in a visualized manner, use Migration Assistant to migrate data, manage repositories, create snapshots, manage plug-ins, and perform O&M operations.

37.2. Architecture

This topic describes the architecture of Apsara Stack Elasticsearch.

The following figure shows the architecture.



Apsara Stack Elasticsearch is deployed on Kubernetes clusters. Kubernetes clusters can be deployed on physical or virtual machines. Then, you can create an Elasticsearch cluster, activate Kibana, and enable Grafana-based monitoring with one click on the operations and maintenance (O&M) platform. You can also activate Logstash on a Kubernetes cluster to import data and use Cerebro to perform O&M operations on Elasticsearch.

37.3. Features

This topic describes the features of Apsara Stack Elasticsearch.

- Base container: Kubernetes.
- Server type: physical or virtual machine. Physical machine models are not limited.
- Number of servers: 3 to 96.
- Elasticsearch version: all versions. Elasticsearch 6.8.2 and 7.2.1 are recommended.
- CPU and memory specifications: all specifications. 4 CPUs and 16 GiB of memory, 8 CPUs and 32 GiB of memory, and 16 CPUs and 64 GiB of memory are recommended.
- Storage type: Physical machines use local disks and do not support auto scaling of disks. One machine uses one disk.
- Resource isolation: Docker.
- Elasticsearch cluster management: You can create and maintain clusters in the Apsara Stack Operations (ASO) console. You can manage clusters in the Apsara Stack Cloud Management (ASCM) console. Elasticsearch supports multi-tenancy.
- Auto scaling: supported.
- Visualized management tools: Kibana and Grafana.
- Dependent services: Kubernetes and ASO.

37.4. Management features

This topic describes the management features of Apsara Stack Elasticsearch.

Target user	Feature
Tenant	View the details of an Elasticsearch cluster Allows you to view the basic information of a cluster, such as the cluster status, endpoint, and port number.
	Use the Kibana console Allows you to log on to the Kibana console. Allows you to scale your businesses. The Kibana console is seamlessly integrated into Elasticsearch. It allows you to view the status of your Elasticsearch cluster in real time and manage the cluster.
	Create Elasticsearch clusters Allows you to create clusters in the Apsara Stack Operations (ASO) console. The created clusters can be synchronized to the Apsara Stack Cloud Management (ASCM) console.
	Enable automatic snapshot creation Allows you to customize the snapshot creation time. After snapshots are created, you can view the basic information of the snapshots.

Target user	Feature
O&M personnel	Manage plug-ins Elasticsearch provides built-in plug-ins for Elasticsearch and Kibana. You can manually install or remove the plug-ins. You can also perform standard or rolling updates for IK dictionaries.
	Perform O&M operations on Elasticsearch clusters Allows you to perform health diagnostics, reset passwords, and configure specifications, zones, and YML files.
	Migrate data Allows you to migrate data between Elasticsearch clusters with one click. The data migration process is visualized.
	Create Logstash clusters Allows you to deploy Logstash clusters in a Kubernetes cluster in the Apsara Bigdata Manager (ABM) console. The created Logstash clusters are synchronized to the O&M platform of Elasticsearch clusters that are deployed in the same Kubernetes cluster as the Logstash clusters. Then, you can manage the Logstash clusters.
	Manage Logstash clusters Allows you to manage pipelines, file groups, and plug-ins and enable auto scaling for Logstash clusters.
	Platform management Allows you to manage OSS repositories and update or delete Elasticsearch, Logstash, and Kibana plug-ins.