Alibaba Cloud
Apsara Stack Agility SE

Product Introduction

Version: 1912, Internal: V3.1.0

Issue: 20200311

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 due to product version upgrades , adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall 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 the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. 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 bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequent**

ial, exemplary, incidental, 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 contact Alibaba Cloud directly if you discover any errors in this document .**

Document conventions

Style	Description	Example
⛔	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.
⚠	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.
ⓘ	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.
📋	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]`

Style	Description	Example
{} or {a|b}	**This format is used for a required value, where only one item can be selected.**	`switch {active|stand}`

Contents

1 Introduction to Apsara Stack Agility SE

1.1 What is Apsara Stack Agility SE?

Private cloud

A private cloud is a cloud computing system deployed on the premises of an enterprise by a cloud computing service provider. Cloud infrastructure, software , and hardware resources are deployed behind a firewall in the private cloud to allow internal users of the enterprise to share the resources of the data center. The private cloud can be managed by the enterprise or a third party and located within or outside the enterprise. Private clouds provide better privacy and exclusivity than public clouds.

Private clouds are divided into the following types based on the enterprise scale or business requirements:

- Multi-tenant comprehensive private cloud for industries and large groups: an end-to-end cloud system created in a top-down manner. The system is designed to drive hyper-scale digital applications and meet IT requirements such as the continuous integration and development of DevOps applications and the operation support of production environments.
- Single-tenant basic private cloud for small- and medium-sized enterprises and scenarios: a cloud system that can perform local computing tasks and host technical systems such as large-scale Software as a Service (SaaS) applications, industrial clouds, and large group clouds.

Alibaba Cloud Apsara Stack

As more and more enterprises migrate their IT infrastructure to the cloud, they must consider construction requirements such as security compliance, reuse of existing data centers, and the benefits of a collocated data center. These enterprises prefer to use their own data centers but want to deliver a service experience that relies on large-scale cloud computing.

Alibaba Cloud Apsara Stack is an extension of Alibaba Cloud public cloud, which brings the public cloud technologies to Apsara Stack. Apsara Stack delivers complete and customizable Alibaba Cloud software solutions and allows enterprise s to experience the same hyper-scale cloud computing and big data products provided by Alibaba Cloud public cloud within their own data centers. Apsara Stack also provides enterprises with a consistent hybrid cloud experience where you can obtain IT resources as needed and ensure business continuity.

Apsara Stack Agility SE

Small-and medium-sized private clouds have the majority share of the private cloud market. Users tend to deploy private clouds on a small scale. Alibaba Cloud has launched an agile cloud application platform for enterprises to migrate their business to small-and medium-sized private clouds. This platform is designed to provide an open, unified, and trusted cloud platform for enterprises, enhance their core competitiveness in the cloud market, and meet their diverse business requirements.

Apsara Stack Agility SE can be directly deployed and managed based on existing hardware such as x86 architecture to provide secure and stable enterprise-level services. Apsara Stack Agility SE features hybrid deployment of the base, Apsara, network, and storage components to reduce the required number of physical servers, improve resource utilization, and provide scalability of resources as needed. Apsara Stack Agility SE can reduce the number of management and control nodes and provide high availability and data security at a lower cost.

Benefits

Based on a variety of products and services and the digitalization practices of Alibaba Group, and in combination with the mature solutions and rich experience in various industries, Apsara Stack Agility SE helps governments and enterprises digitally transform their businesses and services. Apsara Stack Agility SE provides the following benefits:

- Elastic

  Combines all resources into a single supercomputer and flexibly scales out resources to minimize costs and maximize performance and stability.
- Agile

  Uses Internet and microservice integration to speed up innovation.

- **Data**

  Uses digitalization to allow data to flow vertically between businesses and forms a mid-end to handle large amounts of data.
- **Smart**

  Allows smart transformation of businesses globally and helps reinvent business models.

Platform features

As an enterprise-level cloud platform, Apsara Stack Agility SE has the following features:

- Software-defined platform: masks underlying hardware differences, enables resources to scale up or out as required, and does not affect the performance of upper-layer applications.
- Production-level reliability and security compliance: ensures the continuity and security of enterprise data.
- Unified access management: isolates permissions of different roles to facilitate subsequent O&M management.

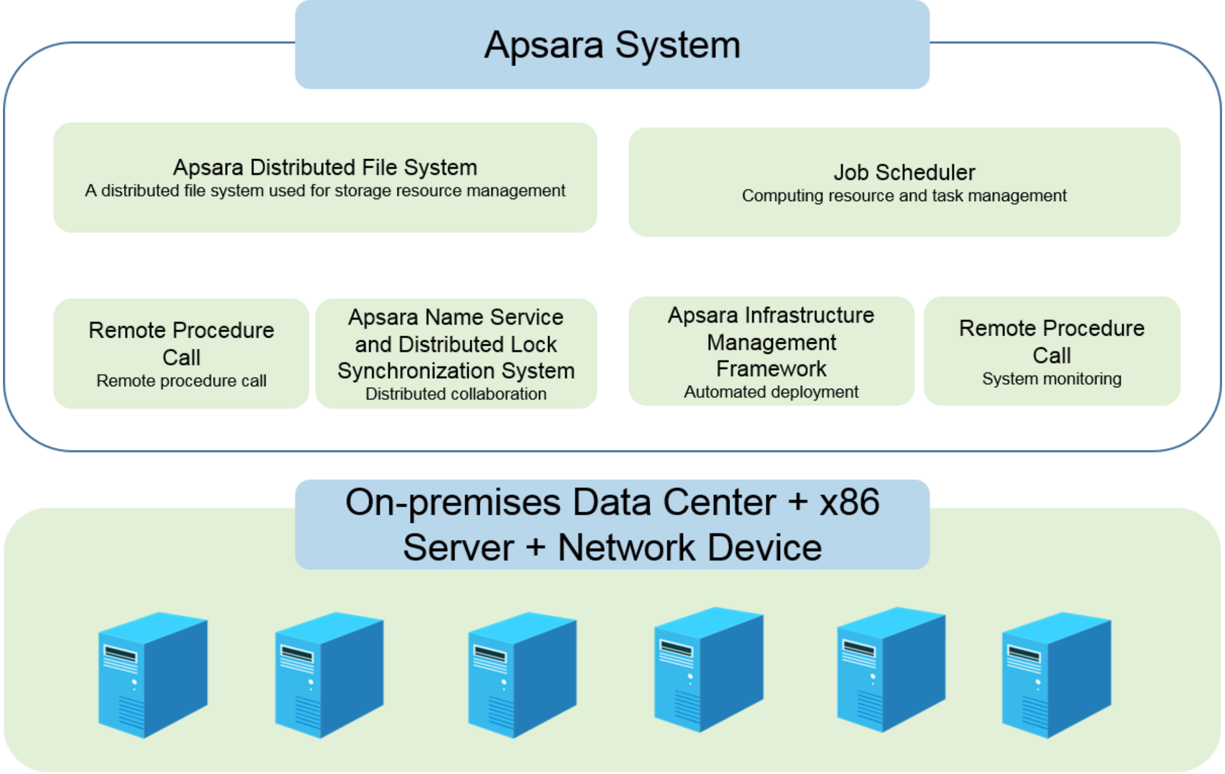
1.2 Why Apsara Stack Agility SE?

This topic describes the benefits of Apsara Stack Agility SE.

1.2.1 Unified distributed cloud operating system

Both Apsara Stack Agility SE and Alibaba Cloud public cloud are based on the Apsara distributed operating system. The Apsara system provides underlying services such as storage, computing, and scheduling for the upper-layer services. The Apsara system is a hyper-scale universal operating system developed by Alibaba Cloud for use both inside and outside China. It connects millions of servers around the world to act as a supercomputer, providing computing capabilities as online public services. The computing capabilities provided by Apsara are powerful, universal, and accessible to everyone.

Figure 1-1: Apsara system kernel architecture


Apsara System
Apsara Distributed File System
A distributed file system used for storage resource management
Job Scheduler
Computing resource and task management
Remote Procedure Call
Remote procedure call
Apsara Name Service and Distributed Lock Synchronization System
Distributed collaboration
Apsara Infrastructure Management Framework
Automated deployment
Remote Procedure Call
System monitoring
On-premises Data Center + x86 Server + Network Device

The Apsara system kernel consists of the following modules:

- Underlying services for distributed systems

  This module provides the coordination, remote procedure call, security management, and resource management services needed in a distributed environment. These services provide support for the upper-layer modules such as the distributed file system and task scheduling module.
- Distributed file system

  This module provides a reliable and scalable service for storing vast amounts of data. It aggregates the storage capabilities of each node in a cluster and automatically protects against hardware and software faults to provide uninterrupted access to data. This module also supports incremental scaling and automatic data balancing. An API similar to Portable Operating System Interface of UNIX (POSIX) is provided to access user space files. Additionally, the module supports random read/write and append write operations.

- Task scheduling

  This module schedules tasks in the cluster system and supports both online services that rely on the response speed and offline tasks that require high data processing throughput. The module can automatically detect faults and hot spots in the system. The module ensures stable and reliable service operations through such methods as error retry and concurrent backup for long-tail operations.
- Cluster monitoring and deployment

  This module monitors the status of clusters as well as the running status and performance metrics of upper-layer application services. It generates alerts and records of exception events. Additionally, the module provides maintenance personnel with deployment and configuration management of the entire Apsara system and its upper-layer applications. The module supports both the online elastic scaling of clusters and the online upgrade of application services.

1.2.2 Apsara Infrastructure Management Framework

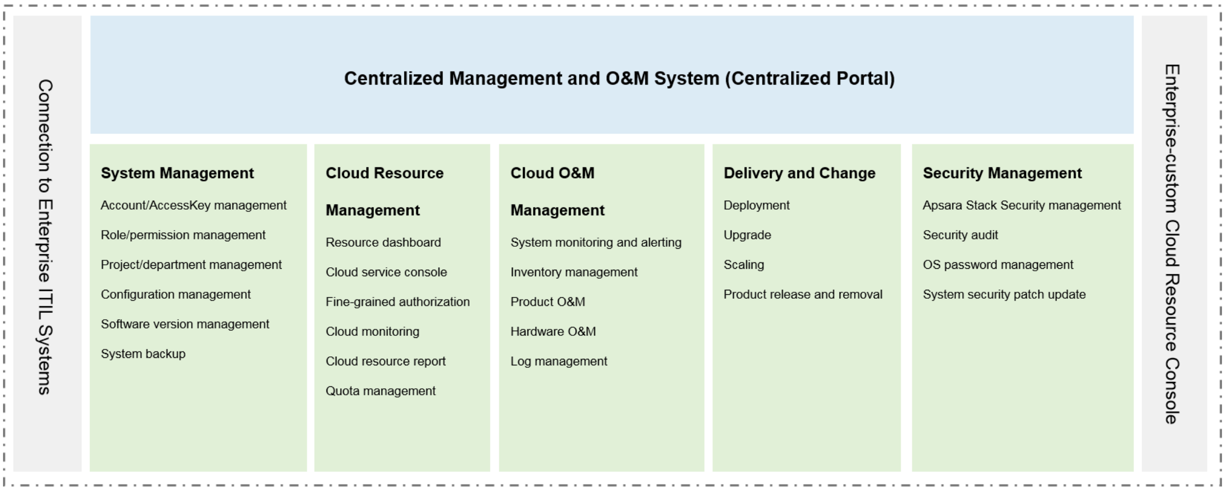
Apsara Infrastructure Management Framework provides cloud services with underlying support capabilities such as centralized deployment, verification, authorization, and control. Apsara Infrastructure Management Framework includes such modules as deployment framework, resource library, metadatabase, Apsara Stack Security, authentication and authorization, API Gateway, Log Service, and control service.

- The deployment framework provides centralized deployment of access platforms and manages service dependencies.
- The resource library stores the executable files of all cloud services and their dependent components.
- Apsara Stack Security protects cloud services from web attacks.
- The authentication and authorization module provides access control capabiliti es for cloud services.
- API Gateway provides a centralized API management platform for cloud services.
- The control service module monitors the basic health status of each cloud service and supports the Apsara Stack O&M system.

1.2.3 Centralized O&M management and automated O&M capability

Apsara Stack Agility SE provides a centralized O&M management portal. You can configure different management permissions for different roles. OpenAPI enables you to manage O&M tasks. You can customize your cloud resource console. Apsara Stack Agility SE supports data synchronization and integration with the existing Information Technology Infrastructure Library (ITIL) systems of enterprises.

Figure 1-2: Centralized O&M management


Connection to Enterprise ITIL Systems
Centralized Management and O&M System (Centralized Portal)
System Management
Account/AccessKey management
Role/permission management
Project/department management
Configuration management
Software version management
System backup
Cloud Resource Management
Resource dashboard
Cloud service console
Fine-grained authorization
Cloud monitoring
Cloud resource report
Quota management
Cloud O&M Management
System monitoring and alerting
Inventory management
Product O&M
Hardware O&M
Log management
Delivery and Change
Deployment
Upgrade
Scaling
Product release and removal
Security Management
Apsara Stack Security management
Security audit
OS password management
System security patch update
Enterprise-custom Cloud Resource Console

1.2.4 OpenAPI

Apsara Stack provides a wide range of SDKs and RESTful APIs on the OpenAPI platform. OpenAPI provides flexible access to a variety of cloud services on Apsara Stack Agility SE. You can also use OpenAPI to obtain the basic control information of Apsara Stack Agility SE and integrate Apsara Stack Agility SE with your centralized control system.

1.3 Architecture

This topic describes the system architecture, network architecture, security architecture, and base modules of Apsara Stack Agility SE.

1.3.1 Types of private cloud architecture

There are two types of private cloud architecture: native cloud architecture and integrated cloud architecture.

- Cloud native architecture

  The cloud native architecture is derived from the Internet-based open architecture. Based on a distributed system framework, the cloud native architecture was used originally for big data and web applications and later used to provide a range of basic services.
- Integrated cloud architecture

  The integrated cloud architecture focuses on the virtualization of computing services. Integrated cloud architecture is a breakthrough from traditional computing architecture developed by OpenStack and has become the most popular choice for private cloud architecture.

Apsara Stack Agility SE employs the cloud native architecture and is based on self-developed distributed technologies and products of Alibaba Cloud. Apsara Stack Agility SE uses a single architecture for a variety of deployment environments to support all cloud products and services. The architecture offers a full set of enterprise-class services, and features complete open capabilities, disaster recovery and backup capabilities, and self-developed and controllable capabilities.

1.3.2 System architecture

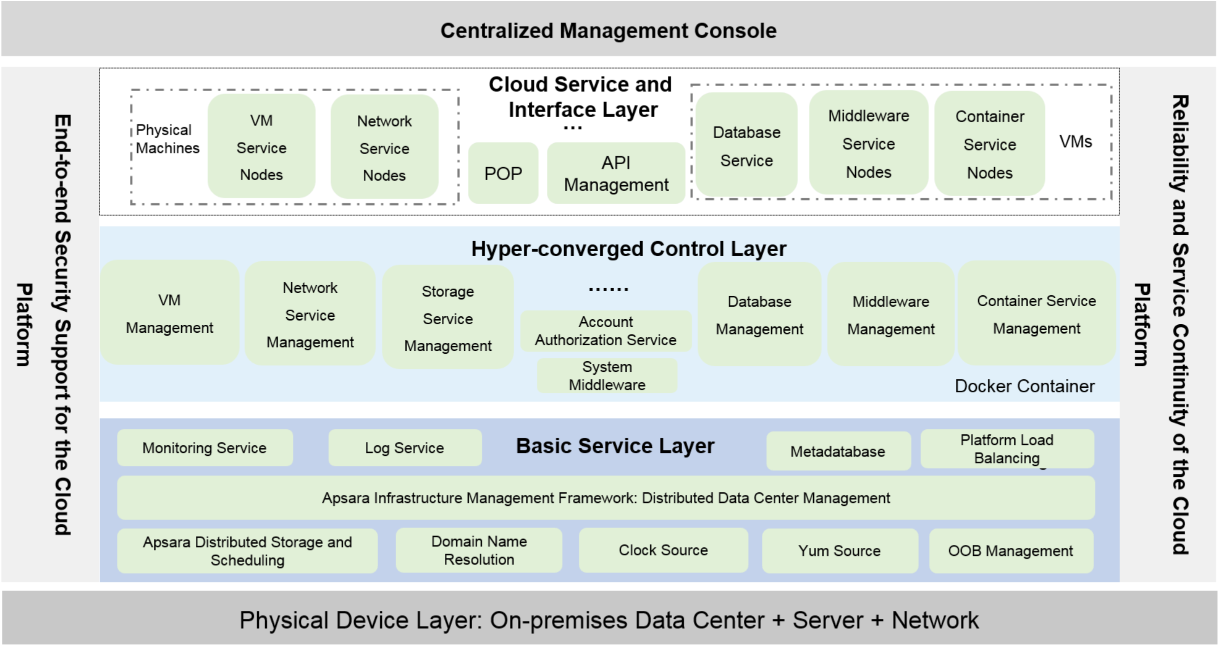
Apsara Stack Agility SE provides a consistent O&M management experience and an enterprise-level cloud security architecture based on the OpenAPI model.

The system architecture of Apsara Stack Agility SE consists of the following layers, as shown in *Figure 1-3: System architecture of Apsara Stack Agility SE*.

- Physical device layer: includes hardware devices for cloud computing, such as physical data centers, servers, and network.
- Basic service layer: provides basic services for upper-layer applications based on the underlying physical environment.
- Hyper-converged control layer: provides centralized scheduling for upper-layer applications and services based on a hyper-converged control architecture.
- Cloud service and interface layer: provides centralized management and O&M for virtual machines and physical machines through converged service node management, and uses the OpenAPI platform to provide centralized API management and support custom development.
- Centralized management layer: provides centralized operations and maintenance management.

Apsara Stack Agility SE also provides end-to-end security to ensure the reliability and service continuity of the cloud platform.

Figure 1-3: System architecture of Apsara Stack Agility SE


Centralized Management Console
End-to-end Security Support for the Cloud Platform
Cloud Service and Interface Layer
Physical Machines
VM Service Nodes
Network Service Nodes
…
POP
API Management
Database Service
Middleware Service Nodes
Container Service Nodes
VMs
Reliability and Service Continuity of the Cloud Platform
Hyper-converged Control Layer
VM Management
Network Service Management
Storage Service Management
……
Account Authorization Service
System Middleware
Database Management
Middleware Management
Container Service Management
Docker Container
Monitoring Service
Log Service
Basic Service Layer
Metadatabase
Platform Load Balancing
Apsara Infrastructure Management Framework: Distributed Data Center Management
Apsara Distributed Storage and Scheduling
Domain Name Resolution
Clock Source
Yum Source
OOB Management
Physical Device Layer: On-premises Data Center + Server + Network

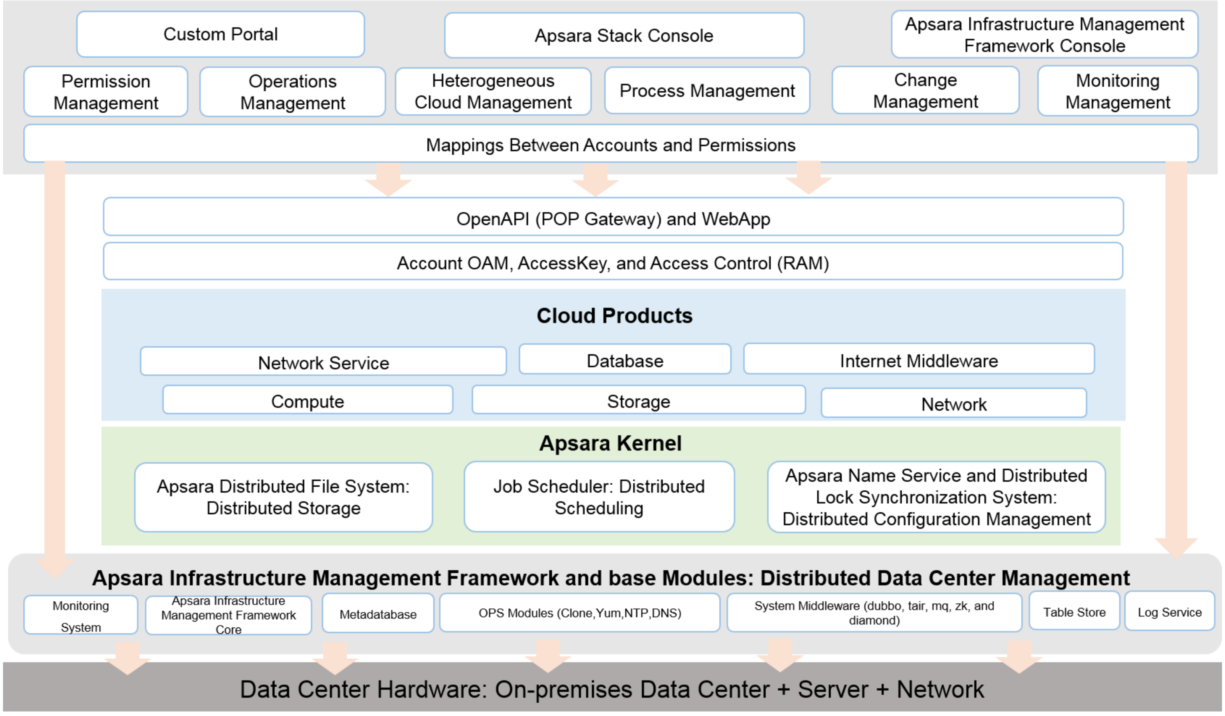
Logical architecture

Apsara Stack Agility SE virtualizes physical servers and network devices to be used for virtual computing, distributed storage, and software defined networks (SDNs). Additionally, it offers ApsaraDB and distributed middleware services to provide IT infrastructure support for applications. Apsara Stack Agility SE can be integrated with your existing account, monitoring, and O&M systems. The logical architecture of Apsara Stack Agility SE has the following characteristics:

- **The hardware infrastructure of Apsara Stack Agility SE consists of on-premises data centers, x86 servers, and network devices.**
- **A variety of cloud services are provided based on the Apsara kernel (distributed engine).**
- **All cloud services are required to comply with a unified API framework, security system, and O&M and management system (account, authorization, monitoring, and log).**

- **A consistent user experience is guaranteed across all services.**

Figure 1-4: Logical architecture of Apsara Stack Agility SE


Custom Portal
Apsara Stack Console
Apsara Infrastructure Management Framework Console
Permission Management
Operations Management
Heterogeneous Cloud Management
Process Management
Change Management
Monitoring Management
Mappings Between Accounts and Permissions
OpenAPI (POP Gateway) and WebApp
Account OAM, AccessKey, and Access Control (RAM)
Cloud Products
Network Service
Database
Internet Middleware
Compute
Storage
Network
Apsara Kernel
Apsara Distributed File System: Distributed Storage
Job Scheduler: Distributed Scheduling
Apsara Name Service and Distributed Lock Synchronization System: Distributed Configuration Management
Apsara Infrastructure Management Framework and base Modules: Distributed Data Center Management
Monitoring System
Apsara Infrastructure Management Framework Core
Metadatabase
OPS Modules (Clone,Yum,NTP,DNS)
System Middleware (dubbo, tair, mq, zk, and diamond)
Table Store
Log Service
Data Center Hardware: On-premises Data Center + Server + Network

1.3.3 Network architecture

Apsara Stack Agility SE is a minimized version of Apsara Stack. The network architecture of Apsara Stack Agility SE is optimized and streamlined to only include Inter-connection Switch (ISW) and Access Switch (ASW) device roles. Apsara Stack Agility SE supports up to 96 servers, and uses MiniLVS in place of SLB to support the Border Gateway Protocol (BGP).

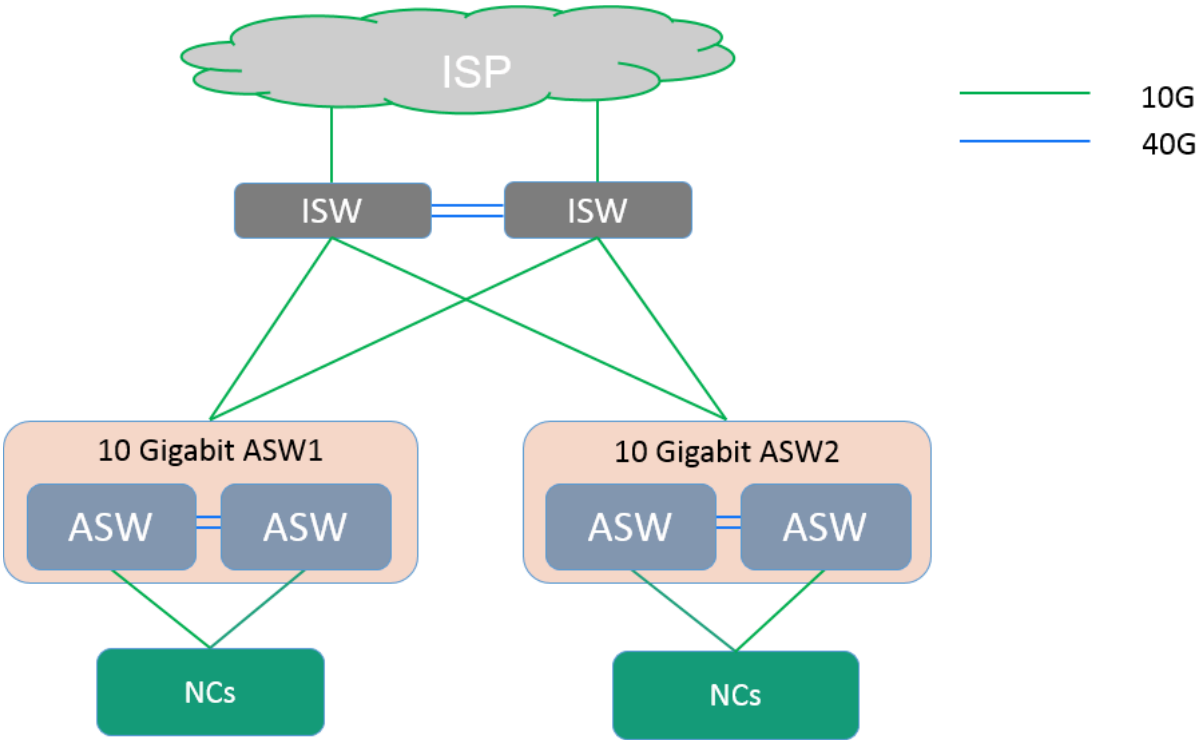
The following table lists the roles and functions of switches at different layers.

Table 1-1: Role definition

Role name	Function
ISW	The inter-connection switch. ISWs provide access to Internet service providers (ISPs) and are internally connected to ASWs.
ASW	The access switch. ASWs provide access to ECS instances and are uplinked to ISWs.
OOB	The out-of-band management.
OMR	The out-of-band manager switch.

Role name	Function
OASW	The out-of-band access switch. OASWs are connected to Intelligent Platform Management Interface (IPMI) cards on servers.
ACS	The advanced console server. An ACS is connected to the console port of a network device for management purposes.

Figure 1-5: Logical zones in the network architecture

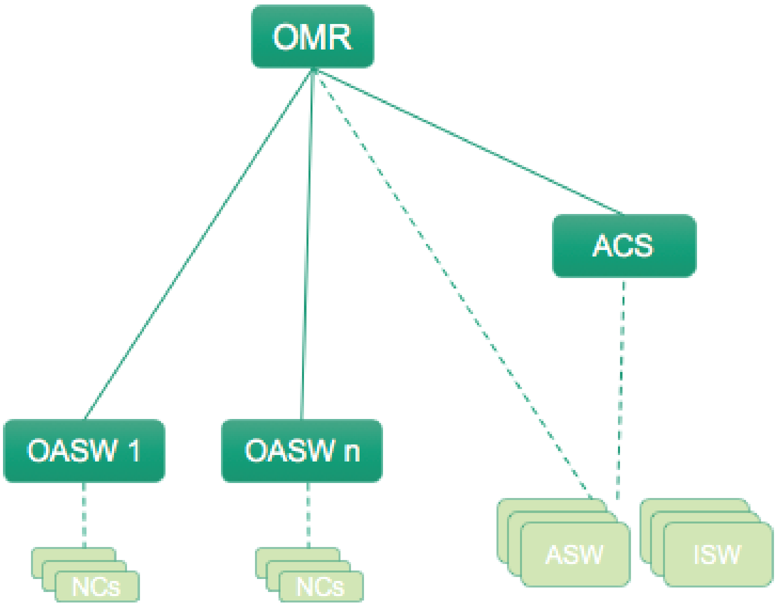


In the network architecture of Apsara Stack Agility SE, ISWs provide access to ISPs and are internally connected to ASWs. The external bandwidth can be configured to suit your needs. Two ISWs are always deployed in this architecture. The two ISWs are interconnected at a bandwidth of 2*40 Gbit/s, and are downlinked to each ASW group at a bandwidth of 320 Gbit/s.

In the network architecture of Apsara Stack Agility SE, ASWs are connected to servers to provide network capacity for all cloud services. Two ASWs are stacked to form a group. Apsara Stack Agility SE supports up to two groups of ASWs and up to 96 servers. Each ASW group is connected to an ISW at a bandwidth of 320 Gbit/s, and connected to a server at a bandwidth of 960 Gbit/s. The network convergence ratio is 1:3.

All servers are configured with two NICs. Each server is connected to two ASWs by means of NIC bonding and provides 20 Gbit/s outbound bandwidth.

Figure 1-6: OOB management network



The OOB management network in the network architecture of Apsara Stack Agility SE manages servers and switches in a cluster. This network is necessary for Apsara Infrastructure Management Framework to perform operations such as installing and restarting physical servers.

Server management network

The IPMI port of each server is uplinked to an OASW through a GE network cable. Each OASW provides 48 GE network ports. The OASW supports Layer 2 passthrough and is uplinked to an OMR. The gateway function is configured on the OMR.

Management port connection in the network device zone

The management port of each network device is uplinked to an OMR through a GE network cable.

Console port connection in the network device zone

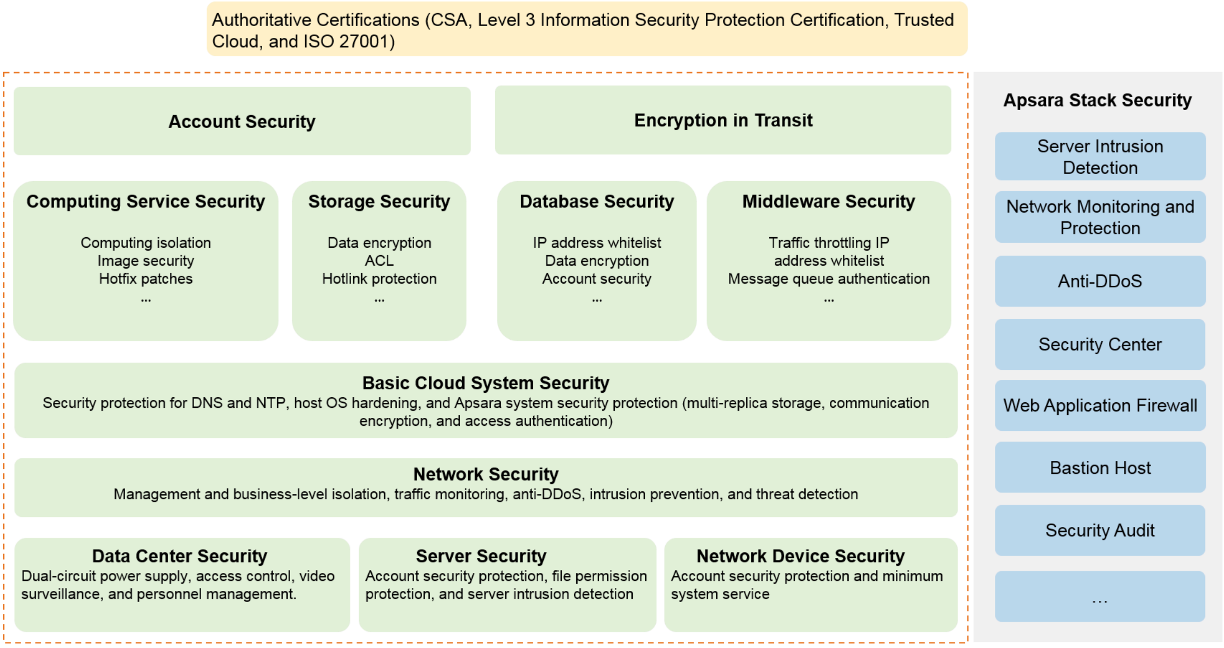
The console port of each network device is uplinked to an ACS through a GE network cable. The ACS is uplinked to an OMR.

1.3.4 Security architecture

Cloud products have both frontend services and backend systems. Therefore, the security architecture of Apsara Stack Agility SE is divided into two layers: the platform layer and the user layer.

Apsara Stack Agility SE provides comprehensive security capabilities from underlying communication protocols all the way to upper-layer applications to secure your data and access. All consoles must be accessed using HTTPS certificates. Apsara Stack Agility SE provides a complete role authorization mechanism to ensure secure and controllable access to resources in multi-tenant mode. Apsara Stack Agility SE supports a variety of security roles, including security administrators, system administrators, and security auditors.

Figure 1-7: Hierarchical security architecture of Apsara Stack Agility SE


Authoritative Certifications (CSA, Level 3 Information Security Protection Certification, Trusted Cloud, and ISO 27001)
Account Security
Encryption in Transit
Computing Service Security
Computing isolation
Image security
Hotfix patches
...
Storage Security
Data encryption
ACL
Hotlink protection
...
Database Security
IP address whitelist
Data encryption
Account security
...
Middleware Security
Traffic throttling IP address whitelist
Message queue authentication
...
Basic Cloud System Security
Security protection for DNS and NTP, host OS hardening, and Apsara system security protection (multi-replica storage, communication encryption, and access authentication)
Network Security
Management and business-level isolation, traffic monitoring, anti-DDoS, intrusion prevention, and threat detection
Data Center Security
Dual-circuit power supply, access control, video surveillance, and personnel management.
Server Security
Account security protection, file permission protection, and server intrusion detection
Network Device Security
Account security protection and minimum system service
Apsara Stack Security
Server Intrusion Detection
Network Monitoring and Protection
Anti-DDoS
Security Center
Web Application Firewall
Bastion Host
Security Audit
...

1.3.5 Base modules

Apsara Stack Agility SE base consists of three module types, all of which provide support for the deployment and O&M of the cloud platform.

Table 1-2: Base modules

Module		Description
OPS modules	Yum	Installation packages Software repositories are deployed in the initial installation stage to install the operating system and deploy application packages such as the Apsara system, and dependent modules of Apsara Stack Agility SE on hosts.
	Clone	Virtual machine cloning service
	NTP	Clock source service NTP is deployed on hosts of Apsara Stack Agility SE to synchronize time from the standard NTP clock source to other hosts.
	DNS	Domain name resolution service DNS provides forward and reverse resolution of domain names for the internal Apsara Stack Agility SE environment. It runs a bind instance on each of the two OPS machines and uses keepalived to provide high availability services. When one machine fails, the other machine automatically takes over its work.
Base middleware	Dubbo	Distributed Remote Procedure Call (RPC) service
	Tair	Cache service
	MQ	Message Queue service
	ZooKeeper	Distributed collaboration

Module		Description
	Diamond	Configuration management service
	SchedulerX	Timing task service
Basic modules of the base	Apsara Infrastructure Management Framework	Data center management
	Monitoring System	Data center monitoring
	Metadatabase	Metadatabase
	POP	Apsara Stack OpenAPI
	OAM	Account system
	RAM	Authentication and authorization system
	WebApps	Support for the Apsara Stack Operations console

1.4 Product panorama

Apsara Stack Agility SE offers a wide range of services to meet the diverse needs of different users.

- IaaS

  Apsara Stack Agility SE provides basic computing, network, and storage capabilities. The main services include Elastic Compute Service (ECS), Virtual Private Cloud (VPC), Server Load Balancer (SLB), and Block Storage.

- Storage services

  Apsara Stack Agility SE provides a wide range of storage services for different storage objects. The main services include Object Storage Service (OSS), Block Storage, and Hybrid Cloud Storage Array (HCSA).

- **Database services**

  Apsara Stack Agility SE provides a variety of database engines that can communicate with each other. The main services include ApsaraDB RDS for MySQL, AnalyticDB for MySQL, AnalyticDB for PostgreSQL, KVStore for Redis, Distributed Relational Database Service (DRDS), and Data Transmission Service (DTS).

1.5 Scenarios

Apsara Stack Agility SE provides flexible and scalable industrial solutions for customers of different scales and sectors. Apsara Stack Agility SE can create customized solutions based on the business traits of different sectors such as industry, agriculture, transportation, government, finance, and education to provide users with end-to-end products and services. This topic highlights the scenarios of small-scale private cloud focusing on storage services.

Scenarios of small-scale private cloud focusing on IaaS

Apsara Stack Agility SE delivers the same IaaS experience as public cloud, such as providing virtual machines, virtual networks, load balancing capabilities, and cloud disks.

Scenarios of small-scale private cloud focusing on storage services

The small-scale private cloud focusing on storage services targets traditional storage markets and is characterized by high availability, high throughput, low latency, and high scalability. The storage services include Object Storage Service (OSS), Block Storage, and Hybrid Cloud Storage Array (HCSA).

Values and features

- **The integrated storage platform improves resource utilization and greatly reduces deployment and operations costs.**
- **The total bandwidth increases linearly with the expansion of nodes, and system performance is guaranteed during elastic scaling to adapt to future business trends.**
- **Compared with the traditional storage, the small-scale private cloud focusing on storage services has greatly improved the concurrent processing capability and read/write speed.**

More and more enterprises, especially financial institutions such as banks, securities and insurance firms, and fund management companies, want to build distributed databases to support Internet-based businesses. Additionally, this lightweight Apsara system can be combined with Apsara Stack Enterprise to provide a low-cost remote disaster recovery solution.

Scenarios of small-scale private cloud focusing on database services

The small-scale private cloud focusing on database services targets traditional database markets and provides high-availability and high-performance transactio nal or analytic database services. The database services include ApsaraDB RDS for MySQL, AnalyticDB for MySQL, AnalyticDB for PostgreSQL, Distributed Relational Database Service (DRDS), and Data Transmission Service (DTS).

Values and features

- **The integrated database platform improves resource utilization and greatly reduces deployment and operations costs.**
- **The total bandwidth increases linearly with the expansion of nodes, and system performance is guaranteed during elastic scaling to adapt to future business trends.**
- **ApsaraDB for RDS Enterprise Edition offers strong consistency.**
- **Failure of any single server does not affect services.**
- **The entire data center is automatically restored after power outage or network disconnection.**

More and more enterprises, especially financial institutions such as banks, securities and insurance firms, and fund management companies, want to build databases. This platform is ideal for these scenarios.

2 Object Storage Service (OSS)

2.1 What is OSS?

Alibaba Cloud Object Storage Service (OSS) is a massive, secure, low-cost, and highly reliable cloud storage service provided by Alibaba Cloud.

It can be considered as an out-of-the-box storage solution with unlimited storage capacity. Compared with the user-created server storage, OSS has many outstandin g advantages in reliability, security, cost, and data processing capabilities. Using OSS, you can store and retrieve a variety of unstructured data files, such as text files , images, audios, and videos, over the network at any time.

OSS uploads data files as objects to buckets. OSS is an object storage service that uses a key-value pair format. You can retrieve object content based on unique object names (keys).

On OSS, you can:

- Create a bucket and upload objects to the bucket.
- Obtain an object URL from OSS to share or download an object.
- Complete the ACL settings of a bucket or object by modifying its properties or metadata.
- Perform basic and advanced OSS tasks through the OSS console.
- Perform basic and advanced OSS tasks using the Alibaba Cloud SDKs or directly calling the RESTful APIs in your application.

2.2 Advantages

Advantages of OSS over user-created server storage

Item	OSS	User-created server storage
Reliability	· Automatically expands capacities without affecting your services. · Supports automatic redundant data backup.	· Prone to errors due to low hardware reliability. If a disk has a bad sector, data may be irretrievably lost. · Manual data recovery is complex and requires a lot of time and technical resources.
Security	· Provides hierarchical security protection for enterprises. · Provides user resource isolation mechanisms and supports zone-disaster recovery. · Provides various authentication and authorization mechanisms. It also provides features such as whitelisting, hotlink protection, RAM, and Security Token Service (STS) for temporary access.	· Additional scrubbing and black hole equipment is required. · A separate security mechanism is required.

More advantages of OSS

- **Ease of use**

  The standard RESTful APIs (some compatible with Amazon S3 APIs) are supported. A wide range of SDKs, client tools, and console are provided. You can upload, download, retrieve, and manage large amounts of data for websites and mobile apps the way you use regular files systems.

  - **There is no limit on the number and size of objects. Therefore, you can expand your buckets in OSS as required.**
  - **Streaming writes and reads are supported, which is suitable for business scenarios where you need to simultaneously read and write videos and other large objects.**
  - **Lifecycle management is supported. You can delete multiple expired data.**

- **Powerful and flexible security mechanisms**

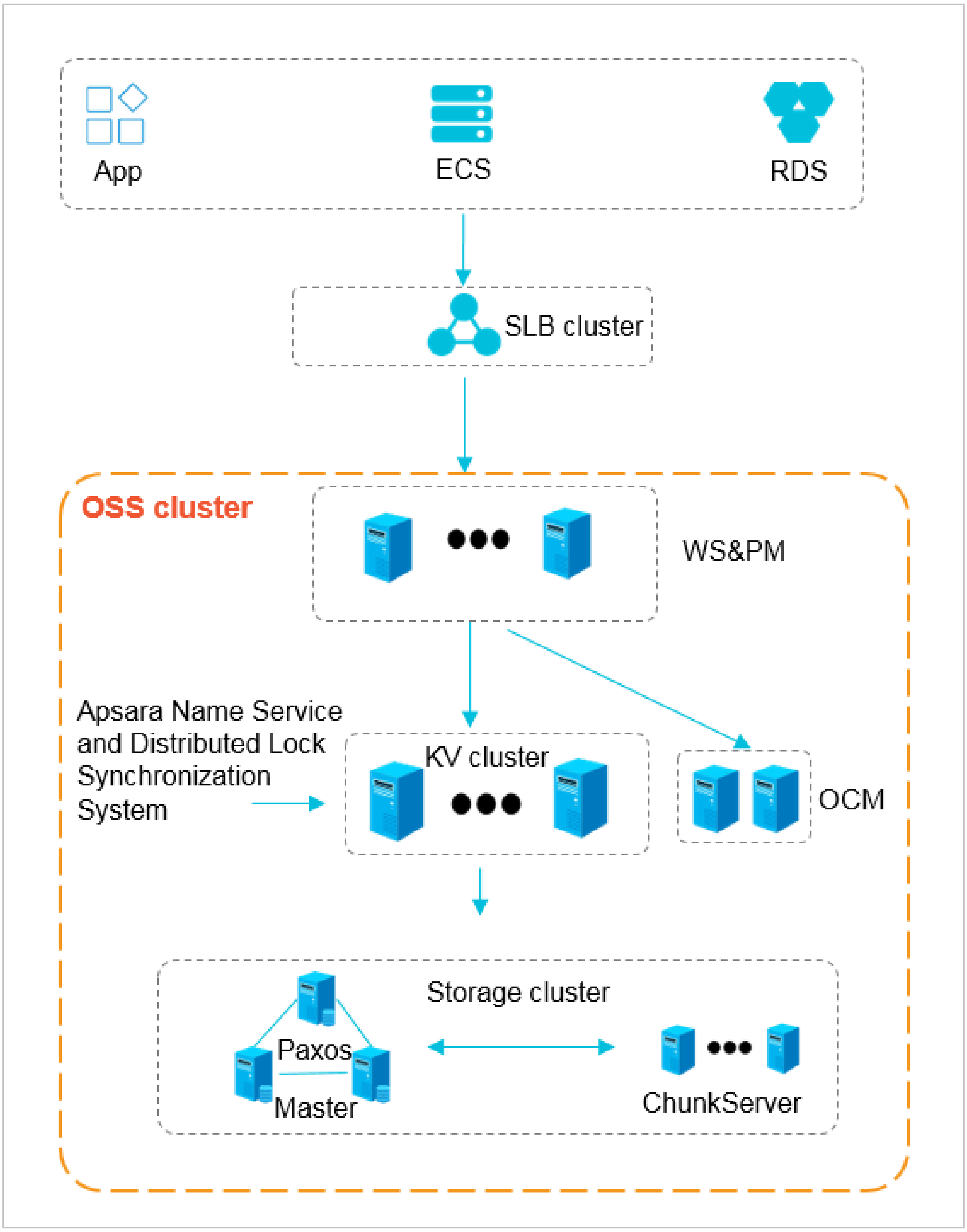
  Flexible authentication and authorization mechanisms are available. OSS provides STS and URL-based authentication and authorization mechanisms, as well as whitelisting, hotlink protection, and RAM.

2.3 OSS architecture

Object Storage Service (OSS) is a storage solution built on the Alibaba Cloud Apsara platform. It is based on infrastructure such as Apsara Distributed File System and SchedulerX. Such infrastructure provides OSS and other Alibaba Cloud services

with distributed scheduling, high-speed networks, and distributed storage features . The following figure shows the OSS architecture.

Figure 2-1: OSS architecture



- **WS & PM (the protocol layer): is used for receiving users' requests sent through the REST protocol and performing authentication. If the authentication succeeds**

, users' requests are forwarded to the key-value engine for further processing. If the authentication fails, an error message is returned.

- KV cluster: is used for processing structured data, including reading and writing data based on keys. The KV cluster also supports large-scale concurrent requests . When a service has to operate on a different physical server due to a change in the service coordination cluster, the KV cluster can quickly coordinate and find the access point.
- Storage cluster: Metadata is stored on the master node. A distributed message consistency protocol (Paxos) is adopted between master nodes to ensure the consistency of metadata. This ensures efficient distributed storage and access of objects.

2.4 Features

The following table describes features of OSS.

Table 2-1: OSS features

Category	Feature	Description
Bucket	Create a bucket	Before you upload an object to OSS, you must have a bucket to contain the object.
	Delete a bucket	If you no longer use a bucket, delete it to avoid incurring further fees.
	Modify the ACL for a bucket	OSS provides ACL for access control. You can configure ACL when creating a bucket and modify the ACL after creating the bucket.
	Configure static website hosting	You can configure static website hosting for your bucket and access this static website through the bucket domain name.
	Configure hotlink protection	To prevent fees incurred by hotlinked OSS data , OSS provides hotlink protection based on the Referer field in the HTTP header.
	Manage CORS	OSS provides CORS in HTML5 to implement cross-origin access.
	Configure lifecycle	You can define and manage the lifecycle of all or a subset of objects in a bucket. Lifecycle is configured to manage multiple objects and automatically delete parts.

Category	Feature	Description
Object	Upload an object	You can upload all types of objects to a bucket.
	Create a folder	You can manage OSS folders the way you manage folders in Windows.
	Search for objects	You can search for objects whose names contain the same prefix in a bucket or folder.
	Obtain an object URL	You can obtain an object URL from OSS to share or download an object.
	Delete objects	You can delete a single object or multiple objects.
	Delete a folder	You can delete a single folder or multiple folders.
	Modify the ACL for an object	You can configure ACL when you upload an object and modify the ACL after you upload the object.
	Manage parts	You can delete all or some parts from a bucket.
API operation	API operation	RESTful API operations are supported and relevant examples are provided.
SDK	SDK	SDK-based development operations and relevant examples for various programming languages are provided.

2.5 Scenarios

Massive storage for image, audio, and video applications

OSS can be used to store large amounts of data, such as images, audios, videos, 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

OSS leverages the BGP bandwidth to achieve ultra-low latency of direct data download.

Offline data storage

OSS is cheap and highly available, enabling enterprises to store data that needs to be archived offline for a long time to OSS.

2.6 Limits

Item	Description
Bucket	· You can create a maximum of 100 buckets. · After a bucket is created, its name and region cannot be modified.
Object upload	· Objects larger than 5 GB cannot be uploaded by using the following methods: console upload, simple upload, form upload, and append upload . To upload an object that is larger than 5 GB, you must use multipart upload. The size of an object uploaded by using multipart upload cannot exceed 48.8 TB. · If you upload an object that has the same name of an existing object in OSS, the new object will overwrite the existing object.
Object deletion	· Deleted objects cannot be recovered. · You can delete up to 50 objects at a time in the Apsara Stack Cloud Management (ASCM) console. To delete more objects at a time, you must use APIs or SDKs.
Lifecycle	You can configure up to 1,000 lifecycle rules for each bucket.

2.7 Terms

This topic describes several basic terms used in OSS.

object

Files that are stored in OSS. They are the basic unit of data storage in OSS. An object is composed of Object Meta, object content, and a key. An object is uniquely identified by a key in the bucket. Object Meta defines the properties of an object , such as the last modification time and the object size. You can also specify User Meta for the object.

The lifecycle of an object starts when it is uploaded, and ends when it is deleted. Throughout the lifecycle of an object, Object Meta cannot be changed. Unlike the file system, OSS does not allow you to modify objects directly. If you want to modify an object, you must upload a new object with the same name as the existing one to replace it.

Note:

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

bucket

A container that stores objects. Objects must be stored in the bucket they are uploaded to. You can set and modify the properties of a bucket for object access control and lifecycle management. These properties apply to all objects in the bucket. Therefore, you can create different buckets to implement different management functions.

- OSS does not have the hierarchical structure of directories and subfolders as in a file system. All objects belong to their corresponding buckets.
- You can have multiple buckets.
- A bucket name must be globally unique within OSS and cannot be changed after a bucket is created.
- A bucket can contain an unlimited number of objects.

strong consistency

A feature of operations in OSS. Object operations in OSS are atomic, which indicates that operations are either successful or failed. There are no intermediate states. OSS never writes corrupted or partial data.

Object operations in OSS are strongly consistent. For example, after you receive a successful upload (PUT) response, the object can be read immediately, and the data is already written in triplicate. Therefore, OSS avoids the situation where no data is obtained when you perform the read-after-write operation. An object also has no intermediate states when you delete the object. After you delete an object, that object no longer exists.

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

Comparison between OSS and the file system

OSS is a distributed object storage service that uses a key-value pair format. You can retrieve object content based on unique object names (keys). Although you can use names like test1/test.jpg, this does not necessarily indicate that the object is saved in a directory named test1. In OSS, test1/test.jpg is only a string, which is no different from a.jpg. Therefore, similar resources are consumed when you access objects that have different names.

A file system uses a typical tree index structure. Before accessing a file named test1 /test.jpg, you must access directory test1 and then locate test.jpg. This makes it easy for a file system to support folder operations, such as renaming, deleting, and moving directories, because these operations are only directory node operations. System performance depends on the capacity of a single device. The more files and directories that are created in the file system, the more resources are consumed, and the lengthier your process becomes.

You can simulate similar functions in OSS, but this operation is costly. For example , if you want to rename test1 directory test2, the actual OSS operation would be to replace all objects whose names start with test1/ with copies whose names start with test2/. Such an operation would consume a large amount of resources. Therefore, try to avoid such operations when using OSS.

You cannot modify objects stored in OSS. A specific API must be called to append an object, and the generated object is of a different type from that of normally uploaded objects. Even if you only want to modify a single Byte, you must re-upload the entire object. 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 sporadic bursts of access, whereas the performance of a file system is subject to the performance of a single device.

Therefore, mapping OSS objects to file systems is inefficient, which is not recommended. If attaching OSS as a file system is required, we recommended that you perform only the operations of writing data to new files, deleting files, and reading files. You can make full use of OSS capabilities. For example, you can use OSS to store and process large amounts of unstructured data such as images, videos , and documents.

3 ApsaraDB for RDS

3.1 What is ApsaraDB for RDS?

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

Originally based on a branch of MySQL, ApsaraDB RDS for MySQL has proven its performance and throughput during the high-volume concurrent traffic of Double 11. ApsaraDB RDS for MySQL provides whitelist configuration, backup and restoration, transparent data encryption, data migration, and management for instances, accounts, and databases. It also provides the following advanced features:

- Read-only instance: In scenarios where RDS has a small number of write requests but a large number of read requests, you can enable read/write splitting to distribute read requests away from the primary instance. Read-only instances allow ApsaraDB RDS for MySQL 5.6 to automatically scale the reading capability and increase the application throughput when a large amount of data is being read.
- Data compression: ApsaraDB RDS for MySQL 5.6 allows you to compress data by using the TokuDB storage engine. Data transferred from the InnoDB storage engine to the TokuDB storage engine can be reduced by 80% to 90% in volume. 2 TB of data in InnoDB can be compressed to 400 GB or less in TokuDB. In addition to data compression, TokuDB supports transaction and online DDL operations. TokuDB is compatible with MyISAM and InnoDB applications.

3.2 Benefits

3.2.1 Ease of use

ApsaraDB for RDS is a ready-to-use service featuring on-demand upgrades, convenient management, high transparency, and high compatibility.

Ready-to-use

You can use the API to create instances of any specified RDS instance type.

On-demand upgrade

When the database load or data storage capacity changes, you can upgrade the RDS instance by changing its type. The upgrades do not interrupt the data link service.

Transparency and compatibility

ApsaraDB for RDS is used in the same way as the native RDS database engine, allowing it to be adopted easily without the need to learn new database engines . ApsaraDB for RDS is compatible with existing programs and tools. Data can be migrated to ApsaraDB for RDS through ordinary import and export tools.

Easy management

Alibaba Cloud is responsible for the routine maintenance and management tasks for ApsaraDB for RDS such as troubleshooting hardware and software issues or issuing database patches and updates. You can also manually add, delete, restart, back up, and restore databases through the Apsara Stack console.

3.2.2 High performance

ApsaraDB for RDS implements parameter optimization, SQL optimization, and high-end backend hardware to achieve high performance.

Parameter optimization

All RDS instance parameters have been optimized over their several years of production. Professional database administrators continue to optimize RDS instances over their lifecycles to ensure that ApsaraDB for RDS runs at peak efficiency.

SQL optimization

ApsaraDB for RDS locks inefficient SQL statements and provides recommendations to optimize code.

High-end backend hardware

All servers used by ApsaraDB for RDS are evaluated by multiple parties to ensure stability.

3.2.3 High security

ApsaraDB for RDS implements anti-DDoS protection, access control, system security, and transparent data encryption (TDE) to guarantee the security of your databases.

DDoS attack prevention

Note:
You must activate Alibaba Cloud security services to use this feature.

When you access an ApsaraDB for RDS instance from the Internet, the instance is vulnerable to DDoS attacks. When a DDoS attack is detected, the RDS security system first scrubs inbound traffic. If traffic scrubbing is insufficient or if the black hole threshold is reached, black hole filtering is triggered.

Triggering conditions for traffic scrubbing and black hole filtering are listed as follows:

- **Traffic scrubbing:**

  Traffic scrubbing only targets traffic from the Internet and does not affect normal operations of your instance.

  ApsaraDB for RDS triggers and stops traffic scrubbing automatically. Traffic scrubbing is triggered for a single ApsaraDB for RDS instance if any of the following conditions are met:

  - **Packets per second (PPS) reaches 30,000.**
  - **Bits per second (BPS) reaches 180 Mbit/s.**
  - **The number of new concurrent connections per second reaches 10,000.**
  - **The number of active concurrent connections reaches 10,000.**
  - **The number of inactive concurrent connections reaches 10,000.**
- **Black hole filtering:**

  Black hole filtering only targets traffic from the Internet. If an RDS instance is undergoing black hole filtering, the instance cannot be accessed from the

Internet and connected applications will not be available. Black hole filtering guarantees availability of RDS.

Conditions for triggering black hole filtering are listed as follows:

- **BPS reaches 2 Gbit/s.**
- **Traffic scrubbing is ineffective.**

Black hole filtering is automatically stopped 2.5 hours after being triggered.

Access control

You can configure an IP address whitelist for ApsaraDB for RDS to allow access for specified IP addresses and deny access for all others.

Each account can only view and operate their own respective database.

System security

ApsaraDB for RDS is protected by several layers of firewalls capable of blocking a variety of attacks to secure data.

ApsaraDB for RDS servers cannot be logged onto directly. Only the ports required for specific database services are provided.

ApsaraDB for RDS servers cannot initiate an external connection. They can only receive access requests.

3.2.4 High reliability

ApsaraDB for RDS provides hot standby, multi-copy redundancy, data backup, and data recovery to achieve high reliability.

Hot standby

ApsaraDB for RDS adopts a hot standby architecture. If the primary server fails, services will fail over to the secondary server within seconds. Applications running on the servers are not affected by the failover process and will continue to run normally.

Multi-copy redundancy

ApsaraDB for RDS servers implement a RAID architecture to store data. Data backup files are stored on OSS.

Data backup

ApsaraDB for RDS provides an automatic backup mechanism. You can schedule backups to be performed periodically, or manually initiate temporary backups as necessary to meet your business needs.

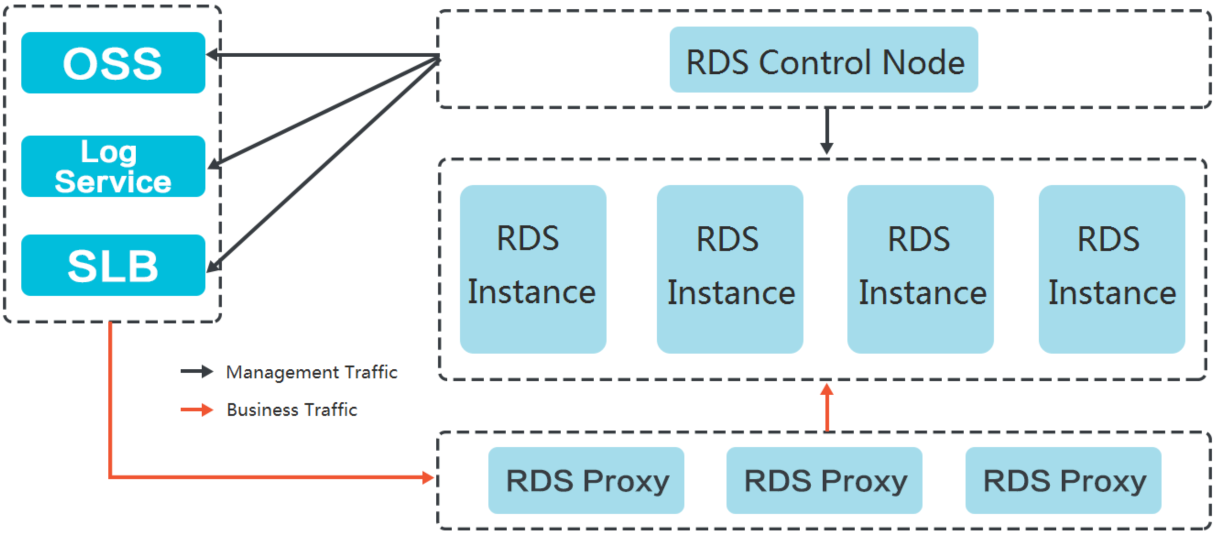
Data recovery

Data can be restored from backup sets or cloned instances created at previous points in time. After data is verified, the data can be migrated back to the primary RDS instance.

3.3 Architecture

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

Figure 3-1: RDS system architecture



3.4 Features

3.4.1 Data link service

ApsaraDB for RDS provides all data link services, including DNS, Server Load Balancer (SLB), and Proxy.

ApsaraDB for RDS uses native database engines with similar database operations to minimize learning costs and facilitate database access.

DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore , IP address changes do not affect the performance of RDS instances. After the domain name of an RDS instance is configured in the connection pool, the RDS instance can be accessed even if its corresponding IP address changes.

For example, the domain name of an ApsaraDB for RDS instance is `test.rds.aliyun.com`**, and its corresponding IP address is** `10.10.10.1`**. The instance can be accessed when either** `test.rds.aliyun.com` **or** `10.10.10.1` **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 `10.10.10.2`**. 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 configured in the connection pool is** `10.10.10.1`**, the instance will no longer be accessible.**

SLB

The SLB module provides both the internal IP address and public IP address of an ApsaraDB for RDS instance. Therefore, server changes do not affect the performanc e of the instance.

For example, 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 hot standby status 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.**

Proxy

The Proxy module provides a number of features including data routing, traffic detection, and session persistence.

- **Data routing: aggregates the distributed complex queries found in big data scenarios and provides the corresponding capacity management capabilities.**
- **Traffic detection: reduces SQL injection risks and supports SQL log backtracking when necessary.**

- Session persistence: prevents database connection interruptions when faults occur.

3.4.2 High-availability service

The high-availability (HA) service consists of modules such as the Detection, Repair, and Notice.

The HA service guarantees the availability of data link services and processes internal database exceptions.

Detection

The Detection module checks whether the primary and secondary nodes of the DB Engine are providing their 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 within 30 seconds.

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.

HA policies

Each HA policy defines a combination of service priorities and data replication modes defined to meet the needs of your business.

There are two service priorities:

- Recovery time objective (RTO): The database preferentially restores services to maximize the availability time. Use the RTO policy if you require longer database uptime.
- Recovery point objective (RPO): The database preferentially ensures data reliability to minimize data loss. Use the RPO policy if you require high data consistency.

There are three data replication modes:

- Asynchronous replication (Async): When an application initiates an update request such as add, delete, or modify operations, the primary node responds to the application immediately after the primary node completes the operation . The primary node then replicates data to the secondary node asynchronously . This means that the operation of the primary database is not affected if the secondary node is unavailable. Data inconsistencies may occur if the primary node is unavailable.
- Forced synchronous replication (Sync): When an application initiates an update request such as add, delete, or modify operations, the primary node replicates data to the secondary node immediately after the primary node completes the operation. The primary node then waits for the secondary node to return a success message before the primary node responds to the application. The primary node replicates data to the secondary node synchronously. Unavailability of the secondary node will affect the operation on the primary node. Data will remain consistent even when the primary node is unavailable.
- Semi-synchronous replication (Semi-Sync): Data is typically replicated in Sync mode. When trying to replicate data to the secondary node, if an exception

occurs causing the primary and secondary nodes to be unable to communicat e with each other, the primary node will suspend response to the application. If the connection cannot be restored, the primary node will degrade to Async mode and restore response to the application after the Sync replication times out . In a situation such as this, the primary node becoming unavailable will lead to data inconsistency. After the secondary node or network connection is recovered , data replication between the two nodes is resumed, and the data replication mode will change from Async to Sync.

You can select different combinations of service priorities and data replication modes to improve availability based on the business features.

3.4.3 Backup and recovery service

This service supports data backup, storage, and recovery functions.

ApsaraDB for RDS can back up databases at any time and restore them to any point in time based on the backup policy, making the data more traceable.

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 stores the backup files to a more cost-effective and persistent Archive Storage system. When the secondary node is operating properly, backup is always initiated on the secondary node. This will not affect the services on the primary node. When the secondary node is unavailable or damaged, the Backup module initiates backup on the primary node.

Recovery

The Recovery module restores backup files stored on OSS to a destination node.

- 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, stores, 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 store backup files from OSS to Archive Storage for more cost-effective and longer-term offline storage.

3.4.4 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 module monitors the connectivity between ECS and RDS and between physical RDS servers, as well as the 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**

3.4.5 Scheduling service

The Resource module implements the scheduling of resources and services.

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 the traffic to and from the instance. This module also allocates and integrates the underlying resources required to migrate RDS instances. 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.

3.4.6 Migration service

RDS provides Data Transmission Service (DTS) to help you migrate databases quickly.

The migration service helps you migrate data from the on-premises database to ApsaraDB for RDS, or migrate data from an instance to another instance in ApsaraDB for RDS.

DTS

DTS enables data migration from on-premises databases to RDS instances or between different RDS instances.

DTS provides three migration methods: schema migration, full migration, and incremental migration.

- Schema migration

  DTS migrates the schema definitions of migration objects to the destination instance. Tables, views, triggers, stored procedures, and stored functions can be migrated in this mode.
- Full migration

  DTS migrates all data of migration objects from the source database to the destination instance.

  ! Notice:

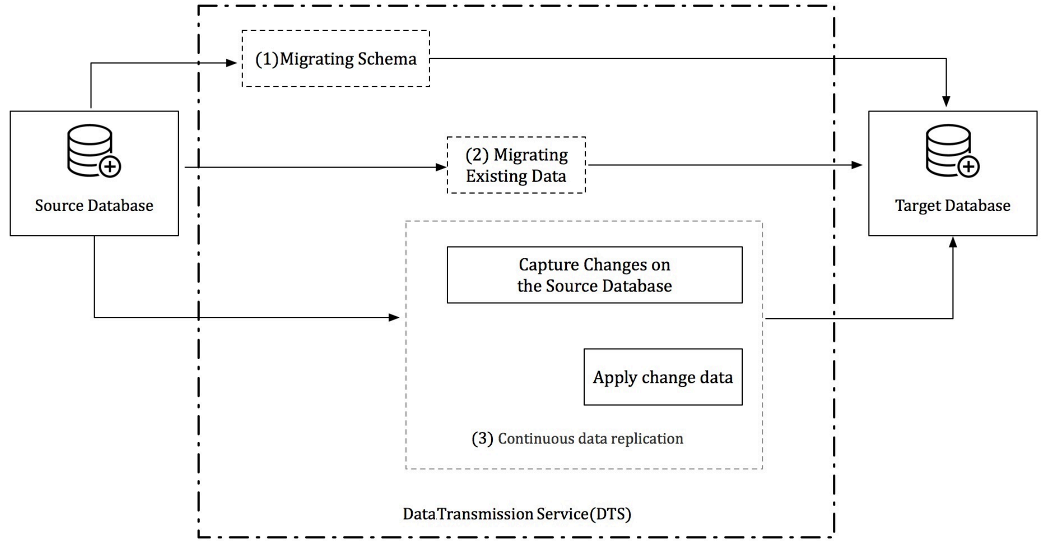
To ensure data consistency, non-transaction tables that do not have primary keys will be locked when performing a full migration. Locked tables cannot be written to. The lock duration depends on the amount of data in the tables. The tables will be unlocked only after they are fully migrated.

- **Incremental migration**

  DTS synchronizes data changes made in the migration process to the destination instance.

Notice:

If a DDL operation is performed during data migration, schema changes will not be synchronized to the destination instance.


(1)Migrating Schema
Source Database
(2) Migrating Existing Data
Target Database
Capture Changes on the Source Database
Apply change data
(3) Continuous data replication
DataTransmission Service(DTS)

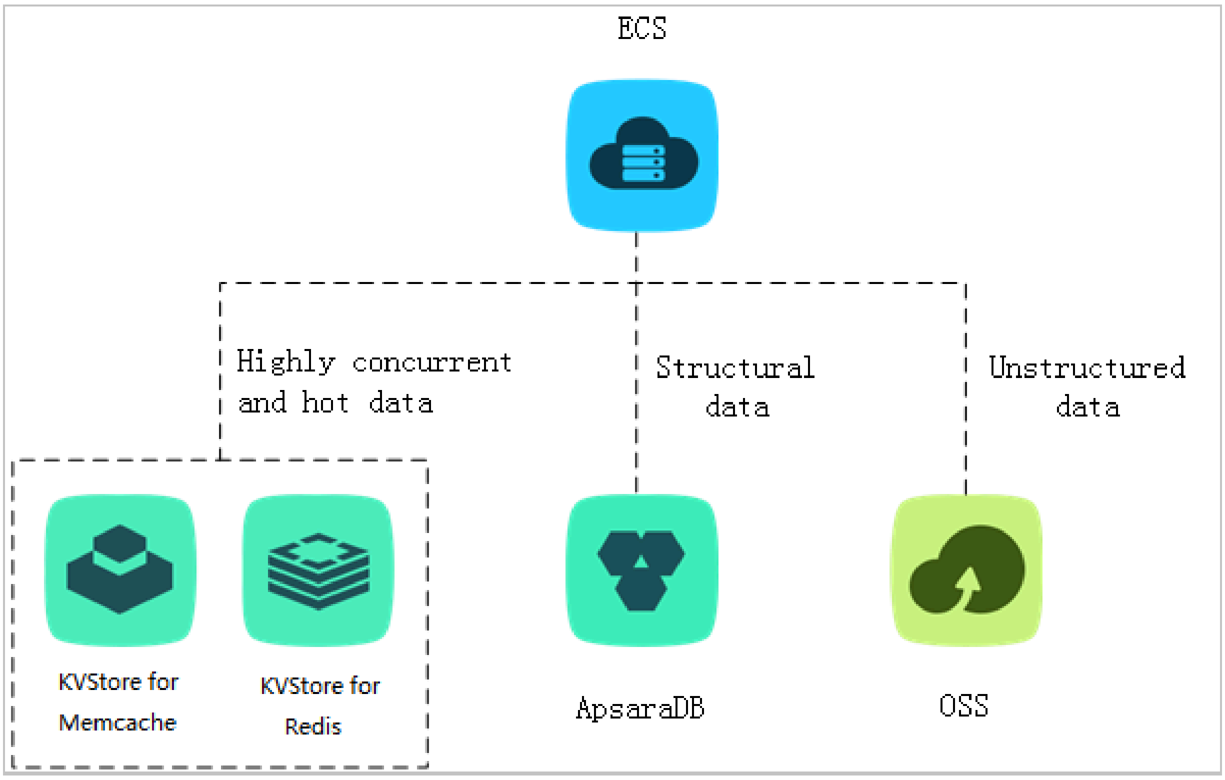
3.5 Scenarios

3.5.1 Diversified data storage

ApsaraDB for RDS provides cache data persistence and multi-structure data storage.

You can diversify the storage capabilities of ApsaraDB for RDS through services such as KVStore for Memcache, KVStore for Redis, and OSS, as shown in *Figure 3-2: Diversified data storage***.**

Figure 3-2: Diversified data storage



Cache data persistence

ApsaraDB for RDS can be used with KVStore for Memcache and KVStore for Redis to form a high-throughput and low-latency storage solution. These cache services have the following benefits over ApsaraDB for RDS:

- **High response speed: The request latency of KVStore for Memcache and KVStore for Redis is usually within just a few milliseconds.**
- **The cache area supports a higher number of queries per second (QPS) than ApsaraDB for RDS.**

Multi-structure data storage

OSS is a secure, reliable, low-cost, and high-capacity storage service from Alibaba Cloud. ApsaraDB for RDS can be used with OSS to implement a multi-type data storage solution. For example, ApsaraDB for RDS and OSS are used together to implement an online forum. Resources such as the images of registered users and posts on the forum can be stored in OSS to reduce storage needs on ApsaraDB for RDS.

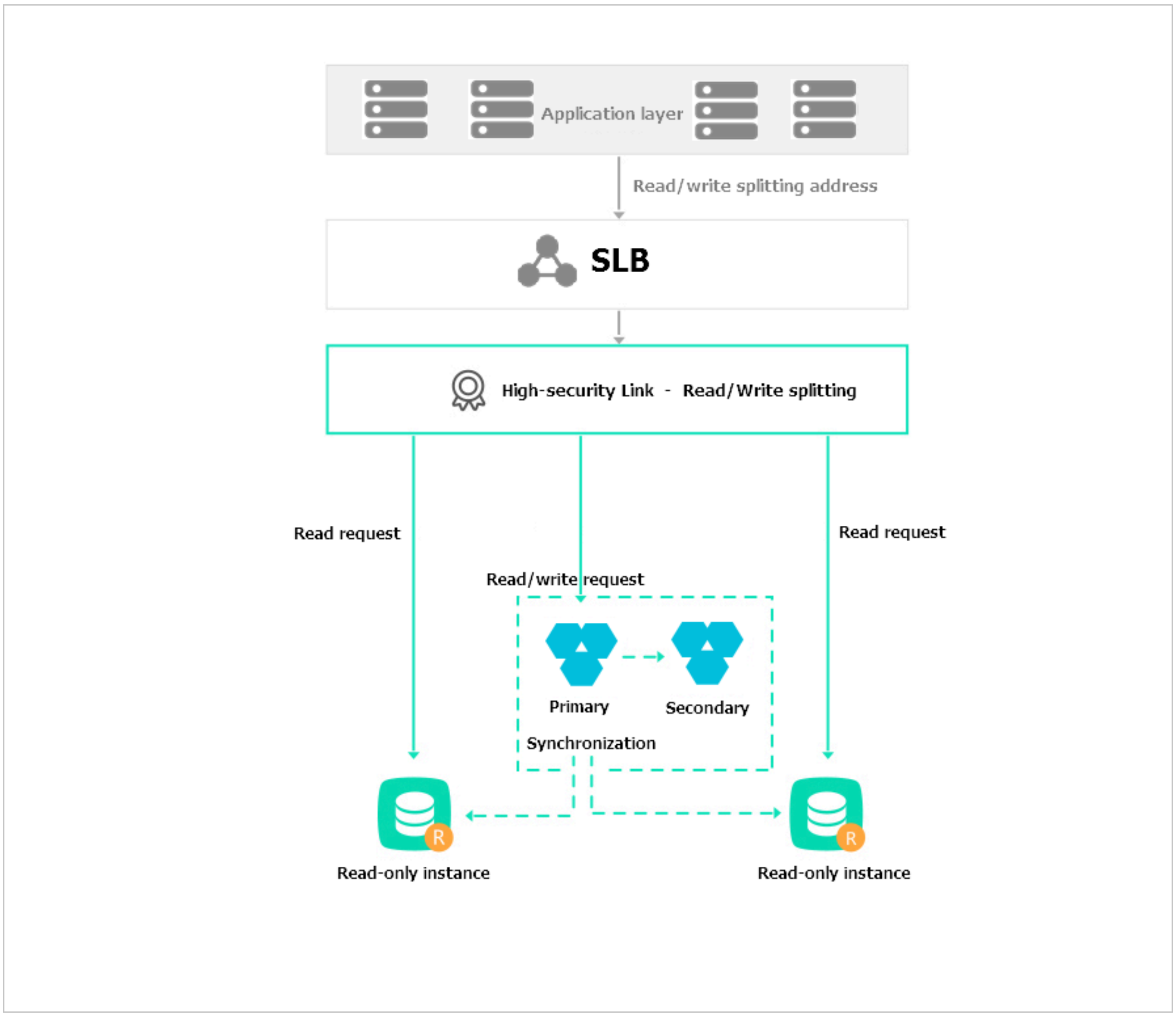
3.5.2 Read/write splitting

This feature allows you to split read requests and write requests across different instances to expand the processing capability of the system.

ApsaraDB RDS for MySQL allows you to directly attach read-only instances to ApsaraDB for RDS to reduce read pressure on the primary instance. The primary instance and read-only instances of ApsaraDB RDS for MySQL each have their own connection endpoints. The system also offers an extra read/write splitting endpoint after read/write splitting is enabled. This endpoint associates the primary instance with all of its read-only instances for automatic read/write splitting, allowing applications to send all read and write requests to a single endpoint. Write requests are automatically routed to the primary instance, and read requests are routed to each read-only instance based on their weights. You can scale out the processing

capability of the system by adding more read-only instances. There is no need to modify applications, as shown in *Read/write splitting*.

Figure 3-3: Read/write splitting

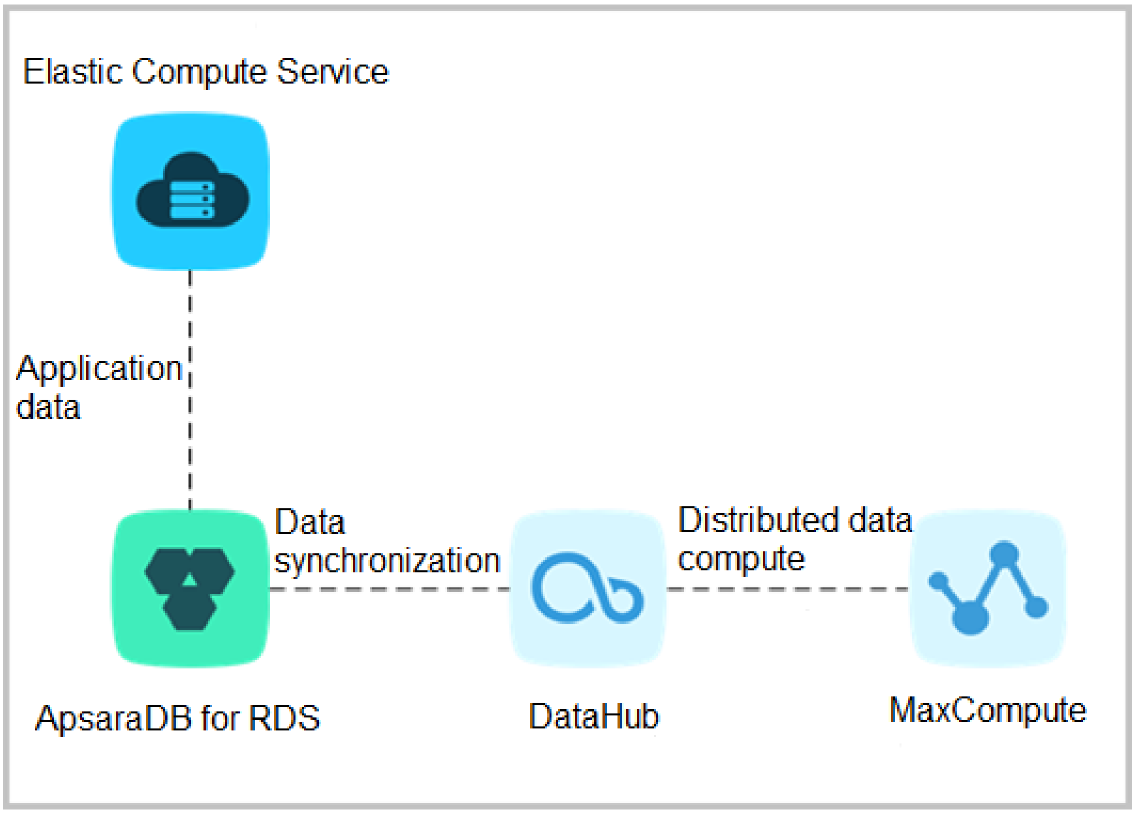


3.5.3 Big data analysis

You can import data from RDS to MaxCompute to enable large-scale data computing.

MaxCompute is used to store and compute batches of structured data. It provides various data warehouse solutions as well as big data analysis and modeling services, as shown in *Big data analysis diagram***.**

Figure 3-4: Big data analysis diagram



3.6 Limits

Before you use ApsaraDB RDS for MySQL, you must understand its limits and take precautions.

To guarantee instance stability and security, ApsaraDB RDS for MySQL has some service limits, as listed in *Table 3-1: Limits on ApsaraDB RDS for MySQL***.**

Table 3-1: Limits on ApsaraDB RDS for MySQL

Operation	Description
Database parameter modification	Database parameters can only be modified through the RDS console or API operations. Due to security and stability considerations, only specific parameters can be modified.
Root permissions of databases	The root and SA permissions are not provided.
Database backup	· Logical backup can be performed through the command line interface (CLI) or graphical user interface (GUI). · Physical backup can only be performed through the RDS console or API operations.
Database restoration	· Logical restoration can be performed through the CLI or GUI. · Physical restoration can only be performed through the RDS console or API operations.
Data import	· Logical import can be performed through the CLI or GUI. · Data can be imported through the MySQL CLI or DTS.
ApsaraDB RDS for MySQL storage engine	· Only InnoDB and TokuDB are supported. Due to the inherent shortcomings of the MyISAM engine, some data may be lost. Only some existing instances use the MylSAM engine. MyISAM engine tables in newly created instances will be automatically converted to InnoDB engine tables. · For safety performance and security considerations, we recommend that you use the InnoDB storage engine. · The Memory engine is not supported. Newly created Memory tables will be automatically converted into InnoDB tables.
Database replication	ApsaraDB RDS for MySQL provides dual-node clusters based on a primary/secondary replication architecture. The secondary instances in this replication architecture are hidden and cannot be accessed directly.
RDS instance restart	Instances must be restarted through the RDS console or API operations.
Account and database management	ApsaraDB RDS for MySQL uses the RDS console to manage accounts and databases. ApsaraDB RDS for MySQL also allows you to create a privileged account to manage users, passwords, and databases.

Operation	Description
Standard account	- Custom authorization is not supported. - The account management and database management interfaces are provided in the RDS console. - Instances that support standard accounts also support privileged accounts.
Privileged account	- Custom authorization is supported. - The RDS console does not provide interfaces to manage accounts or databases. Relevant operations can only be performed through code or DMS. - The privileged account cannot be reverted back to a standard account.

3.7 Terms

Term	Description
region	The geographical location where the server of your RDS instance resides. You must specify a region when you create an RDS instance . The region of an instance cannot be changed after instance creation. RDS must be used together with ECS and only supports internal access. Because of this, RDS instances must be located in the same region as their corresponding ECS instances.
zone	The physical area with an independent power supply and network in a region. Zones in a region can communicate through the internal network. Network latency for resources within the same zone is lower than for those across zones. Faults are isolated between zones. Single zone refers to the case where the three nodes in the RDS instance replica set are all located in the same zone. Network latency is reduced if an ECS instance and its corresponding RDS instance are both deployed in the same zone.
instance	The most basic unit of RDS. An instance is the operating environment of ApsaraDB for RDS and works as an independent process on a host. You can create, modify, or delete an RDS instance in the RDS console. Instances are mutually independent and their resources are isolated. They do not compete for resources such as CPU, memory, or I/O. Each instance has its own features, such as database type and version. RDS controls instance behavior by using corresponding parameters.
memory	The maximum amount of memory that can be used by an ApsaraDB for RDS instance.

Term	Description
disk capacity	The amount of disk space selected when creating an ApsaraDB for RDS instance. Instance data that occupies disk space includes aggregated data as well as data required for normal instance operations such as system databases, database rollback logs, redo logs, and indexing. Ensure that the disk capacity is sufficient for the RDS instance to store data. Otherwise, the RDS may be locked. If the instance is locked due to insufficient disk capacity, you can unlock the instance by expanding the disk capacity.
IOPS	The maximum number of read/write operations performed per second on block devices at a granularity of 4 KB.
CPU core	The maximum computing capability of the instance. A single Intel Xeon series CPU core has at least 2.3 GHz of computational power with hyper-threading capabilities.
number of connections	The number of TCP connections between a client and an RDS instance. If the client uses a connection pool, the connection between the client and RDS instance is a persistent connection. Otherwise, it is a short-lived connection.

4 AnalyticDB for PostgreSQL

4.1 What is AnalyticDB for PostgreSQL?

AnalyticDB for PostgreSQL (formerly known as HybridDB for PostgreSQL) is a distributed analytic database that adopts a massive parallel process (MPP) architecture and consists of multiple compute nodes. AnalyticDB for PostgreSQL provides MPP warehousing services and supports horizontal scaling of storage and compute capabilities, online analysis for petabyte levels of data, and offline extract, transform, and load (ETL) task processing.

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.
- Uses the horizontally scalable MPP architecture and supports range and list partitioning.
- Supports row store, column store, and multiple indexes. It 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 optimizer to ensure high-performance SQL analysis capabilities.
- Supports the primary/secondary architecture to ensure dual-copy data storage.
- Provides online scaling, monitoring, and disaster recovery to reduce O&M costs.

4.2 Benefits

Real-time analysis	Built on the MPP architecture for horizontal scaling and PB/s data processing. AnalyticDB for PostgreSQL supports the leading vector computing feature and intelligent indexes of column store. It also supports the CASCADE-based SQL optimizer to make complex queries without the need for tuning.

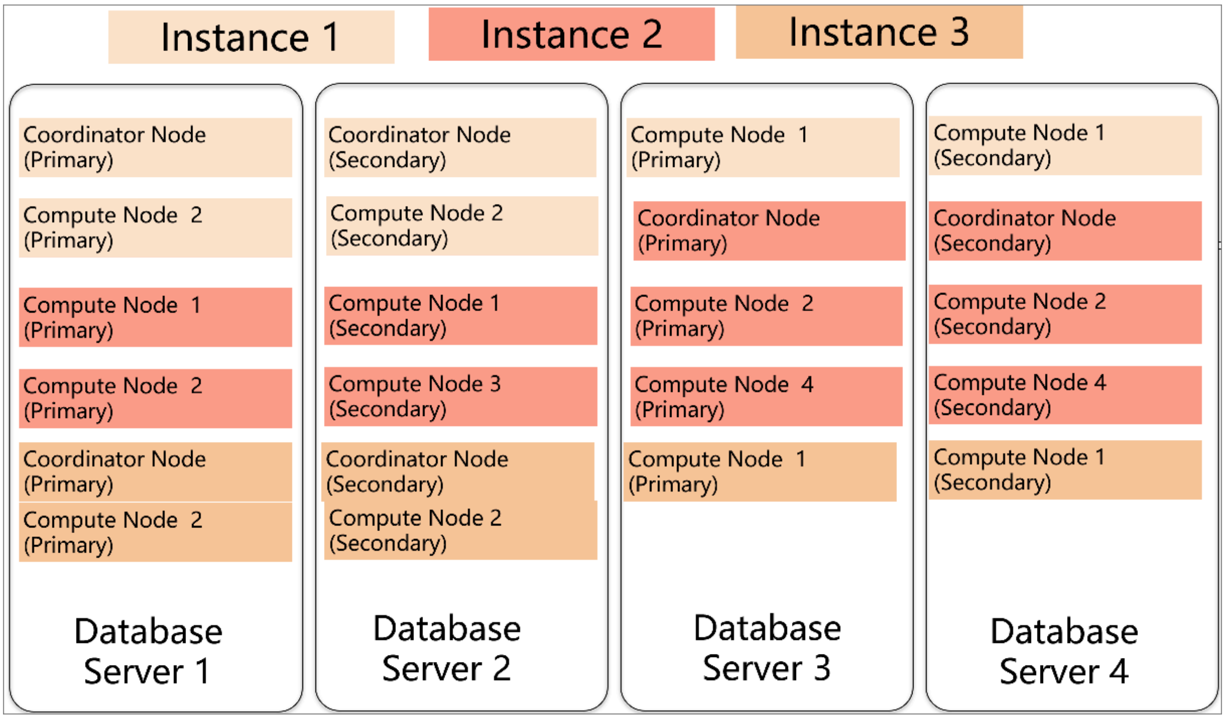
Stability and reliability	Provides ACID properties for distributed transactions. Transactions are consistent across nodes and all data is synchronized between primary and secondary nodes. AnalyticDB for PostgreSQL supports distributed deployment and provides transparent monitoring, switching, and restoration to secure your data infrastructure.
Easy to use	Supports a large number of SQL syntax and functions, Oracle functions, stored procedures, user-defined functions (UDFs), and isolation levels of transactions and databases. You can use popular 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 optimizer enables complex queries without the need for tuning. It supports high-performance parallel import of data from OSS.
Scalability	Enables you to scale up compute nodes, 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.

4.3 Architecture

Physical cluster architecture

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

Figure 4-1: Physical cluster architecture


Instance 1
Instance 2
Instance 3
Coordinator Node (Primary)
Compute Node 2 (Primary)
Compute Node 1 (Primary)
Compute Node 2 (Primary)
Coordinator Node (Primary)
Compute Node 2 (Primary)
Database Server 1
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Compute Node 1 (Secondary)
Compute Node 3 (Secondary)
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Database Server 2
Compute Node 1 (Primary)
Coordinator Node (Primary)
Compute Node 2 (Primary)
Compute Node 4 (Primary)
Compute Node 1 (Primary)
Database Server 3
Compute Node 1 (Secondary)
Coordinator Node (Secondary)
Compute Node 2 (Secondary)
Compute Node 4 (Secondary)
Compute Node 1 (Secondary)
Database Server 4

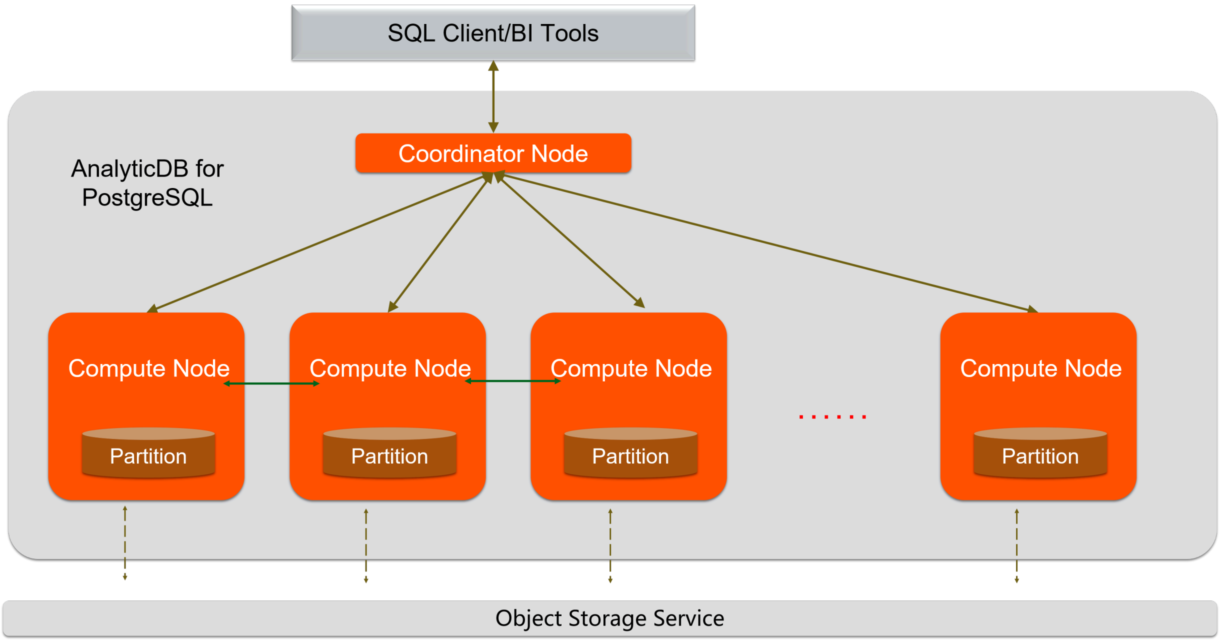
You can create multiple instances within a physical cluster of AnalyticDB for PostgreSQL. Each cluster includes two components: the coordinator node and the compute node.

- **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 consists of a primary node and a secondary node for automatic failover.**

Logical architecture of an instance

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

Figure 4-2: Logical architecture of an instance



Data is distributed across compute nodes by hash value or RANDOM function of a specified distributed column. Each compute node consists of a primary node and a secondary node to ensure dual-copy storage. High-performance network communication is supported across nodes. When the coordinator node receives a request from the 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 will perform an MPP execution of the plan.

4.4 Specification description

AnalyticDB for PostgreSQL supports two storage types: SSD storage and HDD storage. These storage types provide different features that are better suited for different scenarios.

- **SSD storage: provides better I/O capabilities and higher analysis performance.**
- **HDD storage: provides larger and more affordable space to meet higher storage requirements.**

Specifications

AnalyticDB for PostgreSQL supports the following compute node specifications:

Storage type	Core	Memory	Valid storage space	Total dual-copy space	Description
High-performance SSD	1	8 GB	80 GB	160 GB	This storage type is applicable to scenarios where the number of concurrent queries is less than five and the number of nodes of an instance is less than 32. This SSD configuration allows you to create an instance consisting of 4 to 32 nodes.
High-performance SSD	4	32 GB	320 GB	640 GB	The recommended type of high-performance SSD storage. This SSD configuration allows you to create an instance consisting of 8 to 2,048 nodes.
High-capacity HDD	2	16 GB	1 TB	2 TB	This storage type is applicable to scenarios where less than five concurrent queries are performed and the number of nodes of an instance is less than 8. This HDD configuration allows you to create an instance consisting of 4 to 32 nodes.

Storage type	Core	Memory	Valid storage space	Total dual-copy space	Description
High-capacity HDD	4	32 GB	2 TB	4 TB	The recommended type of high-capacity HDD storage. This HDD configuration allows you to create an instance consisting of 8 to 2,048 nodes.

4.5 Features

4.5.1 Distributed architecture

AnalyticDB for PostgreSQL is built on MPP architecture. Data is distributed evenly across nodes by hash value or RANDOM function, and is analyzed and computed in parallel. Storage and computing capacities are scaled horizontally as more nodes are added 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.

4.5.2 High-performance data analysis

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

AnalyticDB for PostgreSQL adopts the CASCADE-based SQL 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.

4.5.3 High-availability service

AnalyticDB for PostgreSQL builds a system based on the Apsara system of Alibaba Cloud for automatic monitoring, diagnostics, and error handling to 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 adopts a primary/secondary architecture to ensure strong consistency of metadata. If the primary coordinator node fails, the service will be automatically switched to the secondary coordinator node.

All compute nodes adopt a primary/secondary architecture to ensure strong data consistency between primary and secondary nodes when data is written into or updated. If the primary compute node fails, the service will be automatically switched to the secondary compute node.

4.5.4 Data synchronization and tools

You can use Data Transmission Service (DTS) or DataWorks to synchronize data from MySQL or PostgreSQL databases to AnalyticDB for PostgreSQL. Popular extract, transform, and load (ETL) tools can import ETL data and schedule jobs on AnalyticDB for PostgreSQL databases. You can also use standard SQL syntax to query data from formatted files stored in OSS by using external tables in real time.

AnalyticDB for PostgreSQL supports Business Intelligence (BI) reporting tools including Quick BI, DataV, Tableau, and FineReport. It also supports ETL tools, including Informatica and Kettle.

4.5.5 Data security

AnalyticDB for PostgreSQL supports IP whitelist configuration. You can add up to 1,000 IP addresses of servers to the whitelist to allow access to your instance and control risks from access sources. AnalyticDB for PostgreSQL also supports Anti-DDoS that monitors inbound traffic in real time. When large amounts of malicious traffic is identified, the traffic is scrubbed through IP filtering. If traffic scrubbing is not sufficient, the black hole process will be triggered.

4.5.6 Supported SQL features

- Supports row store and column store.

- Supports multiple indexes, including the B-tree index, bitmap index, and hash index.
- 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.

Parameter	Alias	Storage size	Range	Description
bigint	int8	8 bytes	-9223372036 854775808 to 9223372036 854775807	Large-range integer
bigserial	serial8	8 bytes	1 to 9223372036 854775807	Large auto-increment integer
bit [(n)]	N/A	n bits	Bit string constant	Fixed-length bit string
bit varying [(n)]	varbit	Variable-length bit string	Bit string constant	Variable-length bit string
boolean	bool	1 byte	true/false, t/f, yes/no, y/n, 1/0	Boolean value (true/false)
box	N/A	32 bytes	((x1,y1),(x2,y2))	A rectangular box on a plane , not allowed in distribution key columns
bytea	N/A	1 byte + binary string	Sequence of octets	Variable-length binary string

Parameter	Alias	Storage size	Range	Description
character [(n)]	char [(n)]	1 byte + n	String up to n characters in length	Fixed-length, blank-padded string
character varying [(n)]	varchar [(n)]	1 byte + string size	String up to n characters in length	Variable length with limit
cidr	N/A	12 or 24 bytes	N/A	IPv4 and IPv6 networks
circle	N/A	24 bytes	<(x,y),r> (center and radius)	A circle on a plane, not allowed in distribution key columns
date	N/A	4 bytes	4,713 BC to 294,277 AD	Calendar date (year, month, day)
decimal [(p, s)]	numeric [(p, s)]	variable	No limit	User-specified precision, exact
double precision	float8 float	8 bytes	Precise to 15 decimal digits	Variable precision, inexact
inet	N/A	12 or 24 bytes	N/A	IPv4 and IPv6 hosts and networks
Integer	int or int4	4 bytes	-2.1E+09 to + 2147483647	Typical choice for integer
interval [(p)]	N/A	12 bytes	-178000000 years to 178000000 years	Time span
json	N/A	1 byte + JSON size	JSON string	Unlimited variable length

Parameter	Alias	Storage size	Range	Description
lseg	N/A	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane, not allowed in distribution key columns
macaddr	N/A	6 bytes	N/A	Media Access Control (MAC) address
money	N/A	8 bytes	-92233720368547758.08 to +92233720368547758.07	Currency amount
path	N/A	16+16n bytes	[(x1,y1),...]	A geometric path on a plane, not allowed in distribution key columns
point	N/A	16 bytes	(x,y)	A geometric point on a plane, not allowed in distribution key columns
polygon	N/A	40+16n bytes	((x1,y1),...)	A closed geometric path on a plane, not allowed in distribution key columns
real	float4	4 bytes	Precise to 6 decimal digits	Variable precision, inexact
serial	serial4	4 bytes	1 to 2147483647	Auto-increment integer
smallint	int2	2 bytes	-32768 to 32767	Small-range integer

Parameter	Alias	Storage size	Range	Description
text	N/A	1 byte + string size	Variable-length string	Unlimited variable length
time [(p)] [without time zone]	N/A	8 bytes	00:00:00[.000000] to 24:00:00[.000000]	Time of day (without time zone)
time [(p)] with time zone	timetz	12 bytes	00:00:00+1359 to 24:00:00-1359	Time of day (with time zone)
timestamp [(p)] [without time zone]	N/A	8 bytes	4,713 BC to 294,277 AD	Date and time
timestamp with time zone	timestamptz	8 bytes	4,713 BC to 294,277 AD	Date and time (with time zone)
xml	N/A	1 byte + XML size	Variable-length XML string	Unlimited variable length

4.6 Scenarios

AnalyticDB for PostgreSQL is applicable to the following OLAP data analysis services.

- ETL for offline data processing

  AnalyticDB for PostgreSQL has the following features that make it ideal for optimizing complex SQL queries and aggregating and analyzing large amounts of data:

  - Supports standard SQL, OLAP window functions, and stored procedures.
  - Provides the CASCADE-based SQL optimizer to make complex queries without the need for tuning.
  - Built on the MPP architecture for horizontal scaling and PB/s data processing.
  - Provides high performance, column store-based storage and aggregation of tables at a high compression ratio to save storage space.

- **Online high-performance query**

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

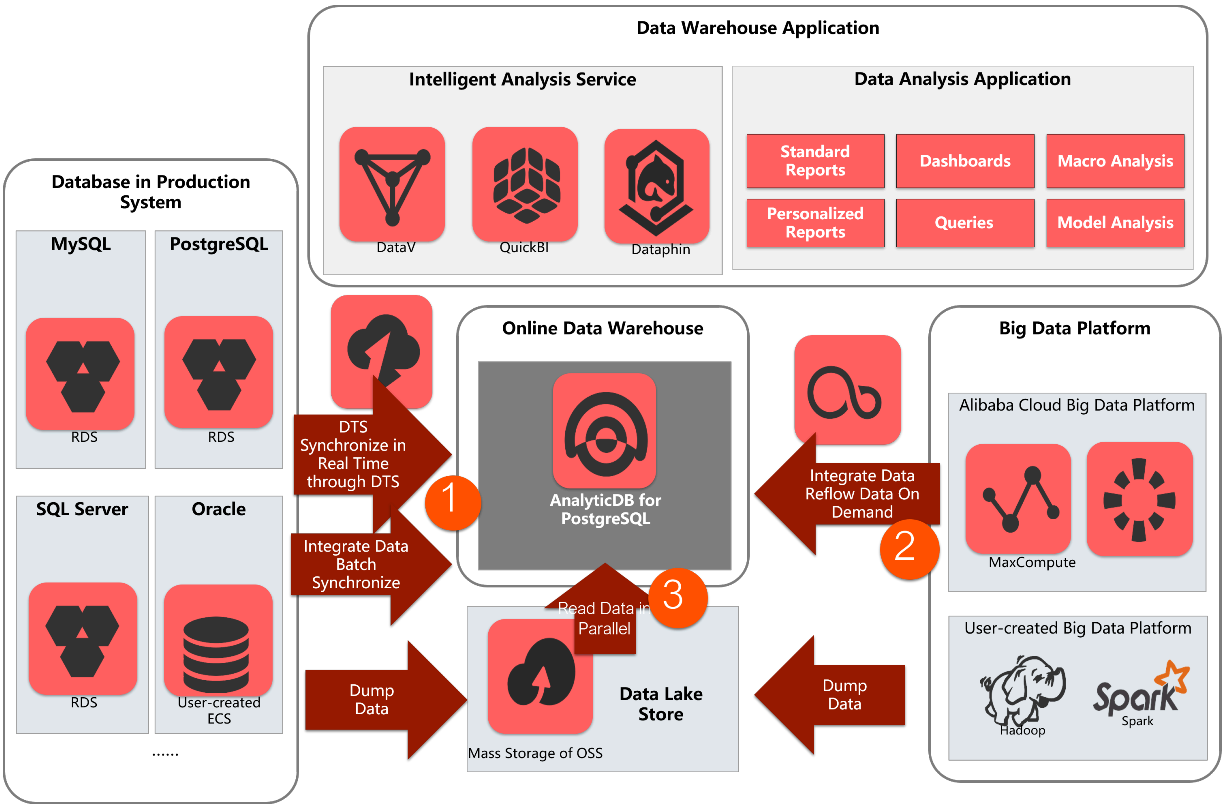
  - **Allows you to write and update high-throughput data through INSERT, UPDATE, and DELETE operations.**
  - **Allows you to query data based on row store and multiple indexes (B-tree, bitmap, and hash) to obtain results in milliseconds.**
  - **Supports distributed transactions, standard database isolation levels, and HTAP.**
- **Multi-model data analysis**

  AnalyticDB for PostgreSQL provides the following benefits for processing of a variety of unstructured data 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 through 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 three 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, Apsara PolarDB, and traditional databases such as Oracle and SQL Server. Data can also be synchronized in batches to AnalyticDB for PostgreSQL through the data integration service (DataX). AnalyticDB for PostgreSQL supports complex extract, transform, and load (ETL) operations on large amounts of data. These tasks can also be scheduled by Dataworks. 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 import huge amounts of data from MaxCompute, Hadoop, and Spark to AnalyticDB for PostgreSQL through DataX or OSS for high-performance analysis, processing, and exploration.

- **Data lake analytics**

  AnalyticDB for PostgreSQL can use an external table mechanism to access the huge amounts of data stored in OSS in parallel and build an Alibaba Cloud data lake analytics platform.

4.7 Introduction to version 6.0

AnalyticDB for PostgreSQL 6.0 is an online database service based on the open-source Greenplum Database kernel 6.0. AnalyticDB for PostgreSQL 6.0 improves the capabilities to process concurrent transactions for real-time data warehouses.

Kernel upgrade

The Greenplum kernel has been upgraded from version 4.3 to version 6.0, and the AnalyticDB for PostgreSQL kernel has been upgraded from version 8.2 to version 9.4 . The new features are as follows:

- **JSONB: supports the JSON and JSONB storage types to implement high-performance JSON data processing and provide more JSON functions.**
- **UUID: supports the UUID data type.**
- **GIN and SP-GiST indexes: provide higher-performance fuzzy matching and Chinese retrieval.**
- **Fine-grained permission control: supports schema-level and column-level permission control and authorization.**
- **Efficient VACUUM statements: When you execute VACUUM statements to release space, locked pages will be skipped and vacuumed at a later time to reduce blocking.**
- **DBLINK: executes remote database queries.**
- **Recursive common table expression (CTE): processes hierarchical or tree-structured data to facilitate multi-level recursive queries.**
- **PL/SQL enhanced:**
  - **Supports the RETURN QUERY EXECUTE statement to execute SQL statements dynamically.**
  - **Supports anonymous blocks.**

Hybrid Transaction/Analytical Processing (HTAP) capabilities enhanced

AnalyticDB for PostgreSQL 6.0 has introduced the global deadlock detection mechanism to dynamically collect and analyze lock information for global deadlock checking and unlocking. Updates and modifications to heap tables can only be completed with fine-grained row locks. AnalyticDB for PostgreSQL 6.0 supports concurrent update, delete, and query operations to improve the concurrency and throughput of the system. AnalyticDB for PostgreSQL 6.0 optimizes transaction locks to reduce lock competition at the beginning and end of transactions. AnalyticDB for PostgreSQL 6.0 provides high-performance OLAP analysis and high-throughput transaction processing features.

New OLAP features

- **Replicated table: provides the DISTRIBUTED REPLICATED clause to create replicated tables for dimension tables in the data warehouse. This reduces data transmission and improves query efficiency.**
- **Zstandard data compression algorithm: provides three times the performance of the zlib algorithm.**

5 Data Transmission Service (DTS)

5.1 What is DTS?

Data Transmission Service (DTS) is a data service provided by Alibaba Cloud. DTS supports data transmission between various types of data sources, such as relational databases.

DTS provides data transmission capabilities such as data migration and change tracking. DTS can be used in many scenarios, such as interruption-free data migration, geo-disaster recovery, cross-border data synchronization, and cache updates. DTS helps you build a data architecture that features high availability, scalability, and security.

- **DTS allows you to simplify data transmission and focus on business development .**
- **DTS supports MySQL as the data source type.**

5.2 Benefits

DTS supports data transmission between data sources such as relational databases and OLAP databases. DTS provides data transmission capabilities such as data migration and change tracking. Compared with other data migration and synchronization tools, DTS provides transmission channels with higher compatibility, performance, security, and reliability. DTS also provides a variety of features to help you create and manage transmission channels.

High compatibility

DTS supports data migration and synchronization between homogeneous and heterogeneous data sources. For migration between heterogeneous data sources, DTS supports schema conversion.

DTS provides data transmission capabilities such as data migration and change tracking. In change tracking, data is transmitted 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 channel.

DTS uses a variety of optimization measures for data migration.

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 channels.

To facilitate channel management, the DTS console shows information about transmission channels, such as transmission status, progress, and performance.

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

5.3 Environment requirements

You must use 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**

You must use the following operating system:

AliOS7U2-x86-64

Notice:

- **Do not use DTS on hosts that are excluded from the preceding models.**
- **The** `/apsara` **directory used by DTS resides on only one hard disk. Make sure that the available space in the directory is larger than 2 TB.**

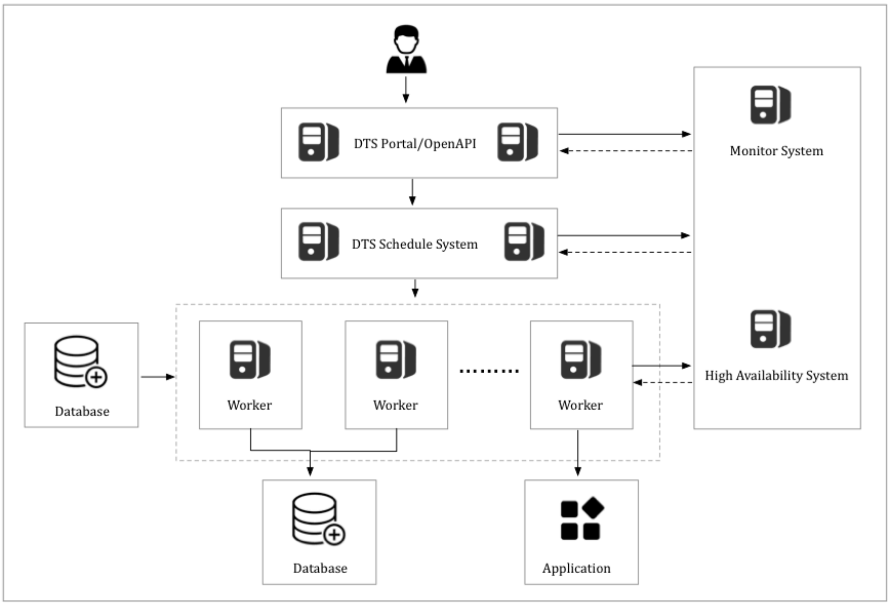
  If the available space in the `/apsara` **directory is less than 2 TB, tasks cannot run as expected and errors will occur. If a task fails, the task recovery and data pulling are affected.**

5.4 Architecture

System architecture

The following figure shows the system architecture of DTS.

Figure 5-1: System architecture



- **High availability**

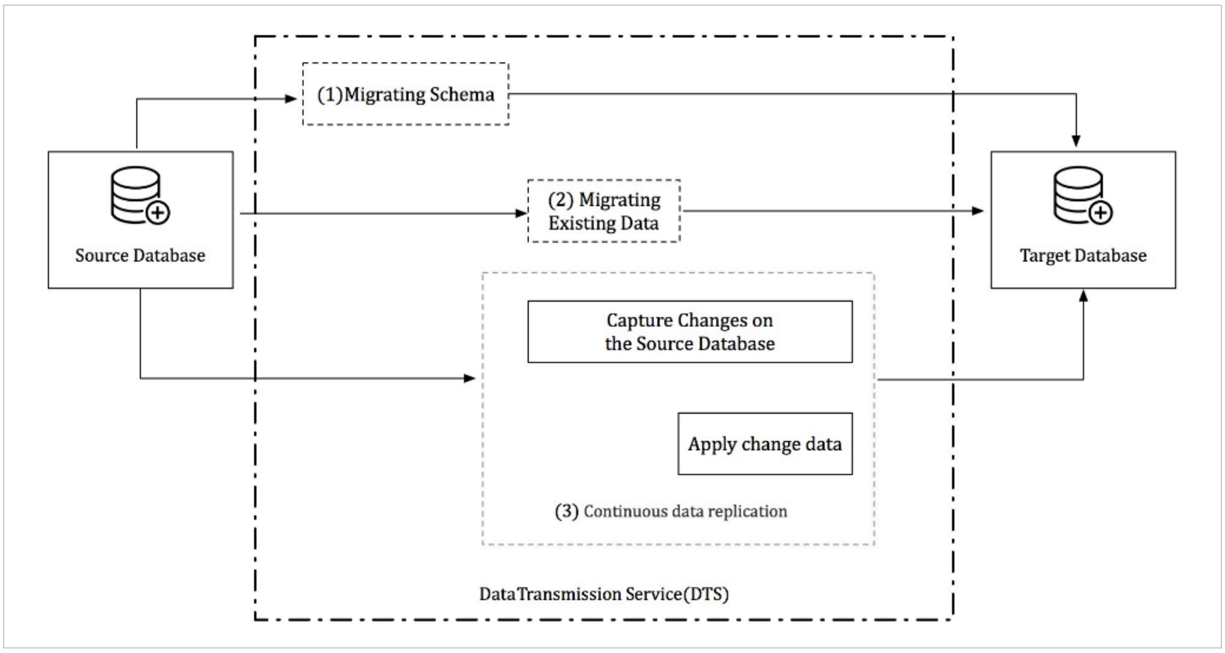
  Each module in DTS has primary and secondary nodes to ensure high availability . The disaster recovery module runs a health check on each node in real time. If a node failure is detected, the module requires only a few seconds to switch the channel to a healthy node.
- **Connection reliability**

  To ensure the connection reliability of change tracking channels, the disaster recovery module checks for configuration changes, such as changes of a data source address. If a data source address is changed, the module allocates a new connection method to ensure the stability of the channel.

Design concept of data migration

The following figure shows the design concept of data migration.

Figure 5-2: Design concept of data migration


(1)Migrating Schema
Source Database
(2) Migrating Existing Data
Target Database
Capture Changes on the Source Database
Apply change data
(3) Continuous data replication
DataTransmission Service(DTS)

Data migration supports schema migration, full data migration, and incremental data migration. The following processes ensure service continuity during data migration:

1. **Schema migration**
2. **Full data migration**
3. **Incremental data migration**

To migrate data between heterogeneous databases, DTS reads the source database schema, translates the schema into the syntax of the destination database, and imports the schema to the destination database.

A full data migration requires a long period of time. During this process, incremental data is continuously written to the source database. To ensure data consistency, DTS starts the incremental data reading module before full data migration. This module retrieves incremental data from the source database, and parses, encapsulates, and locally stores the data.

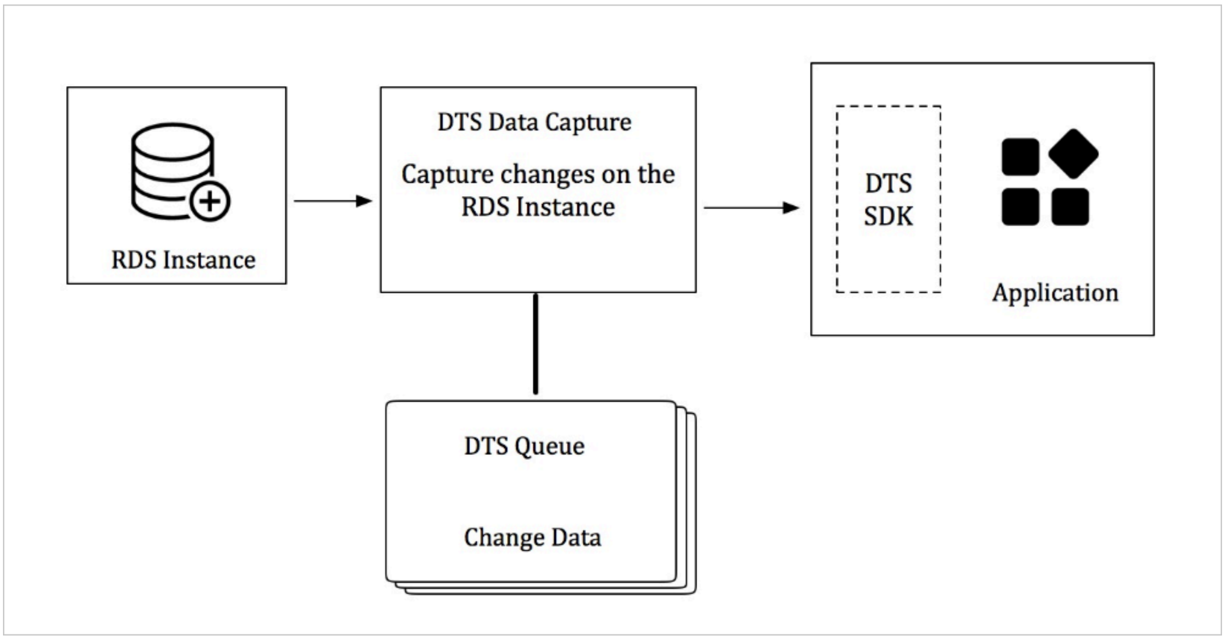
After the full data migration is complete, DTS starts the incremental data loading module. This module retrieves incremental data from the incremental data reading

module. After reverse parsing, filtering, and encapsulation, incremental data is migrated to the destination database in real time.

Design concept of change tracking

The following figure shows the design concept of change tracking.

Figure 5-3: Design concept of change tracking



The change tracking feature allows you to obtain incremental data from an RDS instance in real time. You can subscribe to incremental data on the change tracking server by using DTS SDKs. You can also customize data consumption rules based on your business requirements.

The incremental data reading module on the server side of DTS retrieves raw data from the source instance. After parsing, filtering, and syntax conversion, incremental data is locally stored.

The incremental data reading module connects to the source instance by using a database protocol and retrieves incremental data from the source instance in real time. If the source instance is an ApsaraDB RDS for MySQL instance, the incremental data reading module connects to the source instance by using the binary log dump protocol.

DTS ensures high availability of the incremental data reading module and consumption SDK processes.

If an error is detected in the incremental data reading module, the disaster recovery module restarts the incremental data reading module on a healthy node. This ensures high availability of the incremental data reading module.

DTS ensures high availability of consumption SDK processes on the server. If you start multiple consumption SDK processes for the same change tracking channel, the server pushes incremental data to only one process at a time. If an error occurs on a process, the server pushes data to another healthy consumption process.

5.5 Features

5.5.1 Data migration

You can use DTS to migrate data between various types of data sources. Typical scenarios include data migration to the cloud, data migration between instances within Apsara Stack, and database sharding and scaling. DTS supports data migration between homogeneous and heterogeneous data sources. It also supports extract, transform, and load (ETL) features such as data filtering and object name mapping for databases, tables, and columns.

Data source and data migration types

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

Table 5-1: Data source and data migration types

Data source	Schema migration	Full data migration	Incremental data migration
MySQL database	Supported	Supported	Supported

DTS supports migrating data from the following types of data sources:

- **User-created on-premises database**

DTS supports migrating data to the following types of data sources:

- **User-created on-premises database**

Online migration

DTS uses online migration. You only need to configure the source instance, destination instance, and objects to be migrated. DTS automatically completes the entire data migration process. You can select all of the supported migration types to minimize the impact of online data migration on your services. However, you must ensure that DTS servers can connect to both the source and destination instances.

Data migration types

DTS supports schema migration, full data migration, and incremental data migration.

- **Schema migration: migrates schemas from the source instance to the destination instance.**
- **Full data migration: migrates historical data from the source instance to the destination instance.**
- **Incremental data migration: migrates incremental data that is generated during migration from the source instance to the destination instance in real time. You can select schema migration, full data migration, and incremental migration to ensure service continuity.**

ETL features

Data migration supports the following extract, transform, and load (ETL) features:

- **Object name mapping for databases, tables, and columns: You can migrate data between two databases, tables, or columns that have different names.**
- **Data filtering: You can use SQL conditions to filter the required data in a specific table. For example, you can specify a time range to migrate only the latest data.**

Alerts

If an error occurs during data migration, DTS immediately sends an SMS alert to the task owner. This allows the owner to handle the error at the earliest opportunit y.

Migration task

A migration task is a basic unit of data migration. To migrate data, you must create a migration task in the DTS console. To create a migration task, you must configure the required information such as the source and destination instances

, migration types, and objects to be migrated. You can create, manage, stop, and delete migration tasks in the DTS console.

The following table describes the statuses of a migration task.

Table 5-2: Statuses of a migration task

Status	Description	Available operation
Not Started	The migration task is configured but the precheck is not performed.	· Perform the precheck · Delete the migration task
Prechecking	A precheck is being performed but the migration task is not started.	Delete the migration task
Passed	The migration task has passed the precheck but has not been started.	· Start the migration task · Delete the migration task
Migrating	Data is being migrated.	· Pause the migration task · Stop the migration task · Delete the migration task

Status	Description	Available operation
Migration Failed	An error occurred during migration. You can identify the point of failure based on the progress of the migration task.	Delete the migration task
Paused	The migration task is paused.	· Start the migration task · Delete the migration task
Completed	The migration task is completed, or you have stopped data migration by clicking End.	Delete the migration task

5.5.2 Change tracking

You can use DTS to retrieve incremental data from user-created MySQL databases in real time. This feature applies to the following scenarios: cache updates, business decoupling, asynchronous data processing, and real-time synchronization of heterogeneous data and extract, transform, and load (ETL) operations.

Features

You can use DTS to retrieve incremental data from user-created MySQL databases.

Data sources

The change tracking feature supports the following types of data source:

- MySQL database

Objects for change tracking

The objects for change tracking include tables and databases. You can specify one or more tables from which you want to track data changes.

In change tracking, incremental data includes data manipulation language (DML) operations and data definition language (DDL) operations. When you configure change tracking, you must select the type of operation.

Change tracking channel

A change tracking channel is the basic unit of incremental data tracking and consumption. To subscribe to incremental data of an RDS instance, you must create a change tracking channel in the DTS console for the RDS instance. The change tracking channel pulls incremental data from the RDS instance in real time and locally stores incremental data. You can use the DTS SDK to consume incremental data from the change tracking channel. You can also create, manage, or delete change tracking channels in the DTS console.

A change tracking channel can be consumed by only one downstream SDK client . To subscribe to an RDS instance by using multiple downstream SDK clients, you must create an equivalent number of change tracking channels. The channels pull incremental data from the same RDS instance.

The *Table 5-3: Statuses of a change tracking channel* table describes the statuses of a change tracking channel.

Table 5-3: Statuses of a change tracking channel

Channel status	Description	Available operation
Prechecking	The configuration of the change tracking channel is complete and a precheck is being performed.	Delete the change tracking channel
Not Started	The change tracking channel has passed the precheck but has not been started.	· Start the change tracking channel · Delete the change tracking channel
Performing Initial Change Tracking	The initial change tracking is in progress. This process takes about one minute.	Delete the change tracking channel
Normal	Incremental data is being pulled from the source RDS instance.	· View sample code · View tracked data changes · Delete the change tracking channel

Channel status	Description	Available operation
Error	An error occurs when the change tracking channel is pulling incremental data from the source RDS instance.	· View sample code · Delete the change tracking channel

Advanced features

You can use the following advanced features that are provided for change tracking:

- Add and remove objects for change tracking

  You can add or remove the objects for change tracking.
- View tracked data changes

  You can view the incremental data in the change tracking channel.
- Modify consumption checkpoints

  You can modify consumption checkpoints.
- Monitor the change tracking channel

  You can monitor the status of the change tracking channel and receive an alert if the threshold for downstream consumption is reached. You can set the alert threshold based on the sensitivity of your businesses to consumption latency.

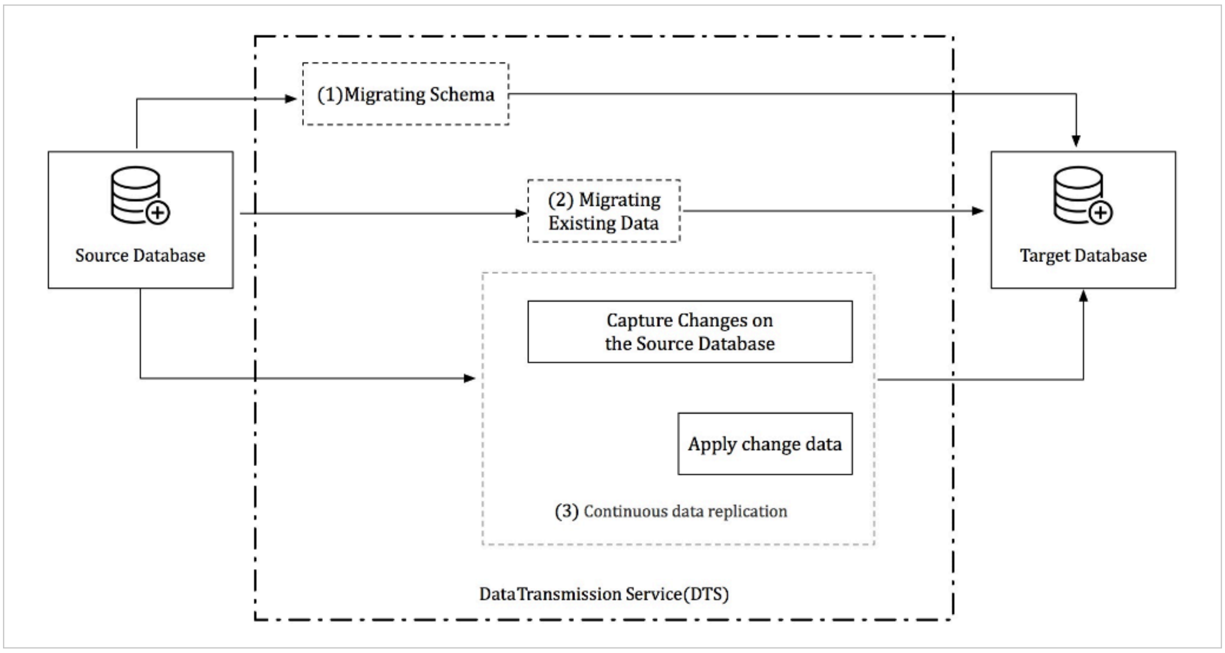
5.6 Scenarios

Data Transmission Service (DTS) provides features such as data migration and change tracking. You can use DTS features in various scenarios.

Database migration with minimized downtime

To ensure data consistency, traditional migration requires that you stop writing data to the source database during data migration. Depending on the data volume and network conditions, the migration may take several hours or even days, which has a great impact on your businesses.

DTS provides migration with minimized downtime. Services are always available except when they are switched from the source instance to the destination instance . The service downtime is minimized to minutes. The following figure shows the architecture of data migration.

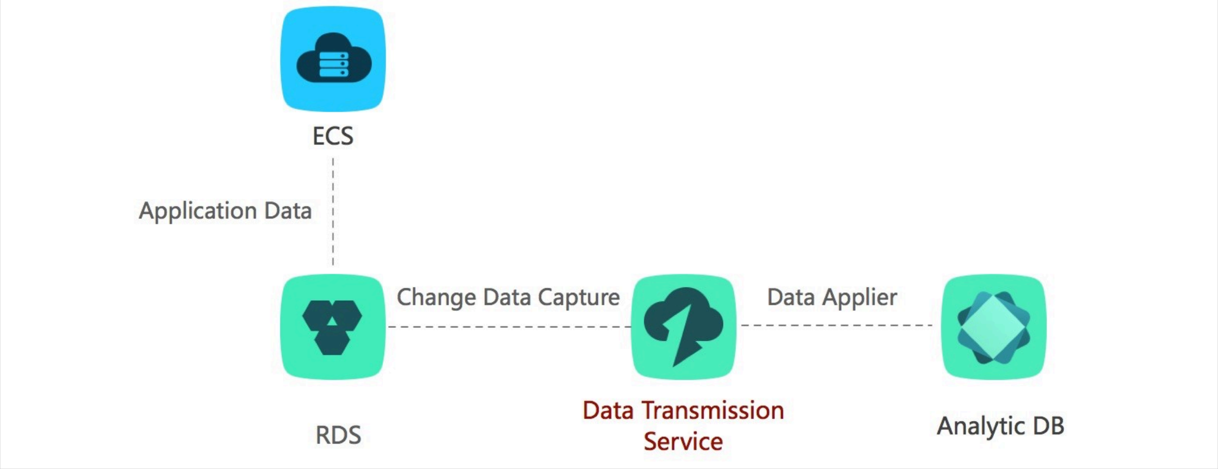

(1)Migrating Schema
Source Database
(2) Migrating Existing Data
Target Database
Capture Changes on the Source Database
Apply change data
(3) Continuous data replication
DataTransmission Service(DTS)

The data migration process includes schema migration, full data migration, and incremental data migration. During incremental data migration, the data in the source instance is synchronized to the destination instance in real time. You can verify businesses in the destination database. After the verification succeeds, you can migrate businesses to the destination database.

Real-time data analysis

Data analysis is essential in improving enterprise insights and user experience. With real-time data analysis, enterprises can adjust marketing strategies to adapt to changing markets and higher demands for better user experience.

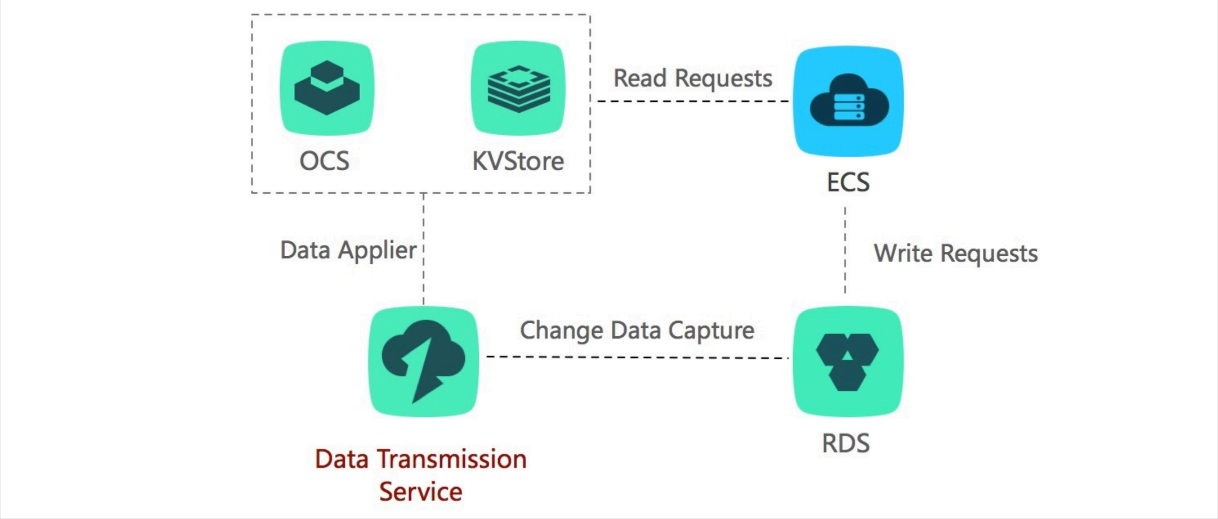
With the change tracking feature provided by DTS, you can acquire real-time incremental data without affecting online businesses. You can use the DTS SDK to synchronize the subscribed incremental data to the analysis system for real-time analysis.



Lightweight cache update policies

To accelerate access speed and improve concurrent read performance, a cache layer is used in the business architecture to receive all read requests. The memory read mechanism of the cache layer can help to improve read performance. The data in the cache memory is not persistent. If the cache memory fails, the data in the cache memory will be lost.

With the change tracking feature provided by DTS, you can subscribe to the incremental data in databases and update the cached data to implement lightweigh t cache update policies.



Benefits

- **Quick update with low latency**

  The business returns data after the database update is complete. For this reason, you do not need to consider the cache invalidation process, and the entire update path is short with low latency.
- **Simple and reliable applications**

  The complex doublewrite logic is not required for the applications. You only need to start the asynchronous thread to monitor the incremental data and update the cached data.
- **Application updates without extra performance consumption**

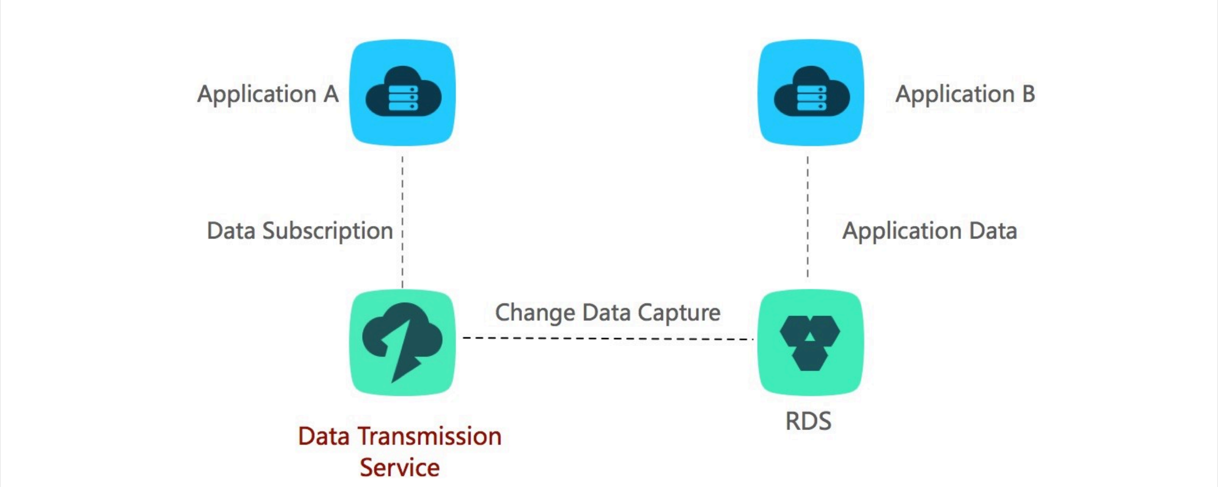
  DTS retrieves incremental data by parsing incremental logs in the database, which does not affect the performance of businesses and databases.

Business decoupling

The e-commerce industry involves many different types of business logic such as ordering, inventory, and logistics. If all of these types of business logic are included in the ordering process, the order result can be returned only after all the changes are complete. However, this may cause the following issues:

- The ordering process consumes a long period of time and results in poor user experience.
- The business system is unstable and downstream faults will affect service availability.

With the change tracking feature provided by DTS, you can optimize your business system and receive notifications in real time. You can decouple different types of business logic and asynchronously process data. This makes the core business logic simpler and more reliable. The following figure shows the architecture of business decoupling.



In this scenario, the ordering system returns the result after the buyer places an order. The underlying layer obtains the data changes that are generated in the ordering system in real time by using the change tracking feature. You can subscribe to these data changes by using the DTS SDK, which triggers different types of downstream business logic such as inventory and logistics. This ensures that the entire business system is simple and reliable.

This scenario has been applied to a wide range of businesses in Alibaba Group. Tens of thousands of downstream businesses in the Taobao ordering system are using the change tracking feature to retrieve real-time data updates and trigger business logic every day.

5.7 Terms

This topic describes the terms that are used in the DTS documentation.

Term	Description
precheck	The system performs a precheck before starting a data migration task or change tracking task. For example, the following items are checked: the connectivity between DTS servers and the source and destination databases, database account permissions, whether binary logging is enabled, and database version numbers. Note: If the precheck fails, click the info icon next to each failed item to view the related details. Troubleshoot the issues based on the cause of failure and run the precheck again.

Term	Description
schema migration	DTS migrates the schemas of the required objects from the source database to the destination database. The schemas of tables, views , triggers, and stored procedures can be migrated. For schema migration between heterogeneous databases, DTS converts the schema syntax based on the syntax of the source and destination databases. For example, DTS converts the NUMBER data type in Oracle databases to the DECIMAL data type in MySQL databases.
full data migration	DTS migrates historical data of the required objects from the source database to the destination database. If you select only schema migration and full data migration, new data that is generated in the source database will not be migrated to the destination database. To ensure data consistency, do not write new data into the source database during full data migration. Note: To migrate data with minimal downtime, you must select schema migration, full data migration, and incremental data migration.
incremental data migration	DTS retrieves static snapshots from the source database and migrates the snapshot data to the destination database. Then, DTS synchronizes incremental data that is generated in the source database to the destination database in real time. Note: During incremental data migration, data between the source and destination databases is synchronized in real time. The migration task does not automatically end. You need to manually end the migration task.
data update	Data updates are operations that modify data without modifying the schema, such as INSERT, DELETE, and UPDATE operations.
schema update	Schema updates are operations that modify the schema syntax, such as CREATE TABLE, ALTER TABLE, and DROP VIEW operations.

Term	Description
timestamp range	The timestamp range is the range of timestamps for incremental data that is stored in a change tracking channel. By default, the change tracking channel retains the data that is generated in the most recent 24 hours. DTS clears expired incremental data on a regular basis and updates the timestamp range of the change tracking channel. The timestamp of incremental data is generated when the data is updated in the source database and written to the transaction log.
consumption checkpoint	The consumption checkpoint is the timestamp of the latest incremental data that is consumed by a downstream SDK client. When the SDK client consumes a data record, it returns a confirmation message to DTS. DTS updates and saves the consumption checkpoint. If the SDK client restarts due to exceptions, DTS pushes incremental data from the last consumption checkpoint.

6 KVStore for Redis

6.1 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. This service allows you to focus on business innovation.

6.2 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.**
- **Dual-copy 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.**

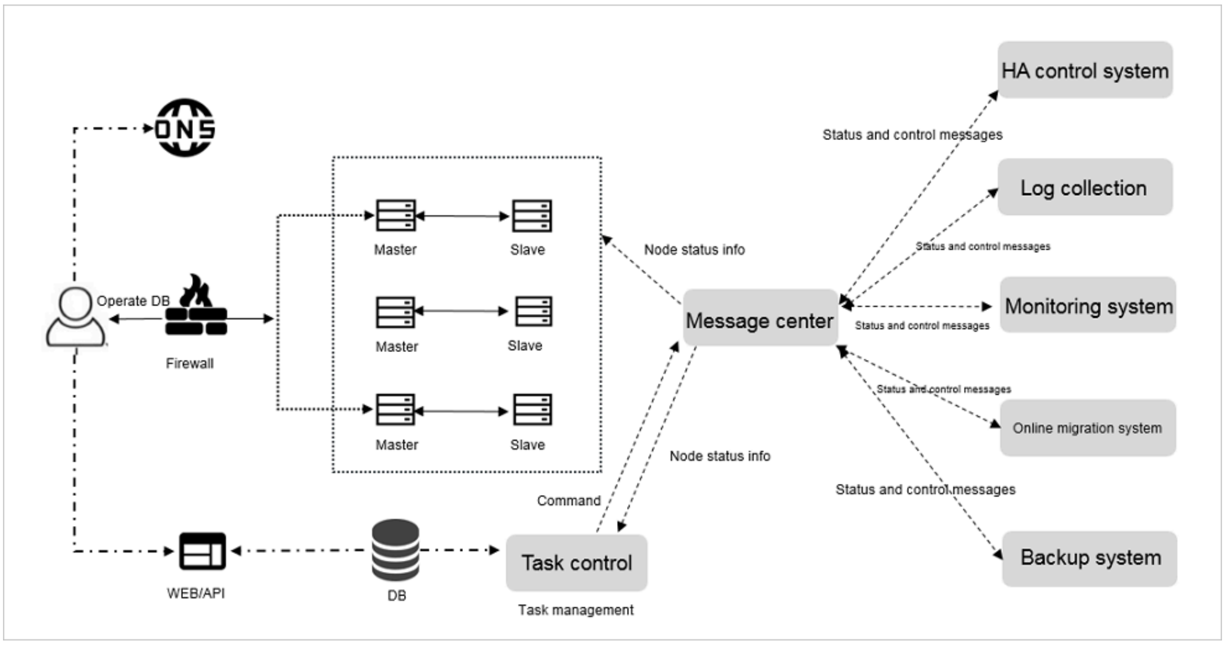
Easy to use

- **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.**

6.3 Architecture

The architecture of KVStore for Redis is as shown in *Figure 6-1: Architecture diagram*.

Figure 6-1: Architecture diagram



KVStore for Redis automatically builds a primary/secondary structure. You can use this structure directly.

- **HA control system**

  A high-availability (HA) detection module is used to detect and monitor the operating status of KVStore for Redis instances. If this module determines that a primary node is unavailable, the module automatically performs the failover operation to ensure high availability of KVStore for Redis instances.

- **Log collection**

  This module collects instance operation logs, including slow query logs and access control logs.

- **Monitoring system**

  This module collects performance monitoring information of KVStore for Redis instances, including basic group monitoring, key group monitoring, and string group monitoring.

- **Online migration system**

  When an error occurs on the physical server that hosts a KVStore for Redis instance, this module recreates an instance on the fly based on the backup files stored in the backup system. This ensures high availability of your business.
- **Backup system**

  This module generates backup files of KVStore for Redis instances, and stores the backup files in Object Storage Service (OSS). The backup system allows you to customize the backup settings, and retains backup files for up to seven days.
- **Task Control**

  KVStore for Redis instances support various management and control tasks, including instance creation, specifications changes, and instance backups. The task system flexibly controls and tracks tasks and manages errors according to your instructions.

6.4 Features

- **High-availability technology ensures service stability**

  The system synchronizes data between the primary node and the secondary node in real time. If the primary node fails, the system automatically performs the failover operation and restores services within a few seconds. The secondary node takes over services. This process does not affect your business, and ensures high availability of system services.

  Cluster instances run in a distributed architecture. Each node uses a primary /secondary high-availability structure to automatically perform failover and disaster recovery and ensure high availability of system services.
- **Easy backup and recovery support custom backup policies**

  You can back up data in the console and customize automatic backup policies . The system automatically retains backup data for seven days. You can easily restore data in the case of accidental data operations to minimize your business losses.
- **Multiple network security protections secure your data**

  A Virtual Private Cloud (VPC) isolates network transmission at the transport layer. The Anti-Distributed-Denial-of-Service (DDoS) protection service monitors and protects against DDoS attacks. The system supports a whitelist that contains a maximum of 1,000 IP addresses or CIDR blocks to prevent malicious login attempts.

- Kernel optimization avoids vulnerability exploits

  The experts of Alibaba Cloud have performed in-depth kernel optimization for the Redis source code to effectively prevent running out of memory, fix security vulnerabilities, and protect your business.
- Elastic scaling eliminates capacity and performance bottlenecks

  KVStore for Redis supports multiple memory types. You can upgrade the memory type based on your service requirements.

  The cluster architecture allows you to elastically scale the storage space and throughput performance of the database system. This eliminates the performance bottlenecks.
- Multiple instance types support flexible specifications changes

  The single-node cache architecture and two-node storage architecture are applicable to various service scenarios. You can flexibly change instance specifications.
- Monitoring and alerts allow you to check instance status in real time

  KVStore for Redis provides monitoring and alerts of instance information, such as CPU usage, connections, and disk utilization. You can check instance status anywhere and at any time.
- Visual management simplifies operations and maintenance

  The KVStore for Redis console, a visual management platform, allows you to easily perform frequent and risky operations, such as instance cloning, backup, and data restoration.
- Automatic engine version upgrades prevent software flaws

  The system automatically upgrades engine versions and efficiently fixes flaws so that you can easily manage database versions.
- Custom parameters support individual requirements

  You can set parameters in the KVStore for Redis console to make full use of system resources.

6.5 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.

6.6 Limits

Item	Description
List data type	The number of lists is not limited. The size of each element is 512 MB or less. We recommend that the number of elements in a list is less than 8,192. The value length is 1 MB or less.
Set data type	The number of sets is not limited. The size of each element is 512 MB or less. We recommend that the number of elements in a set is less than 8,192. The value length is 1 MB or less.

Item	Description
Sorted set data type	The number of sorted sets is not limited. The size of each element is 512 MB or less. We recommend that the number of elements in a sorted set is less than 8,192. The value length is 1 MB or less.
Hash data type	The number of fields is not limited. The size of each element in a hash table is 512 MB or less. We recommend that the number of elements in a hash table is less than 8, 192. The value length is 1 MB or less.
Number of databases (DBs)	Each instance supports 256 DBs.
Supported Redis commands	For more information, see the "Supported Redis commands" topic of *KVStore for Redis User Guide* .
Monitoring and alerts	KVStore for Redis does not provide capacity alerts. You have to configure this feature in CloudMonitor. We recommend that you set alerts for the following metrics: instance faults, instance failover, connection usage, failed operations, capacity usage, write bandwidth usage, and read bandwidth usage.
Expired data deletion policies	· Active expiration: the system periodically detects and deletes expired keys in the background. · Passive expiration: the system deletes expired keys when you access these keys.
Idle connection recycling mechanism	KVStore for Redis does not actively recycle idle connection s to KVStore for Redis. You can manage the connections.
Data persistence policy	KVStore for Redis uses the AOF_FSYNC_EVERYSEC policy, and runs the fysnc command at a one-second interval.

6.7 Terms

Redis

A high-performance key-value storage system that works as a cache and store and that is compatible with BSD open-source protocols.

Instance ID

An instance corresponds to a user space, and serves as the basic unit of using Redis.

Redis has limits on instance configurations, such as connections, bandwidth, and CPU processing capacity. These limits vary according to different instance types. You can view the list of instance identifiers that you have purchased in the console . KVStore for Redis instances are classified into master-replica instances and high-performance cluster instances.

Master-replica instance

The KVStore for Redis instance that contains a master-replica structure. The master -replica instance provides limited capacity and performance.

High-performance cluster instance

The KVStore for Redis instance that runs in a scalable cluster architecture. Cluster instances provide better scalability and performance, but they still have limited features.

Connection address

The host address for connecting to KVStore for Redis. The connection address is displayed as a domain name. To obtain the connection address, go to the Instance Information tab page, and check the address in the Connection Information field.

Eviction policy

The policy that KVStore for Redis uses to delete earlier data when the memory of KVStore for Redis reaches the upper limit as specified in `maxmemory`. Eviction policies of KVStore for Redis are consistent with Redis eviction policies. For more information, see *Using Redis as an LRU cache*.

DB

The abbreviation of the word "database" to indicate a database in KVStore for Redis . Each KVStore for Redis instance supports 256 databases numbered DB 0 to DB 255.

6.8 Instance types

Note:

The maximum bandwidth includes the maximum upstream bandwidth and the maximum downstream bandwidth.

Standard dual-replica edition

Table 6-1: Standard plan

Type	Service code	Maximu number of connecti s	Maximum bandwidth (MB)	CPU	Description	Zone-disaster recovery deployment
1 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb1g.1db.0rodb.4proxy.default	10,000	10	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region
2 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb2g.1db.0rodb.4proxy.default	10,000	16	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region
4 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb4g.1db.0rodb.4proxy.default	10,000	24	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region

Type	Service code	Maximu number of connecti s	Maximum bandwidth (MB)	CPU	Description	Zone-disaster recovery deployment
8 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb8g.1db.0rodb.4proxy.default	10,000	24	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region
16 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb16g.1db.0rodb.4proxy.default	10,000	32	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region
32 GB standard primary/secondary edition for zone-disaster recovery	redis.logic.sharding.drredissdb32g.1db.0rodb.4proxy.default	10,000	32	1-core	Primary/secondary instance for zone-disaster recovery	Deployed across two zones in one region

Table 6-2: Premium plan

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description	Zone-disaster recovery deployment
1 GB advanced primary/secondary edition	redis.master.small.special2x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone
2 GB advanced primary/secondary edition	redis.master.mid.special2x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone
4 GB advanced primary/secondary edition	redis.master.stand.special2x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone
8 GB advanced primary/secondary edition	redis.master.large.special1x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone
16 GB advanced primary/secondary edition	redis.master.2xlarge.special1x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone
32 GB advanced primary/secondary edition	redis.master.4xlarge.special1x	20,000	48	1-core	Primary/secondary instance	Deployed in one zone

Table 6-3: Cluster edition

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description
16 GB cluster edition	redis.sharding.small.default	80,000	384	4-core	High-performance cluster instance
32 GB cluster edition	redis.sharding.mid.default	80,000	384	8-core	High-performance cluster instance
64 GB cluster edition	redis.sharding.large.default	80,000	384	8-core	High-performance cluster instance
128 GB cluster edition	redis.sharding.2xlarge.default	160,000	768	16-core	High-performance cluster instance
256 GB cluster edition	redis.sharding.4xlarge.default	160,000	768	16-core	High-performance cluster instance
512 GB cluster edition	redis.logic.sharding.16g.32db.0rodb.32proxy.default	320,000	1,536	8-core	
1 TB cluster edition	redis.sharding.16xlarge.default	640,000	3,072	8-core	
2 TB cluster edition	redis.sharding.32xlarge.default	1,280,000	6,144	8-core	

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description
4 TB cluster edition	redis.logic.sharding.16g.256db.0rodb.256proxy.default	2,560,000	12,288	16-core	

Table 6-4: Cluster edition for zone-disaster recovery

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description
16 GB cluster edition for zone-disaster recovery	redis.logic.sharding.drredismdb16g.8db.0rodb.8proxy.default	80,000	384	8-core	Cluster instance for zone-disaster recovery
32 GB cluster edition for zone-disaster recovery	redis.logic.sharding.drredismdb32g.8db.0rodb.8proxy.default	80,000	384	8-core	Cluster instance for zone-disaster recovery
64 GB cluster edition for zone-disaster recovery	redis.logic.sharding.drredismdb64g.8db.0rodb.8proxy.default	80,000	384	8-core	Cluster instance for zone-disaster recovery

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description
128 GB cluster edition for zone-disaster recovery	redis.logic.sharding.drredismdb128g.16db.0rodb.16proxy.default	160,000	768	16-core	Cluster instance for zone-disaster recovery
256 GB cluster edition for zone-disaster recovery	redis.logic.sharding.drredismdb256g.16db.0rodb.16proxy.default	160,000	768	16-core	Cluster instance for zone-disaster recovery
256 GB cluster edition for zone-disaster recovery	redis.logic.sharding.16g.32db.0rodb.32proxy.default	320,000	1,536	8-core	Cluster instance for zone-disaster recovery
1 TB cluster edition for zone-disaster recovery	redis.sharding.16xlarge.default	640,000	3,072	8-core	Cluster instance for zone-disaster recovery
2 TB cluster edition for zone-disaster recovery	redis.sharding.32xlarge.default	1,280,000	6,144	8-core	Cluster instance for zone-disaster recovery

Type	Service code	Maximum number of connections	Maximum bandwidth (MB)	CPU	Description
4 TB cluster edition for zone-disaster recovery	redis.logic.sharding.16g.256db.0rodb.256proxy.default	2,560,000	12,288	16-core	Cluster instance for zone-disaster recovery

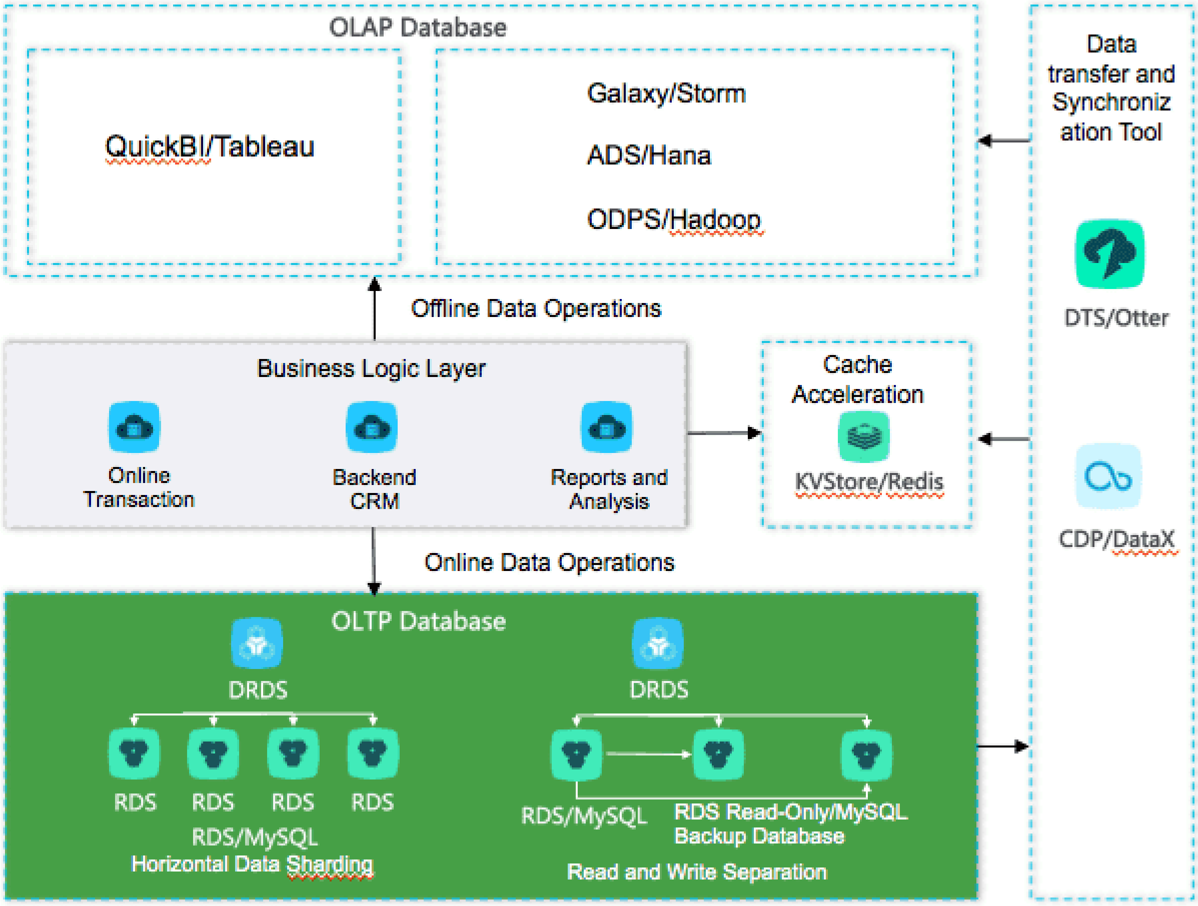
7 Distributed Relational Database Service (DRDS)

7.1 What is Distributed Relational Database Service?

Distributed Relational Database Service (DRDS) is a middleware product independently developed by Alibaba Group to solve scaling problems of single-instance relational databases. DRDS is compatible with the MySQL protocol and supports most MySQL data manipulation language (DML) and data definition language (DDL) syntax. It has the core capabilities and features of distributed databases, such as database sharding, table sharding, smooth scale-out, configuration upgrade and downgrade, and transparent read/write splitting. DRDS features lightweight (stateless), flexibility, stability, and efficiency, and provides you with O&M capabilities throughout the lifecycle of distributed databases.

It is mainly used for operations on large-scale online data. By splitting data in specific business scenarios, DRDS maximizes the operation efficiency, meeting the requirements of online businesses on relational databases.

Figure 7-1: DRDS architecture



Problems solved

- **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. Distribute d solutions use multiple instances to work jointly, effectively resolving the data storage capacity and access volume bottlenecks.**
- **Difficult scale-out of relational databases: Due to the inherent attributes of distributed databases, shard storage nodes can be changed through smooth data migration, supporting the dynamic scale-out of relational databases.**

7.2 Benefits

Distributed architecture

Through horizontal partitioning and cluster deployment of a single service, single-point of failures (SPOFs) of Server Load Balancer (SLB), Distributed Relational Database Service (DRDS), and ApsaraDB for RDS (RDS) are resolved, supporting scalability of distributed databases.

Auto scaling

DRDS instances and RDS instances can be dynamically upgraded and downgraded for flexible service capabilities.

High performance

DRDS for RDS (MySQL) splits data in specific business scenarios and clusters data with major business operations to speed up the response of online transactional operations. By using the columnar storage and knowledge grid, DRDS for HiStore significantly speeds up the response of common analytic operations such as large-scale data aggregation and ad hoc queries. It also controls storage costs through high compression.

Secure and controllable

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

7.3 Architecture

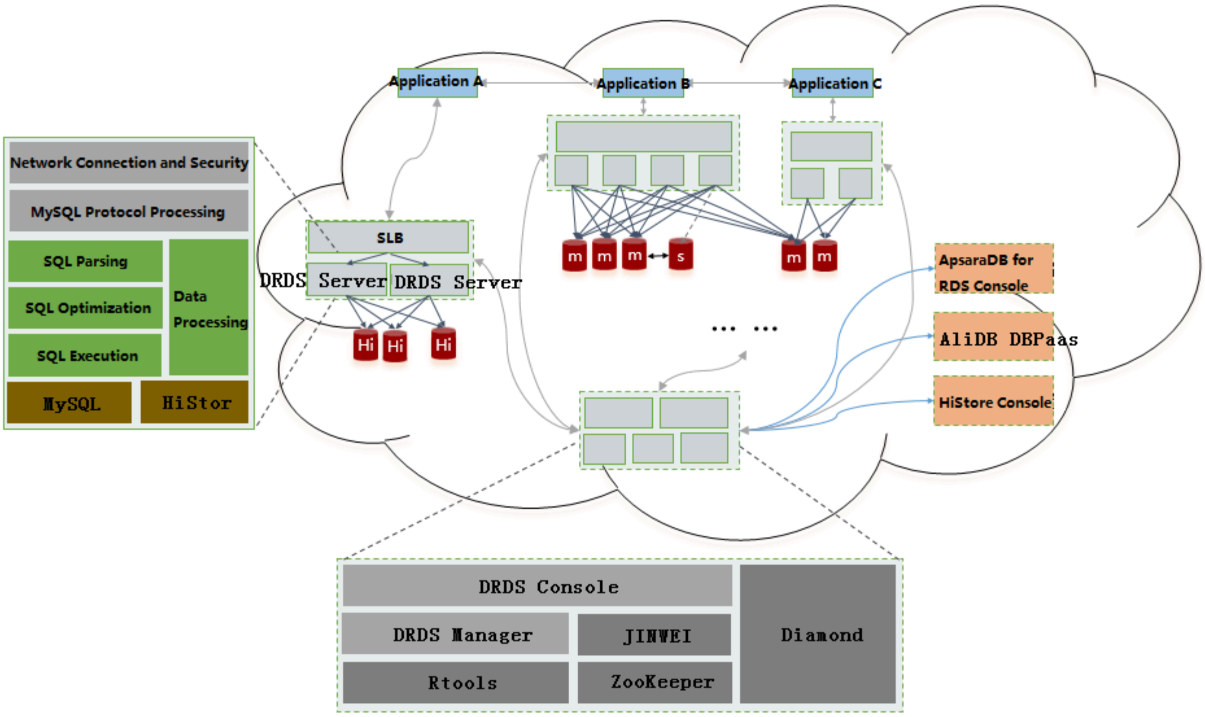
Distributed Relational Database Service (DRDS) supports two data output methods: overall output by Alibaba Cloud and separate output by Alibaba middleware. The two output methods differ in features and components on which DRDS depends.

The following table describes the differences between them.

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

The following figure shows the system architecture of DRDS.

Figure 7-2: DRDS system architecture



DRDS server

The DRDS servers are the service layer of DRDS. Multiple DRDS servers make up a cluster to provide distributed database services, including the read/write splitting , routed SQL execution, result merging, dynamic database configuration, and globally unique ID (GUID).

MySQL (m and s in the figure)

ApsaraDB RDS for MySQL (MySQL) stores and operates online data. It implements high availability (HA) through MySQL primary-secondary replication, and implements the dynamic database failover with the ApsaraDB for RDS (RDS) primary-secondary failover system.

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

HiStore

When DRDS outputs data separately (not overall output by Alibaba Cloud), it uses HiStore as the physical storage. HiStore is a low-cost, 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 a lower cost than row storage (such as MySQL).

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

DBPaaS

When DRDS outputs data separately (not overall output of Alibaba Cloud), it comes with the MySQL O&M platform DBPaaS to implement management, monitoring, alerting, and resource management in the MySQL lifecycle.

SLB

You do not need to install a client on user instances. User requests are distributed through Server Load Balancer (SLB). When an instance fails or a new instance is added, SLB ensures that traffic on the bound instances is distributed evenly.

VIPServer

You need to install a client on user instances, with a weak dependence 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 is a system responsible for DRDS configuration storage and management . It provides the configuration storage, query, and notification functions. Diamond stores the database source data, sharding rules, and DRDS switch configuration.

Data Replication System

Data Replication System is responsible for data migration and synchronization of DRDS. 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 (GSI). Data Replication System requires the support of ZooKeeper and DRDS Rtools.

DRDS console

The DRDS console is designed for business database administrators (DBAs) to isolate resources as required and perform operations, such as instance management, database and table management, read/write splitting configuration, smooth scale-out, monitoring data display, and IP address whitelisting.

DRDS manager

The DRDS manager is designed for global O&M personnel and DBAs. It provides the following DRDS resource management and system monitoring functions:

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

Rtools

Rtools is the O&M support system of DRDS. It allows you to manage database configuration, read/write weight, connection parameters, database and table topologies, and sharding rules.

7.4 Features

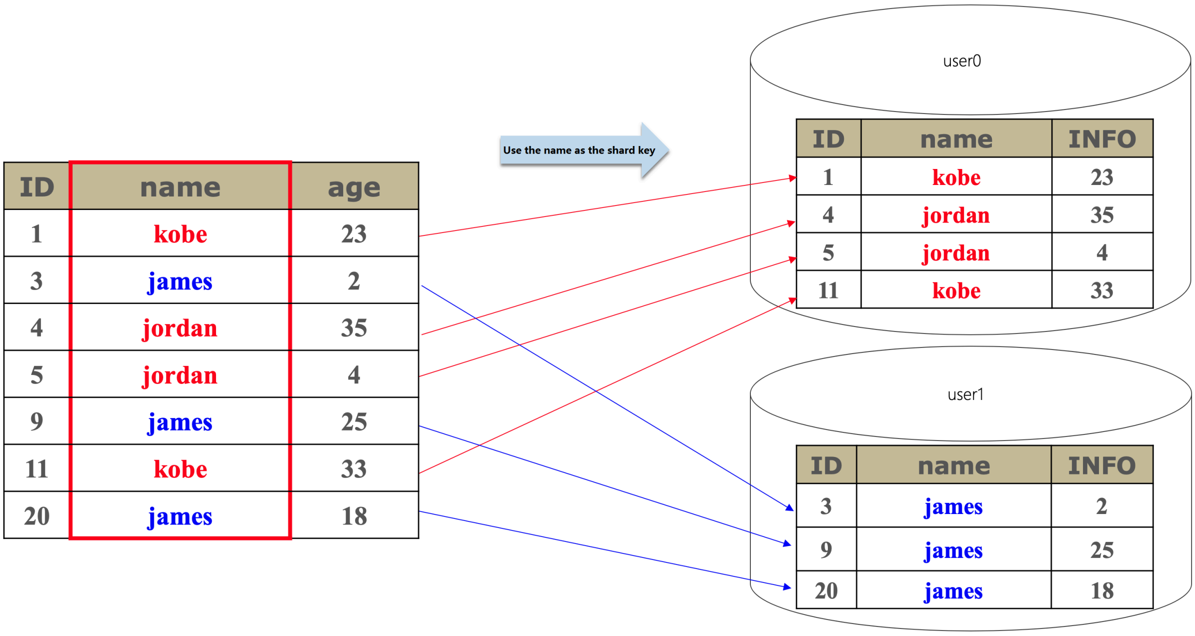
7.4.1 Scalability principle

Concurrency and storage capacity scalability

The essence of scalability lies in splitting. Distributed Relational Database Service (DRDS) distributes data to multiple ApsaraDB RDS for MySQL (MySQL) instances to obtain the distribution of read/write requests and storage through *Horizontal partitioning (database and table shards)*. **The DRDS layer is stateless and increases nodes to cope with concurrent SQL loads, which is similar to a business application.**

Horizontal partitioning (database and table shards)

Data is distributed to multiple MySQL instances according to certain calculatio n and routing rules. In fact, DRDS has many algorithms to cope with the load in various scenarios.



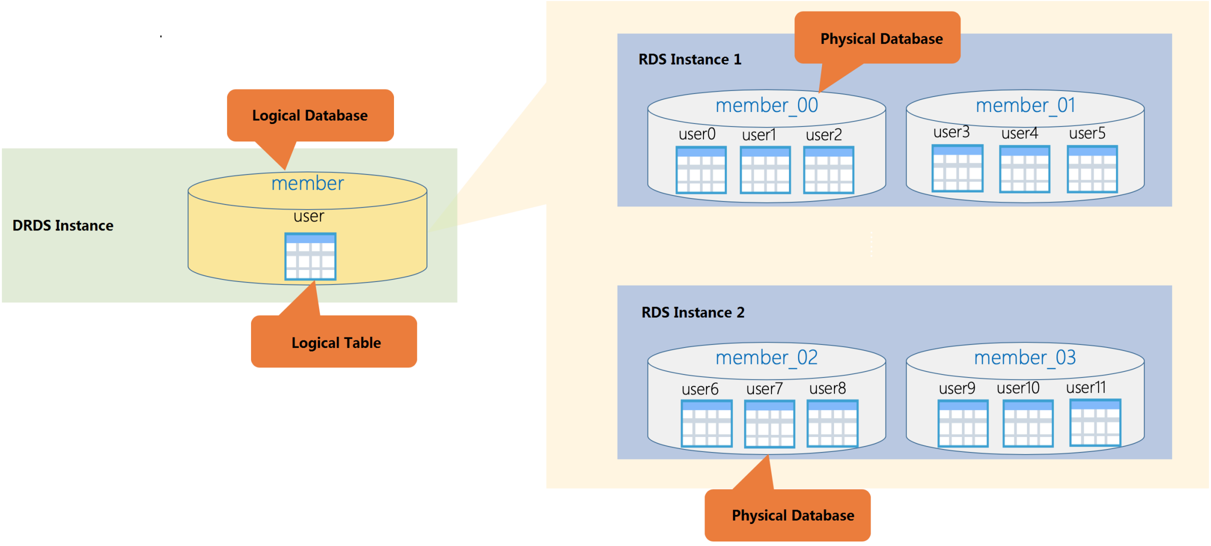
ID	name	age
1	kobe	23
3	james	2
4	jordan	35
5	jordan	4
9	james	25
11	kobe	33
20	james	18

user0

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

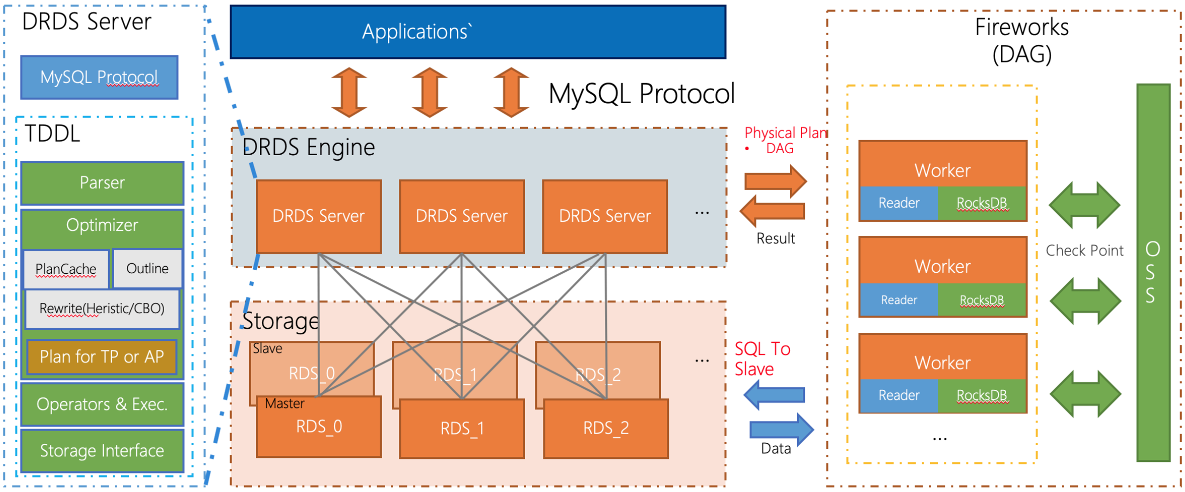
user1

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



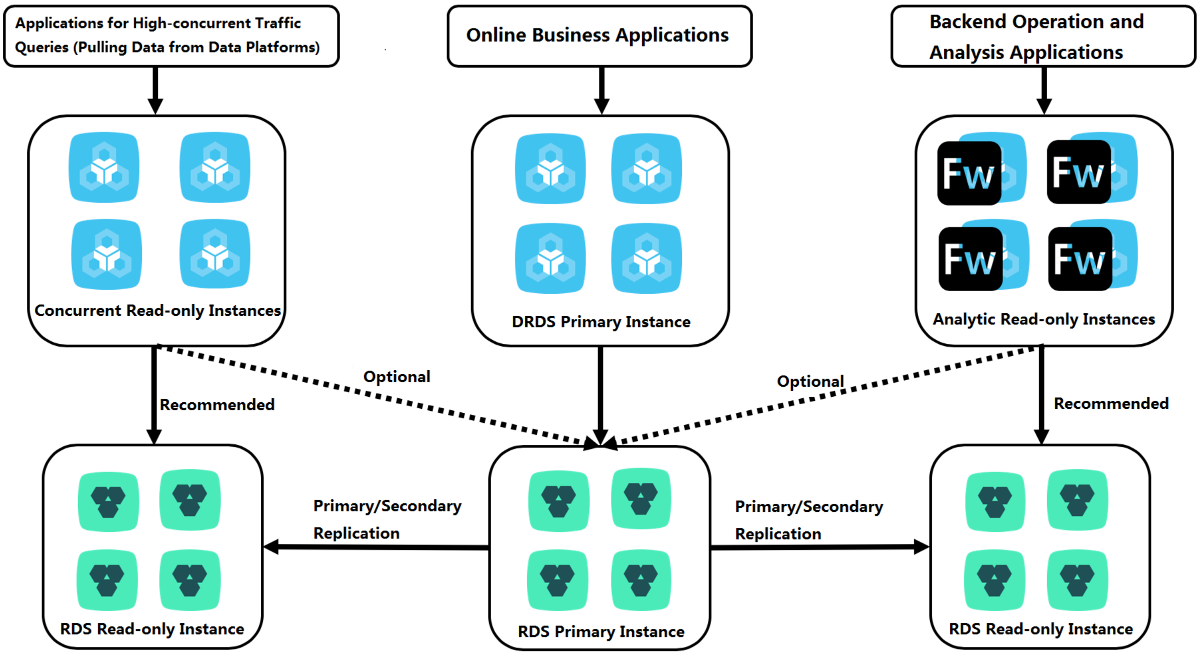
Computing scalability

DRDS often needs to perform complex computing on data far exceeding the capacity of a single instance. These SQL statements include multi-table join, multi-layer nested subqueries, grouping, sorting, and aggregation.



To process complex SQL statements in the online databases, DRDS has expanded the symmetric multiprocessing (SMP) and directed acyclic graph (DAG). SMP is fully integrated into the DRDS kernel, while DAG builds a compute cluster that dynamically obtains execution plans for distributed computing at runtime and improves the computing capability by adding nodes.

Currently, the DRDS instances that concurrently process data on multiple instances are provided for businesses in the form of analytic read-only instances.


Applications for High-concurrent Traffic Queries (Pulling Data from Data Platforms)
Online Business Applications
Backend Operation and Analysis Applications
Concurrent Read-only Instances
DRDS Primary Instance
Analytic Read-only Instances
Optional
Optional
Recommended
Recommended
Primary/Secondary Replication
Primary/Secondary Replication
RDS Read-only Instance
RDS Primary Instance
RDS Read-only Instance

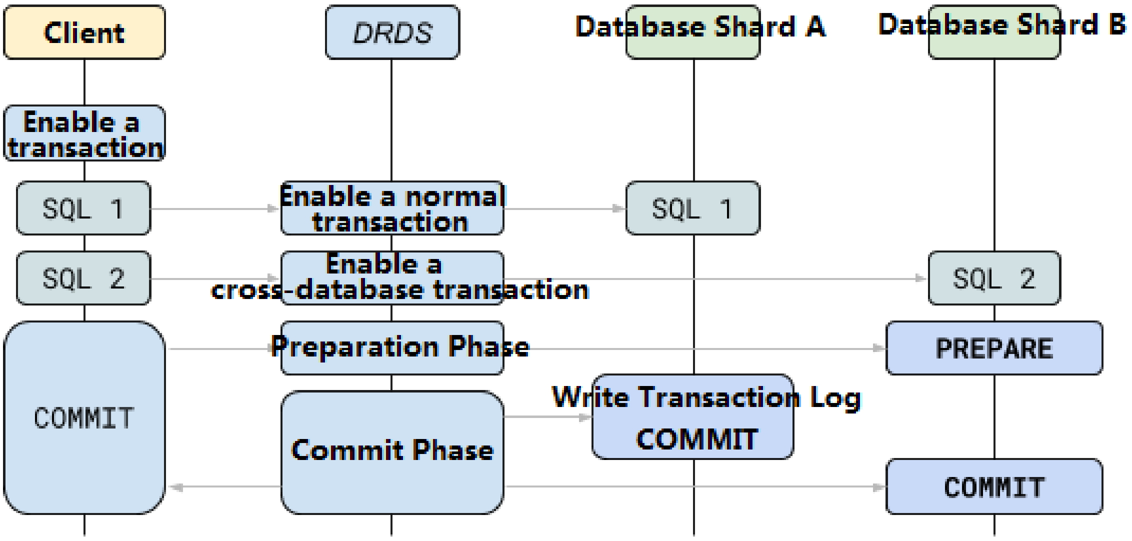
7.4.2 Distributed transactions

Distributed transactions use two-phase commit (2PC) to ensure the atomicity and consistency of transactions.

A 2PC transaction is divided into the prepare and commit phases.

- In the prepare phase, data nodes prepare all the resources (such as resources locking and logging) required for transaction commitment.
- In the commit phase, each data node actually commits transactions.

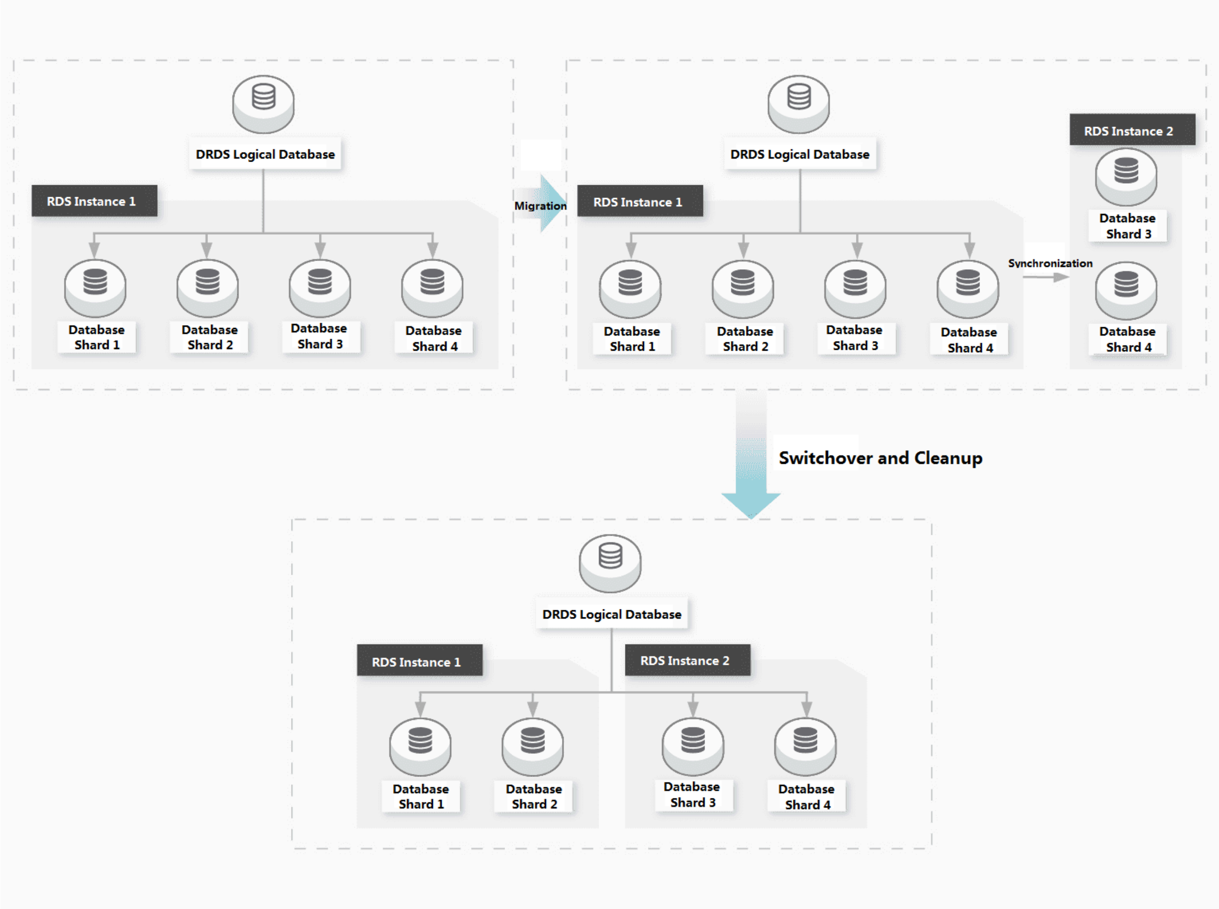
When you submit a distributed transaction, the Distributed Relational Database Service (DRDS) server, as a transaction manager, sends a commit request to each data node only after all data nodes (MySQL servers) have their resources prepared.



7.4.3 Smooth scale-out

When the underlying storage of the logical database reaches the physical bottleneck and needs to be scaled, for example, when the disk margin reaches 30%, you can smoothly scale out it to improve the performance.

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



7.4.4 Read/write splitting

When a primary instance is heavily loaded with many read requests, you can use the read/write splitting function of Distributed Relational Database Service (DRDS) to distribute the read traffic, which reduces the read pressure on the ApsaraDB for RDS (RDS) primary instance.

The read/write splitting function of DRDS is transparent to applications. The read traffic can be distributed to the primary RDS instance and multiple RDS read-only instances according to the read weight set in the DRDS console, without changing any code of the application. All the write traffic is distributed to the primary instance.

After read/write splitting is set, data is read from the primary RDS instance in real time with strong consistency, while data is read from read-only RDS instances without strong consistency as data on the read-only instances is replicated asynchronously from the primary instance, with a millisecond-level latency. For SQL statements requiring real-time and strongly consistent read, the primary instance can be specified for their execution through DRDS hints.

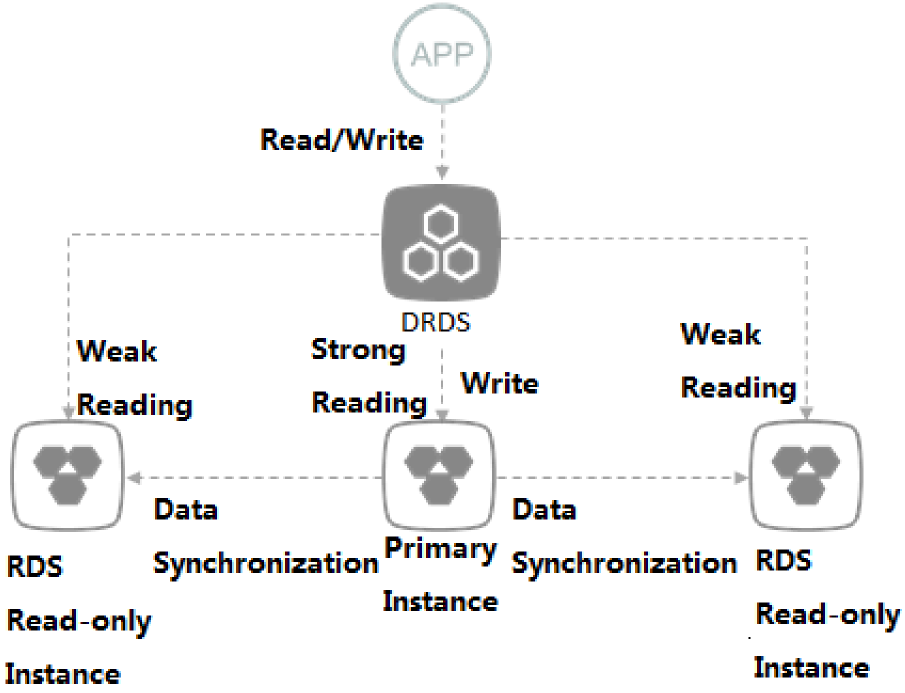
Read/write splitting in non-partition mode

In non-partition mode, DRDS can implement read/write splitting without horizontal partitioning. When you select an RDS instance to create a DRDS database in the DRDS console, you can directly introduce a database in the RDS instance to DRDS for read/write splitting. In this case, data migration is not required and horizontal partitioning cannot be performed on tables in the DRDS database.

Support for transactions by read/write splitting

Read/write splitting is valid only for read requests (query requests) that are not in explicit transactions (transactions that need to be explicitly committed or rolled back). Write requests and read requests (including read-only transactions) in explicit transactions are executed in the primary instance and are not distributed to read-only instances.

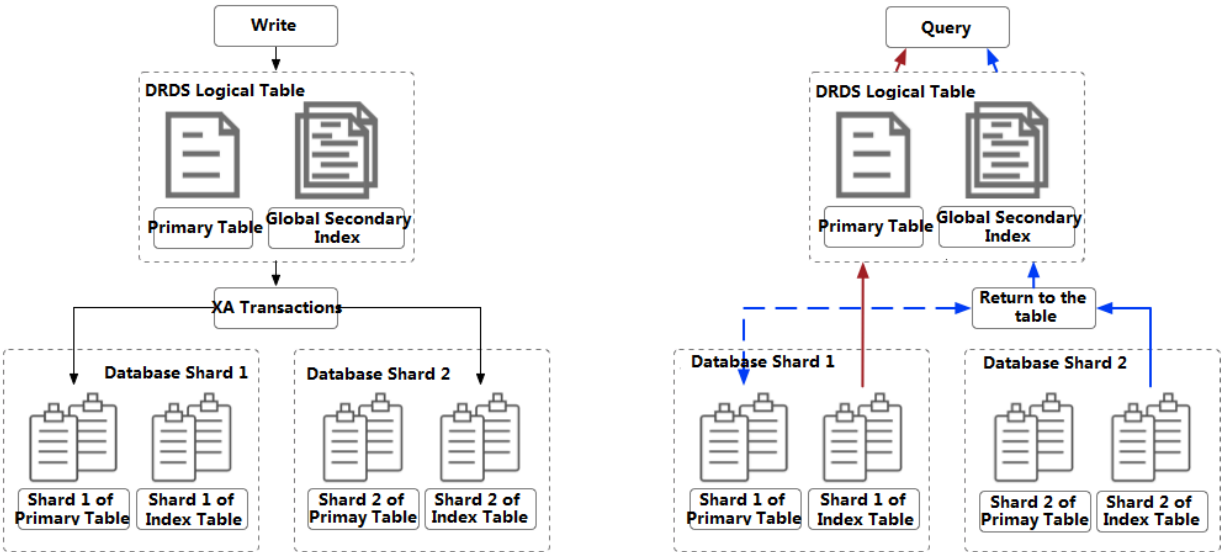
- Common SQL statements for read requests: SELECT, SHOW, EXPLAIN, and DESCRIBE
- Common SQL statements for write requests: INSERT, REPLACE, UPDATE, DELETE , and CALL



Read/Write Splitting

7.4.5 GSI

Global Secondary Index (GSI) of Distributed Relational Database Service (DRDS) allows users to add dimensions for sharding as needed and provides globally unique constraints. Each GSI corresponds to an index table and uses XA transaction multi-write to ensure strong data consistency between primary tables and index tables.



DRDS GSI provides the following capabilities:

- **Adds dimensions for sharding.**
- **Supports globally unique indexes.**
- **Provides XA transaction multi-write to ensure strong data consistency between primary tables and index tables.**
- **Supports column overwriting to reduce non-index data retrieval from tables and prevent extra overheads.**
- **Adds a primary table with GSI unlocked in online schema change.**
- **Uses hints to specify indexes to automatically determine whether non-index data retrieval from tables is needed.**

FAQ

Q: What problems can GSI solve?

A: If the queried dimension is different from the dimension for sharding of a logical table, cross-shard queries are initiated. As cross-shard queries increase, performance problems such as slow query and connection pool exhaustion may occur. GSI reduces cross-shard queries and eliminates performance bottlenecks by

adding dimensions for sharding. When creating GSI, you need to select a shard key that is different from that of the primary table.

Q: What is the relationship between a GSI and a local secondary index (LSI)?

A:

- LSI: An LSI stores data rows and corresponding index rows on the same shard in a distributed database. In DRDS, it is a MySQL secondary index of a physical table .
- GSI: A GSI stores data rows and corresponding index rows on different shards. It quickly determines the data shards involved in the query.
- Relationship between the GSI and LSI: When DRDS distributes queries to a single shard through the GSI, the LSI on the shard can improve the query performance of the shard.

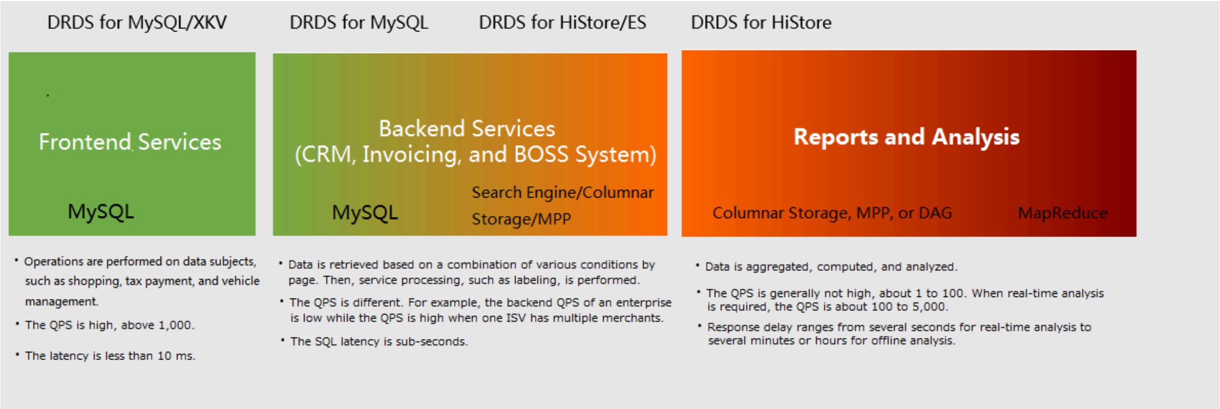
7.5 Scenarios

This topic describes the typical scenarios of Distributed Relational Database Service (DRDS).

DRDS is suitable for frontend high-concurrency and low-latency businesses. It splits data in specific business scenarios and provides distributed secondary indexes, enabling business databases to keep a high queries per second (QPS) upper limit.

DRDS is trying to support Alibaba columnar databases to meet the needs of the huge-volume storage with low costs, efficient data aggregation, and ad hoc queries.

Figure 7-3: DRDS scenarios



- Operations are performed on data subjects, such as shopping, tax payment, and vehicle management.
- The QPS is high, above 1,000.
- The latency is less than 10 ms.

- Data is retrieved based on a combination of various conditions by page. Then, service processing, such as labeling, is performed.
- The QPS is different. For example, the backend QPS of an enterprise is low while the QPS is high when one ISV has multiple merchants.
- The SQL latency is sub-seconds.

- Data is aggregated, computed, and analyzed.
- The QPS is generally not high, about 1 to 100. When real-time analysis is required, the QPS is about 100 to 5,000.
- Response delay ranges from several seconds for real-time analysis to several minutes or hours for offline analysis.

The following provides some examples of business scenarios:

- Customer-oriented Internet applications to carry out the business for users (DRDS for MySQL).
- Frontend high-concurrency and low-latency data businesses, such as the bank and hospital counter businesses, Internet of Vehicles (IoV) data operations, tracing, and fuel consumption curves (DRDS for MySQL).
- Storage and aggregation analysis of archived data that is unchangeable (including historical data), such as completed orders, logs, and operation and behavior records (DRDS for HiStore).

7.6 Limits

This topic describes the limits of Distributed Relational Database Service (DRDS).

Item	Limit
Table shard size	We recommend that a table shard contain a maximum of 5 million records.
Table shard quantity	Theoretically, the number of table shards in each database shard is not restricted, but depends on the hardware resources of the DRDS server.
Default database shard quantity for one ApsaraDB for RDS (RDS) instance	8, which cannot be changed.
Distributed join	DRDS supports most join semantics. However, DRDS has some restrictions on complex join semantics. For example , join between large tables may result in performance or system unavailability due to the high cost and slow speed. Therefore, prevent it whenever possible.

7.7 Terms

This topic defines and analyzes the terms related to Distributed Relational Database Service (DRDS).

Term	Description
DRDS	A distributed relational database service middleware developed by Alibaba, which is highly compatible with the MySQL protocol and syntax.

Term	Description
DRDS server	A core component of DRDS, which provides the SQL statement parsing, optimization, routing, and result aggregation functions.
DRDS instance	A distributed database service cluster that consists of a group of DRDS servers. Each DRDS server is stateless and processes SQL requests at the same time.
DRDS instance type	Reflection of the processing capability of DRDS instances. Each type of instance provides different CPU and memory resources. A higher instance type indicates a higher processing capability. For example , in a standard DRDS test scenario, the processing capability of an instance with 8 cores and 16 GB memory is twice that of an instance with 4 cores and 8 GB memory.
Instance upgrade and downgrade	DRDS can adjust the processing capability by upgrading or downgrading instance types.
Horizontal partitioning	A process of sharding and distributing table data in a single-instance database to databases on multiple storage instances according to specified sharding rules.
Sharding rule	A rule used to shard a logical database table into multiple physical table shards during horizontal partitioning.
Shard key	A database field that generates sharding rules during horizontal partitioning.
Database shard	After horizontal partitioning of DRDS, data in the logical database is stored in multiple physical storage instances. The physical database in each storage instance is a database shard.
Table shard	A physical data table in each database shard after horizontal partitioning of DRDS.
Logical SQL	An SQL statement sent by an application to DRDS.
Physical SQL	An SQL statement obtained after DRDS parses a logical SQL statement. The obtained physical SQL statement is then sent to ApsaraDB for RDS (RDS) for execution.

Term	Description
Transparent read/write splitting	When a single storage instance of DRDS encounters an access bottleneck, read-only instances can be added to share the load on the primary instance. No application code needs to be modified for the read /write splitting function of DRDS, which is called transparent read/write splitting.
Non-partition mode	DRDS supports expansion of database service capabilities through transparent read/write splitting without horizontal partitioning. This is non-partition mode.
Smooth scale-out	DRDS can scale out the database by adding storage instances. Smooth scale-out does not affect access to original data.
Small table broadcast	DRDS stores tables with small data volumes and infrequent updates as single tables, which are called small tables. A solution, which copies a small table to its joined database shards through data synchronization to improve the join efficiency, is called small table broadcast or small table copy.
Full table scan	In database sharding mode, if no shard key is specified in the SQL statement, DRDS executes the SQL statement on all table shards, and merges and returns the results, which is called full table scan. To prevent impact on performance, we recommend that you not perform full table scan.
DRDS sequence	A DRDS sequence (a 64-digit number of the BIGINT data type in MySQL) aims to ensure that the data (for example, PRIMARY KEY and UNIQUE KEY) in the defined unique field is globally unique and in ordered increments.
DRDS hint	A custom hint provided by DRDS to specify certain special actions. It uses related syntax to control the SQL execution to optimize SQL statements.

8 AnalyticDB for MySQL

8.1 What is AnalyticDB for MySQL?

AnalyticDB for MySQL (originally named ADS) is an Alibaba Cloud developed real-time online analytical processing (RT-OLAP) service that enables online analytics of large amounts of data at high concurrency. It can analyze hundreds of billions of data records from multiple dimensions at millisecond-level timing to provide you with data-driven insights into your business.

> **Note:**
> OLAP systems are often compared with online transaction processing (OLTP) systems. OLAP systems are good at performing multidimensional and complex query and analytics on large amounts of data, and the OLAP model is usually adopted in analytical databases. On the other hand, OLTP systems are good at performing transactional processing. In the OLTP model, data processing follows strong consistency and atomicity. The OLTP model supports frequent INSERT and UPDATE operations, and is usually adopted in relational database management systems such as MySQL and Microsoft SQL Server.

AnalyticDB for MySQL is an RT-OLAP system that has the following benefits:

- Compatible with MySQL, business intelligence (BI) tools, and extract, transform, and load (ETL) tools. You can use AnalyticDB for MySQL to analyze and integrate your data in a cost-effective, efficient, and simple manner.
- Uses relational models to store data and provides SQL statements to flexibly compute and analyze data. No advanced data modeling is required.
- Uses distributed computing technologies to provide excellent real-time computing capabilities. When processing tens of billions of data records or more , the performance of AnalyticDB for MySQL can achieve or even surpass that of multidimensional online analytical processing (MOLAP) systems. AnalyticDB for MySQL can compute tens of billions of data records within several hundred milliseconds. You can then explore large amounts of data without constraints, instead of viewing data reports based on a predefined logic.

- Computes hundreds of billions of data records in real time. In the AnalyticDB for MySQL system, all data generated in your business system is used for data analysis instead of a sample. This maximizes the effectiveness of analysis results.
- 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 end user products (including Internet products and internal enterprise analysis products). AnalyticDB for MySQL has been 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.

AnalyticDB for MySQL is a real-time computing system that provides rapid and flexible online data analysis and computation.

8.2 Benefits

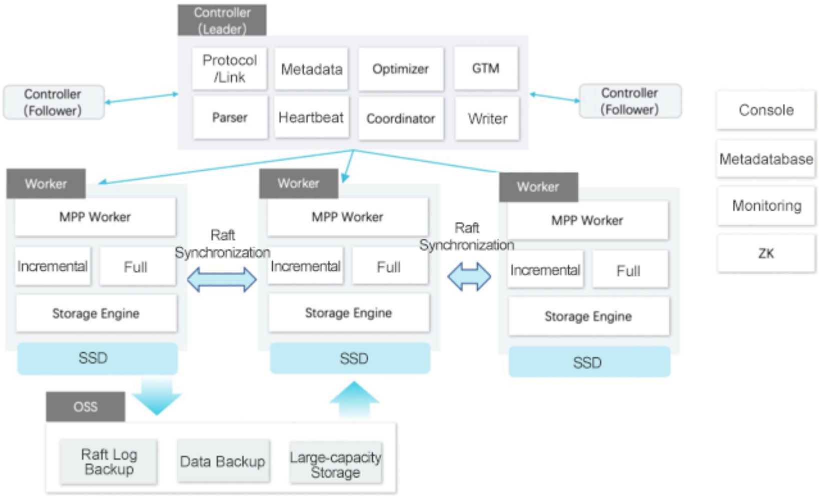
This topic describes the benefits of AnalyticDB for MySQL.

Benefit	Description
Computation for large amounts of data	Computes trillions of entries or PBs of data in a table.
Full data analysis	Analyzes all data but not samples to maximize the effectiveness of analysis results.
Rapid query response	Provides multidimensional pivoting for tens of billions of data entries within milliseconds.
High concurrency and availability	Provides high concurrency queries and ensures high availability through dynamic multi-replica data storage and computing. AnalyticDB for MySQL can serve as a backend system for products towards users, including Internet products and internal analysis products.
Flexible query method	Provides SQL statements to perform multidimensional analysis, pivoting, and filtering for large amounts of data in a flexible manner.
Parallel data import through multiple channels	Provides both online and offline channels to import data . The import performance increases linearly with the size of the cluster.

Benefit	Description
Refined security mechanism	Provides column-based permission management and finer-grained user operation audits. AnalyticDB for MySQL uses a public/private key mechanism to protect data security.
High compatibility	Provides full compatibility with MySQL protocols (including data element information), inherent compatibility with commercial analysis tools and applications, and built-in support for fast data access from various types of data sources. This greatly reduces the cost of access to business systems and software.

8.3 Architecture

AnalyticDB for MySQL is a distributed real-time computing system based on the MPP architecture. It is constructed based on the Apsara system and integrated with distributed retrieval technology. AnalyticDB for MySQL consists of the underlying dependencies, compute nodes, controllers, and storage nodes.



Underlying dependencies

Underlying dependencies include the following parts:

- Apsara system: is used to isolate resources in a virtualized manner, store data persistently, and construct data schemas and indexes.

- Metadatabase: refers to ApsaraDB for RDS or Table Store that stores 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 distribution of compute resources. It can also manage compute nodes and tasks running in the database background. A controller consists of multiple modules:

- SLB: manages grouping and load balancing of controllers.
- Client access manager.
- SQL parser.
- AnalyticDB for MySQL console.

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

- Clients and drivers that support MySQL 5.1, 5.5, or 5.6 protocols: 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 carry out computing tasks issued by controllers to read, filter, merge, and compute data.

Storage nodes

Storage nodes are responsible for writing data, saving data to disk storage, and copying data between nodes. Storage nodes support data backup and restoration.

8.4 Features

8.4.1 DDL

AnalyticDB for MySQL provides DDL statements to manage databases.

- Allows you to view all databases on which you have permissions by using SHOW DATABASES.
- Allows you to create tables and modify table attributes.
- Allows you to add columns to a table.
- Allows you to modify indexes.
- Allows you to create and delete views.

Note:

AnalyticDB for MySQL allows you to create and delete database clusters only in the ASCM console. DDL statements are not supported for these operations.

8.4.2 DML

8.4.2.1 SELECT

AnalyticDB for MySQL is over 95% compatible with standard MySQL queries. You can use SELECT statements to query data.

- Provides column mapping methods such as expressions, functions, aliases, column names, and CASE WHEN.
- Provides clauses such as FROM table name AS alias and JOIN table name AS alias.

  Provides joins between tables, including LEFT JOIN, RIGHT JOIN, FULL JOIN, and OUTER JOIN.
- Provides WHERE clauses combined with AND and OR operators, function expressions, or BETWEEN and IS operators.
- Provides GROUP BY operations for multiple columns and alias names generated from column mapping expressions such as CASE WHEN. Common aggregate functions are supported.
- Provides ORDER BY operations for expressions and columns in either ascending or descending order.

  Provides HAVING operations.
- Provides subqueries.

- **Provides COUNT(DISTINCT) operations.**
- **Provides constant columns.**
- **Provides operators such as UNION, UNION ALL, MINUS, and INTERSECT.**

8.4.2.2 INSERT, DELETE, and UPDATE

AnalyticDB for MySQL allows you to use INSERT, DELETE, and UPDATE statements to update data within the database.

- **Allows you to perform INSERT, DELETE, and UPDATE operations on real-time tables with defined primary keys.**
- **Provides multiple mechanisms to ensure that the written data is not lost. Both REPLACE INTO/INSERT OVERWRITE and INSERT IGNORE INTO statements are supported.**
  - `REPLACE INTO` **can be used to overwrite data in a table. The statement checks whether the data to be written already exists in the table based on the primary key. If yes, the statement deletes the row and inserts new data. Otherwise, the statement inserts new data directly.**
  - `INSERT INTO` **can be used to insert data to a table. An entry is not inserted if it is a duplicate of the primary key value, equivalent to** `INSERT IGNORE INTO`**.**
- **Provides** `INSERT INTO...SELECT FROM` **statements.**

8.4.3 System resource management

AnalyticDB for MySQL uses elastic compute units (ECUs) to manage resources, and provides distributed resource scheduling capabilities by using the underlying operating system and the Apsara system.

AnalyticDB for MySQL provides each database cluster with independent controller s, compute nodes, and storage nodes. You can control the resource usage of controllers, compute nodes, and storage nodes by selecting different ECU specificat ions. The following resources vary depending on ECU specifications: CPU cores, memory size (dedicated), and disk size.

8.4.4 Permissions and authorization

AnalyticDB for MySQL supports the standard MySQL permission model.

- **Provides ACL authorization for databases, tables, and columns.**
- **Allows privileged accounts to grant permissions to valid accounts.**

- Allows you to configure IP address whitelists for access from clients.
- Provides role-based permissions.
- Provides ADD USER and REMOVE USER statements to add and remove users.

  Provides GRANT statements to grant permissions and REVOKE statements to revoke permissions.
- Provides SHOW GRANTS ON statements to view user permissions of each object.
- Provides LIST USERS statements to view all users that have permissions.
- Privileged account: the account that is created in the console after a cluster is created. It has permissions to create standard accounts and grant permissions.
- Standard account: an account that can perform DDL and DML operations on the database after being authorized.

8.4.5 Metadata

This topic describes the metadata used in AnalyticDB for MySQL.

- TABLES: stores the basic information of all tables in a database cluster, including tables created by users and system metadata tables.
- USER_PRIVILEGES: stores all user authorization information within a database cluster.
- COLUMNS: stores the definition of each field for all tables within a database cluster.

8.4.6 Data import and export

This topic describes the data import and export methods supported by AnalyticDB for MySQL.

AnalyticDB for MySQL provides the following data import methods:

- Allows you to write data by using synchronization tools such as Kettle.
- Allows you to import data by using LOAD DATA statements in the format of CSV files, TEXT files, or files with multiple delimiters.
- Allows you to import data from OSS or MySQL by using external tables.

AnalyticDB for MySQL provides the following methods to export data in parallel:

- Allows you to display query results by using SELECT statements.
- Allows you to export data to OSS or MySQL by using external tables.

8.5 Scenarios

8.5.1 Overview

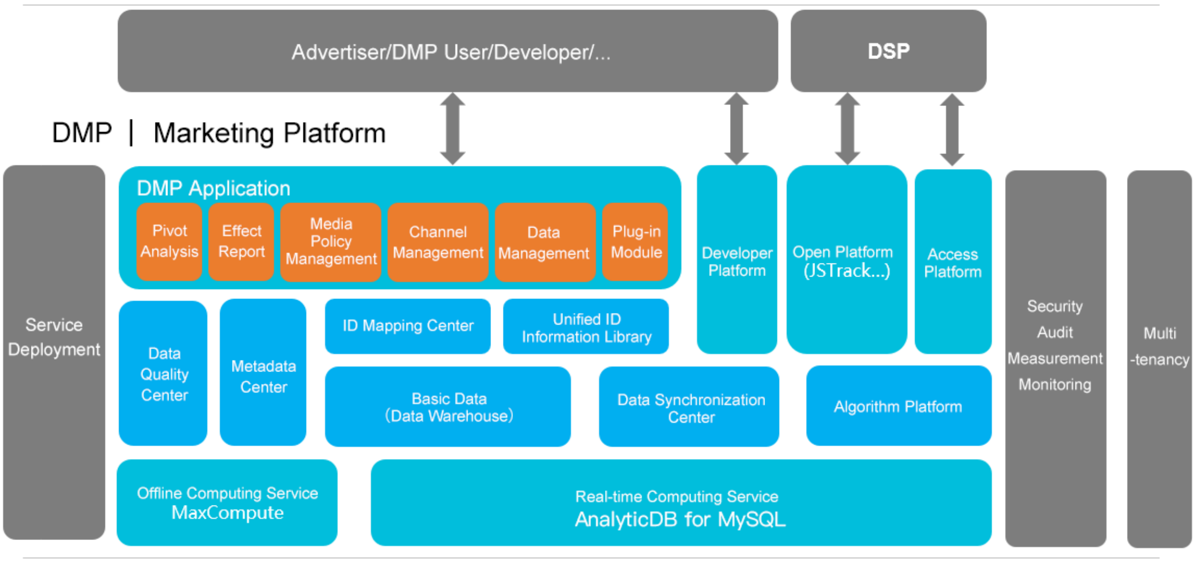
This topic describes the typical industries where AnalyticDB applies.

The following table describes the common scenarios of AnalyticDB.

Scenario	Description
E-commerce industry	A-CRM, popular product selection, automated operations, and SKU combination analysis
O2O	Data analysis, CRM system, and geo-fencing system
Advertising industry	Digital marketing and M-DMP system
Financial industry	Real-time multi-dimensional data analysis, transaction flow query system, and reporting system
High security	Crowd analysis, potential key element mining, relational network analysis, and detail query
Traffic and traffic police	Checkpoint-related vehicle data analysis and determination
Logistics and IoT	VoT data analysis, enterprise security monitoring data analysis, sensor data storage and retrieval, and real-time logistics databases

8.5.2 Alimama DMP

The following figure shows the application architecture of Alimama Data Management Platform (DMP).

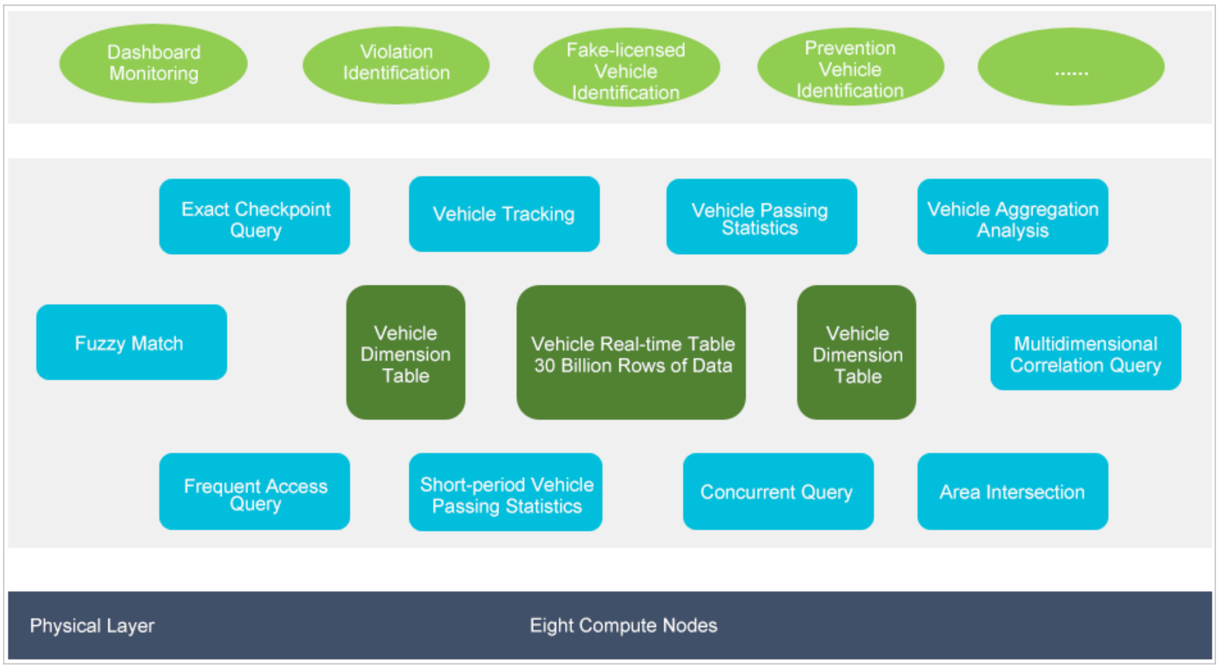


Big data services play core roles in the DMP system:

- MaxCompute performs user data scrubbing and tag mining.
- AnalyticDB provides advertisers with a way to view big data and compute crowd management data. AnalyticDB can export large amounts of specified user group data to the KVStore system that features faster query.
- The targeting engine serves the Demand-Side Platform (DSP) based on the KVStore data.

8.5.3 Traffic police

The following figure shows an example of traffic police application.


Dashboard Monitoring
Violation Identification
Fake-licensed Vehicle Identification
Prevention Vehicle Identification
......
Exact Checkpoint Query
Vehicle Tracking
Vehicle Passing Statistics
Vehicle Aggregation Analysis
Fuzzy Match
Vehicle Dimension Table
Vehicle Real-time Table 30 Billion Rows of Data
Vehicle Dimension Table
Multidimensional Correlation Query
Frequent Access Query
Short-period Vehicle Passing Statistics
Concurrent Query
Area Intersection
Physical Layer
Eight Compute Nodes

The traffic police business system has the following features:

- Large amounts of data: There are 30 to 50 billion tables that record the vehicles passing through various checkpoints around the city. These tables are stored for six months, and consumes 20 to 30 TBs of data.
- Rapid increase: Ten million rows of data are added every day in a municipal system.
- Complex query: Multiple departments need to be queried by using multiple query methods. Business application query applies to multiple scenarios, such as single table query, multi-table query (join), fuzzy search (like), trace analysis (in), area intersection (intersect), vehicle quantity query in a short period of time

(having count), and multi-user query. Complex queries have high requirements for table schema design, memory usage, CPU utilization, and query concurrency.

8.6 Limits

This topic describes the naming rules and limits for objects in AnalyticDB for MySQL.

Object	Naming rule	Limit
Database name	The database name can be up to 64 characters in length and can contain letters, digits, and underscores (_). It must start with a lowercase letter and cannot contain two or more consecutive underscores (_).	You cannot name a database as analyticdb because it is the name of a built-in database.
Table name	The table name must be 1 to 127 characters in length and can contain letters, digits, and underscores (_). It must start with a letter or underscore (_).	· The table name cannot contain single quotation marks ('), double quotation marks ("), exclamation marks (!), or spaces. · The table name cannot be an SQL reserved keyword.
Column name	The column name must be 1 to 127 characters in length and can contain letters, digits, and underscores (_). It must start with a letter or underscore (_).	· The column name cannot contain single quotation marks ('), double quotation marks ("), exclamation marks (!), or spaces. · The column name cannot be an SQL reserved keyword.

Object	Naming rule	Limit
Account name	The account name must be 2 to 16 characters in length and can contain lowercase letters, digits, and underscores (_). It must start with a lowercase letter and end with a lowercase letter or digit.	None
Password	The password must be 8 to 32 characters in length and must contain at least three of the following character types: uppercase letters, lowercase letters, digits, and special characters. Special characters include ! @ # $ % ^ & * ( ) _ + - =	None

8.7 Terms

This topic describes the basic concepts used in AnalyticDB for MySQL.

database cluster

A warehouse used to organize, store, and manage data. Database clusters are the basic unit used to isolate tenants. Each database cluster has independent computing resources, user permissions, and user quotas.

database account

An account used in AnalyticDB for MySQL. It can be a privileged or standard account. You can create a privileged account after creating an AnalyticDB for MySQL cluster as an administrator. You can create a standard account through SQL statements after logging on as a privileged account to an AnalyticDB for MySQL cluster. A privileged account can grant specific permissions to different departments. User operations can be audited in fine granularity.

table

AnalyticDB for MySQL supports standard relational table models.

column

Table data in AnalyticDB for MySQL is stored in columns. A column has the following features:

- Supports standard MySQL data types, such as BOOLEAN, TINYINT, SMALLINT, INT, BIGINT, FLOAT, DOUBLE, VARCHAR, DATE, and TIMESTAMP.
- Supports automatic creation and manual deletion of full table indexes.

index

AnalyticDB for MySQL automatically creates indexes for all columns. If a column does not need an index, you can execute the DISABLE INDEX statement on the column to delete its index.

primary key

AnalyticDB for MySQL allows you to specify a primary key for a table. When you execute an INSERT, UPDATE, or DELETE statement, AnalyticDB for MySQL can use the primary key to identify unique entries.

Note:

The primary key in AnalyticDB for MySQL is only used to identify unique entries, and cannot be modified. If you want to modify the primary key, you must create a new table.