Alibaba Cloud
Apsara Stack Enterprise

Technical Whitepaper

Version: 1911, Internal: V3.10.0

Issue: 20200319

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 intellectu al 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

Legal disclaimer.....I
Document conventions.....I
1 Elastic Compute Service (ECS).....1
1.1 What is ECS?.....1
1.2 Architecture.....3
1.2.1 Overview.....3
1.2.2 Virtualization platform and distributed storage.....3
1.2.3 Control system.....4
1.3 Features.....5
2 Auto Scaling (ESS).....7
2.1 What is ESS?.....7
2.2 Architecture.....9
2.3 Features.....10
2.3.1 Typical scenarios.....10
2.3.1.1 Overview.....10
2.3.1.2 Elastic scale-out.....10
2.3.1.3 Elastic scale-in.....11
2.3.1.4 Elastic recovery.....13
2.3.2 Function components.....13
3 Object Storage Service (OSS).....15
3.1 What is OSS?.....15
3.1.1 Basic concepts.....15
3.1.2 Advantages.....16
3.1.3 Scenarios.....17
3.2 Benefits.....17
3.3 Architecture.....19
3.3.1 System architecture.....19
3.3.2 Data forwarding procedure.....21
3.4 Features and principles.....22
3.4.1 Components.....22
3.4.2 Features.....23
3.4.3 Terms.....25
4 Table Store.....28
4.1 What is Table Store?.....28
4.1.1 Technical background.....28
4.1.2 Table Store technologies.....30
4.2 Benefits.....31
4.3 Architecture.....32
4.4 Features.....34

4.4.1 Users and instances..... 34
4.4.2 Data tables..... 35
4.4.3 Data partitioning..... 36
4.4.4 Common commands and functions..... 36
4.4.5 Authorization and access control..... 37
5 ApsaraDB for RDS..... 38
5.1 What is ApsaraDB for RDS?..... 38
5.2 Architecture..... 39
5.3 Features..... 39
5.3.1 Data link service..... 40
5.3.2 High-availability service..... 43
5.3.3 Backup service..... 45
5.3.4 Monitoring service..... 46
5.3.5 Scheduling service..... 47
5.3.6 Migration service..... 47
6 AnalyticDB for PostgreSQL..... 48
6.1 What is AnalyticDB for PostgreSQL?..... 48
6.1.1 Scenarios..... 48
6.2 Benefits..... 51
6.3 Architecture..... 52
6.4 Features..... 53
6.4.1 Distributed architecture..... 53
6.4.2 High-performance data analysis..... 54
6.4.3 High-availability service..... 54
6.4.4 Data synchronization and tools..... 54
6.4.5 Data security..... 55
6.4.6 Supported SQL features..... 55
7 KVStore for Redis..... 59
7.1 What is KVStore for Redis?..... 59
7.1.1 Scenarios..... 59
7.2 Benefits..... 61
7.3 Architectures..... 62
7.3.1 Overall system architecture..... 62
7.3.2 Components..... 64
7.4 Features..... 65
7.4.1 Data link service..... 65
7.4.1.1 Overview..... 65
7.4.1.2 DNS..... 66
7.4.1.3 SLB..... 66
7.4.1.4 Proxy..... 66
7.4.1.5 DB Engine..... 67
7.4.2 HA service..... 67
7.4.2.1 Overview..... 67
7.4.2.2 Detection..... 68

7.4.2.3 Repair....68
7.4.2.4 Notice....69
7.4.3 Monitoring service....69
7.4.3.1 Service-level monitoring....69
7.4.3.2 Network-level monitoring....69
7.4.3.3 OS-level monitoring....69
7.4.3.4 Instance-level monitoring....70
7.4.4 Scheduling service....70
8 Data Transmission Service (DTS)....71
8.1 What is DTS?....71
8.2 Benefits....71
8.3 Architecture....72
8.4 Environment requirements....73
8.5 Features....74
8.5.1 Data migration....74
8.5.1.1 Data migration....74
8.5.1.2 Data sources....74
8.5.1.3 Online migration....75
8.5.1.4 Migration modes....75
8.5.1.5 ETL features....75
8.5.1.6 Migration task....76
8.5.2 Data synchronization....76
8.5.2.1 Overview....76
8.5.2.2 Synchronization tasks....76
8.5.2.3 Synchronization objects....79
8.5.2.4 Advanced features....80
8.5.3 Data subscription....80
8.5.3.1 Real-time data subscription....80
8.5.3.2 Subscription channels and objects....80
8.5.3.3 Advanced features....82
9 Data Management (DMS)....83
9.1 What is DMS?....83
9.1.1 Product value....83
9.2 Technical advantages....86
9.3 Architecture....87
9.4 Features....89
10 Server Load Balancer (SLB)....91
10.1 What is Server Load Balancer?....91
10.2 Architecture....92
10.3 Features....95
10.4 Benefits....96
10.4.1 LVS in Layer-4 SLB....96
10.4.2 Tengine in Layer-7 SLB....100
11 Virtual Private Cloud (VPC)....101

11.1 What is VPC?....101
11.2 Benefits....103
11.3 Architecture....103
11.4 Features....106
12 Apsara Stack Security....107
12.1 What is Apsara Stack Security?....107
12.2 Advantages....108
12.3 Architecture....110
12.4 Features....111
12.4.1 Apsara Stack Security Standard Edition....111
12.4.1.1 Threat Detection Service....111
12.4.1.2 Traffic Security Monitoring....115
12.4.1.3 Server Guard....116
12.4.1.4 Web Application Firewall....123
12.4.2 Optional security services....125
12.4.2.1 DDoS Traffic Scrubbing....125
12.4.2.2 Sensitive Data Discovery and Protection....127
13 Apsara Stack DNS....132
13.1 What is Apsara Stack DNS?....132
13.2 Benefits....132
13.3 Architecture....134
13.4 Features....135
14 MaxCompute....137
14.1 What is MaxCompute?....137
14.1.1 Overview....137
14.1.2 Features and benefits....139
14.1.3 Benefits....141
14.1.4 Scenarios....142
14.1.5 Service specifications....147
14.1.5.1 Software specifications....147
14.1.5.1.1 Overview....147
14.1.5.1.2 Control and service....147
14.1.5.1.3 Data storage....148
14.1.5.1.4 Size of a single cluster....148
14.1.5.1.5 Projects....148
14.1.5.1.6 User management and security and access control....149
14.1.5.1.7 Resource management and task scheduling....153
14.1.5.1.8 Data tables....153
14.1.5.1.9 SQL....154
14.1.5.1.9.1 DDL....154
14.1.5.1.9.2 DML....155
14.1.5.1.9.3 Built-in functions....157
14.1.5.1.9.4 User-defined functions....157
14.1.5.1.10 MapReduce....157

14.1.5.1.10.1 Programming support....157
14.1.5.1.10.2 Job size....158
14.1.5.1.10.3 Input and output....158
14.1.5.1.10.4 MapReduce computing....159
14.1.5.1.11 Graph....159
14.1.5.1.11.1 Programming support....159
14.1.5.1.11.2 Job size....160
14.1.5.1.11.3 Graph loading....160
14.1.5.1.11.4 Iterative computing....160
14.1.5.1.12 Processing of unstructured data....161
14.1.5.1.12.1 Processing of Table Store data....161
14.1.5.1.12.2 Processing of OSS data....161
14.1.5.1.12.3 Multiple data sources....162
14.1.5.1.13 Spark on MaxCompute....162
14.1.5.1.13.1 Programming support....162
14.1.5.1.13.2 Data sources....162
14.1.5.1.13.3 Scalability....163
14.1.5.1.14 Elasticsearch on MaxCompute....163
14.1.5.1.14.1 Programming support....163
14.1.5.1.14.2 System capabilities....163
14.1.5.1.15 Other extensions....164
14.1.5.2 Hardware specifications....164
14.1.5.3 Specifications of DNS resources....169
14.2 Architecture....171
14.3 Features....176
14.3.1 Tunnel....176
14.3.1.1 Overview....176
14.3.1.2 TableTunnel....176
14.3.1.3 InstanceTunnel....178
14.3.1.4 UploadSession....179
14.3.1.5 DownloadSession....181
14.3.1.6 TunnelBufferedWriter....182
14.3.2 SQL....183
14.3.3 MapReduce....184
14.3.4 Graph....185
14.3.5 Unstructured data processing (integrated computing scenarios)....186
14.3.6 Unstructured data processing in MaxCompute....187
14.3.7 Enhanced features....187
14.3.7.1 Spark on MaxCompute....187
14.3.7.1.1 Open-source platform - Cupid....187
14.3.7.1.1.1 Overview....187
14.3.7.1.1.2 Compatibility with YARN....188
14.3.7.1.1.3 Compatibility with FileSystem....189
14.3.7.1.1.4 DiskDrive....189

14.3.7.1.2 Feature extensions....189
14.3.7.1.2.1 Overview....189
14.3.7.1.2.2 Security isolation....190
14.3.7.1.2.3 Data interconnection....190
14.3.7.1.2.4 Client mode....190
14.3.7.1.2.5 Spark ecosystem support....192
14.3.7.2 Elasticsearch on MaxCompute....192
14.3.7.2.1 Terms....192
14.3.7.2.2 How Elasticsearch on MaxCompute works....194
14.3.7.2.2.1 Overview....194
14.3.7.2.2.2 How distributed architecture works....194
14.3.7.2.2.3 How full-text retrieval works....196
14.3.7.2.2.4 How authentication control works....197
15 DataWorks....198
15.1 What is DataWorks?....198
15.1.1 Product overview....198
15.1.2 Scenarios....199
15.2 Technical advantages....200
15.3 Architecture....202
15.4 Services....203
15.4.1 DataStudio....203
15.4.2 Data Management....204
15.4.3 Data Integration....204
15.4.4 Tenant management....209
15.4.5 Data Quality....209
15.4.5.1 Overview of Data Quality....209
15.4.5.2 Use Data Quality to monitor batch data....211
15.4.5.3 Use Data Quality to monitor real-time data....214
15.4.6 Data Asset Management....215
15.4.7 Real-Time Analysis....215
15.4.8 Data Service....216
15.4.9 Intelligent Monitor....216
15.4.10 Scheduling system....219
15.4.10.1 Overview....219
15.4.10.2 Concepts....219
15.4.10.3 Architecture....220
15.4.10.4 State machine....221
15.4.10.5 Task dependencies....222
16 Realtime Compute....226
16.1 What is Realtime Compute?....226
16.1.1 Background....226
16.1.2 Key challenges of Realtime Compute....227
16.2 Technical advantages....228
16.3 Product architecture....232

16.3.1 Business architecture....232
16.3.2 Technical architecture....233
16.4 Functional principles....235
17 Machine Learning Platform for AI....236
17.1 What is machine learning?....236
17.2 Benefits....238
17.3 Architecture....238
17.3.1 System architecture....238
17.3.2 Architecture....239
17.4 Functions....241
17.4.1 Resource allocation and task scheduling....241
17.4.2 Model and compilation optimization....242
17.4.3 Compute engine....244
17.4.4 Online prediction system....245
17.4.5 List of functions by module....248
17.5 System metrics....252
18 Quick BI....255
18.1 What is Quick BI?....255
18.2 Benefits....256
18.3 Product architecture....256
18.3.1 System architecture....256
18.3.2 Components....257
18.3.3 Deployment....260
18.3.4 Server roles....260
18.4 Features....261
19 Apsara Big Data Manager (ABM)....262
19.1 What is Apsara Big Data Manager?....262
19.2 Benefits....263
19.3 Architecture....264
19.3.1 System architecture....264
19.4 Features....266
19.4.1 Small file merging....266
19.4.2 Job snapshot....267

1 Elastic Compute Service (ECS)

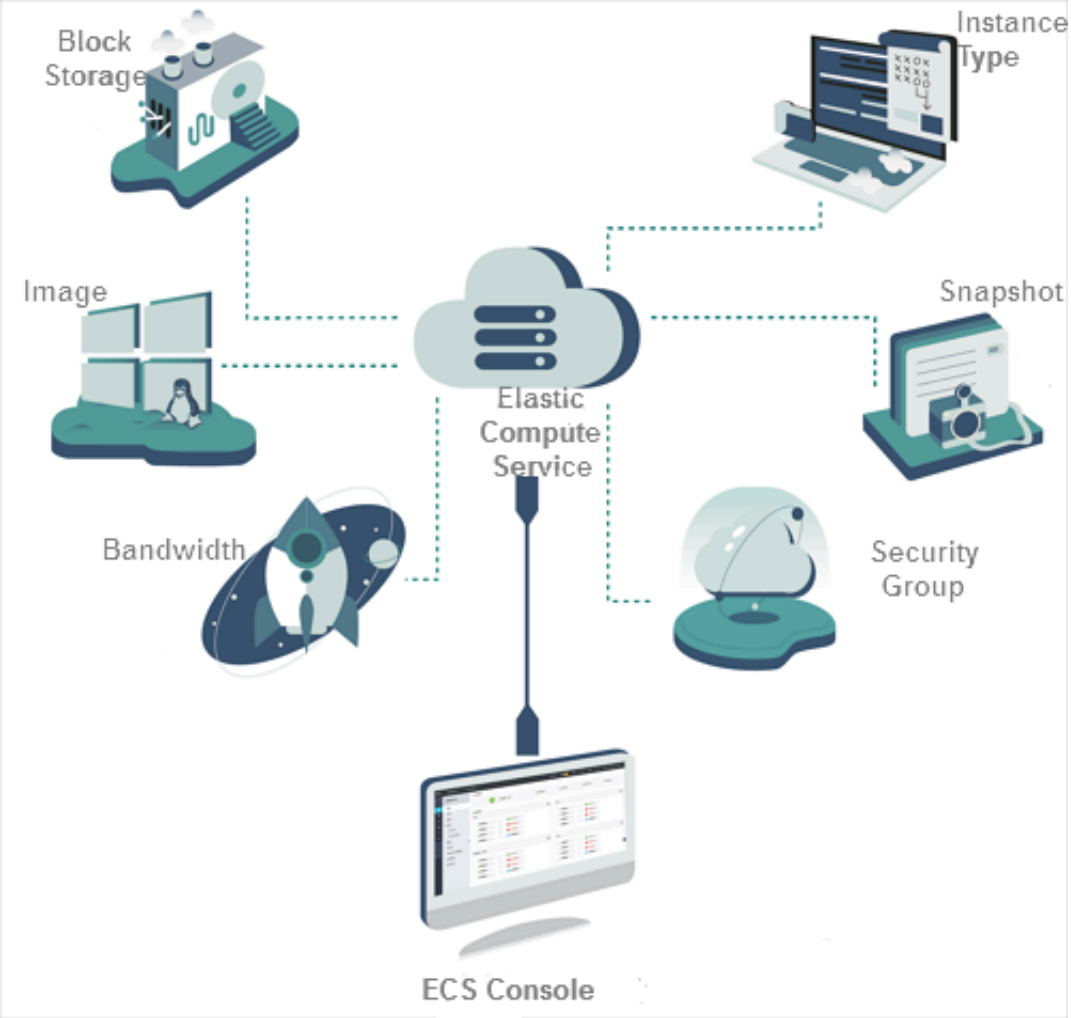
1.1 What is ECS?

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

An ECS instance is a virtual computing environment that contains the most basic components of computers such as the CPU, memory, and storage. Users perform operations on ECS instances. Instances are core components of ECS, and operations can be performed on instances through the ECS console. Other resources, such as

block storage, images, and snapshots, can only be used after they are integrated with ECS instances. For more information, see *Figure 1-1: ECS components*.

Figure 1-1: ECS components



1.2 Architecture

1.2.1 Overview

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

1.2.2 Virtualization platform and distributed storage

The foundation of Elastic Compute Service (ECS) as a service is virtualization. Apsara Stack uses KVM virtualization to virtualize physical resources and provide them as ECS resources.

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

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

The principles of the virtualization platform and distributed storage are shown in the following figures.

Figure 1-2: Triplicate backup

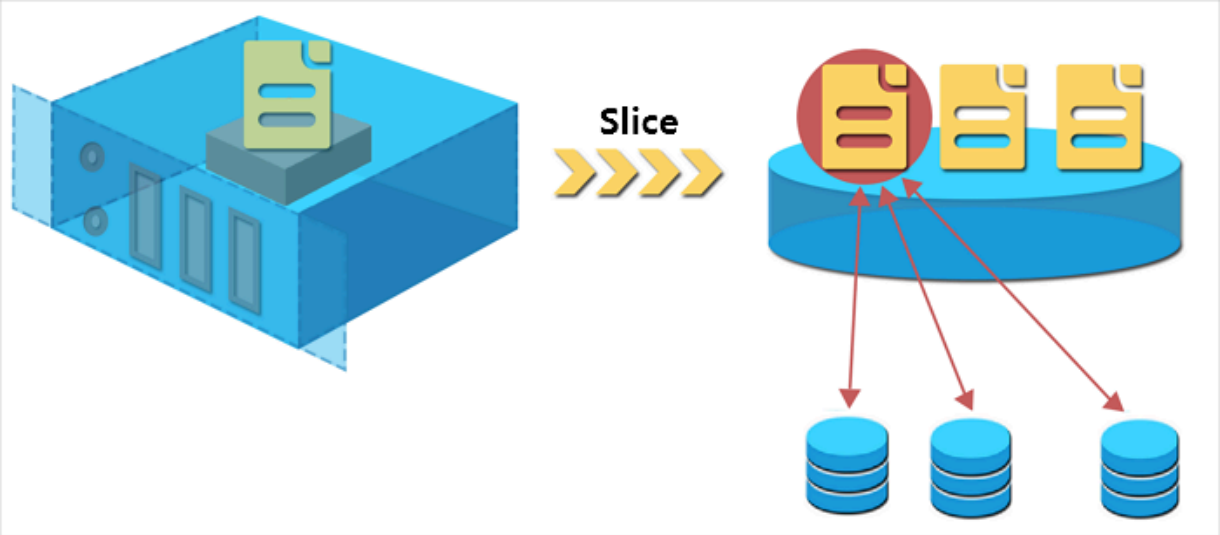
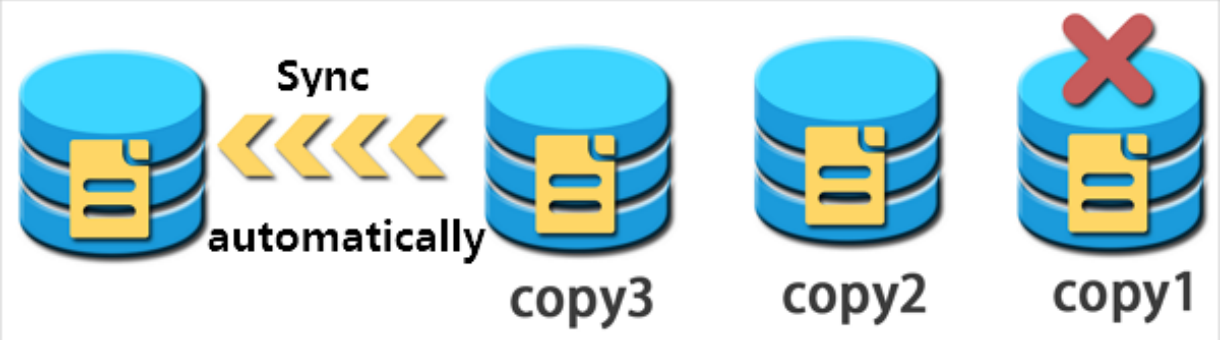


Figure 1-3: Automatic replication



1.2.3 Control system

The control system is the core of the ECS platform. It determines which physical servers to start ECS instances on and processes and maintains all ECS functions and information.

The control system is composed of the following modules:

- **Data collection module**

  This module is responsible for data collection throughout the virtualization platform, including computing, storage, and network resource usage. The data collection module serves as the basis for resource scheduling and allows you to centrally monitor and manage cluster resource usage.

- **Resource scheduling system**

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

- **ECS management module**

  This module can manage and control ECS instances through the start, stop, and restart operations.

- **Security control module**

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

1.3 Features

Instances are the core component that provides computation services to users in Elastic Compute Service (ECS). It only takes a few minutes to create and start an ECS instance. Once an ECS instance is created, it has specific system configurations. ECS instances allow you to compute business data efficiently compared with traditional servers.

ECS instances are used and operated in the same way as traditionally-hosted physical servers. You can perform a series of basic operations on ECS instances remotely or through APIs accessed in the control panel.

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

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

The ECS console in Apsara Stack console consists of the following tabs:

- **Overview**

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

- **Instances**

  You can view and manage the instances you have created on the VMs tab. You can start, stop, restart, and release online instances, as well as log on to a VNC, replace system disks, modify passwords, and change instance configurations. You can also view the basic information and configurations of instances.

- **Disks**

  You can view and manage the disks you have created on the Disks tab. You can reinitialize disks online, create snapshots, set automatic snapshot policies, release disks, and attach or detach disks. You can also view basic information and mounting information of disks.

- **Images**

  On the Images tab, you can view, manage, copy, share, and delete images that you have created.

- **Snapshots**

  On the Snapshots tab, you can view and manage the snapshots you have created. You can roll back disks online, create custom images, and delete snapshots.

- **Automatic snapshot policies**

  You can view and manage created automatic snapshot policies. You can also set automatic snapshot policies in batches, modify automatic snapshot policy information, and delete automatic snapshot policies.

- **Security groups**

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

- **ENIs**

  You can create, modify, delete, view, and manage Elastic Network Interfaces (ENIs), as well as bind or unbind ENIs to ECS instances.

- **Deployment sets**

  You can create, modify, delete, query, manage, and view the basic information of deployment sets.

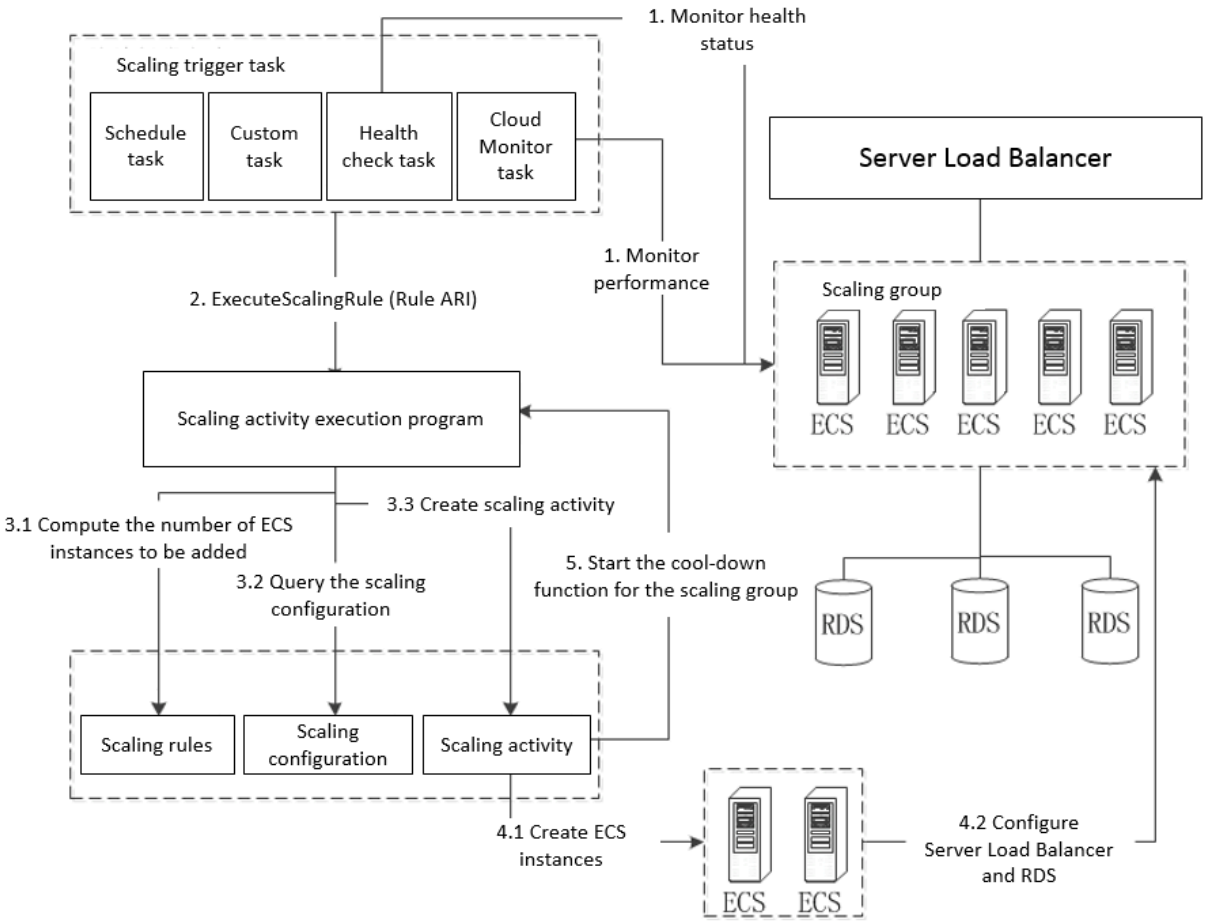
2 Auto Scaling (ESS)

2.1 What is ESS?

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

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

In addition, ESS is seamlessly integrated with Server Load Balancer (SLB) and ApsaraDB for Relational Database Service (RDS). This allows ESS to add or remove ECS instances to or from an SLB backend server group, as well as to add or remove IP addresses of ECS instances to or from an RDS whitelist. ESS eliminates the need to manually perform O&M operations, as it adapts to various complex scenarios and automatically processes business loads based on actual requirements.


1. Monitor health status
Scaling trigger task
Schedule task
Custom task
Health check task
Cloud Monitor task
Server Load Balancer
1. Monitor performance
2. ExecuteScalingRule (Rule ARI)
Scaling group
ECS
ECS
ECS
ECS
ECS
Scaling activity execution program
3.3 Create scaling activity
3.1 Compute the number of ECS instances to be added
3.2 Query the scaling configuration
5. Start the cool-down function for the scaling group
RDS
RDS
RDS
Scaling rules
Scaling configuration
Scaling activity
4.1 Create ECS instances
ECS
ECS
4.2 Configure Server Load Balancer and RDS

2.2 Architecture

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

Figure 2-1: Architecture

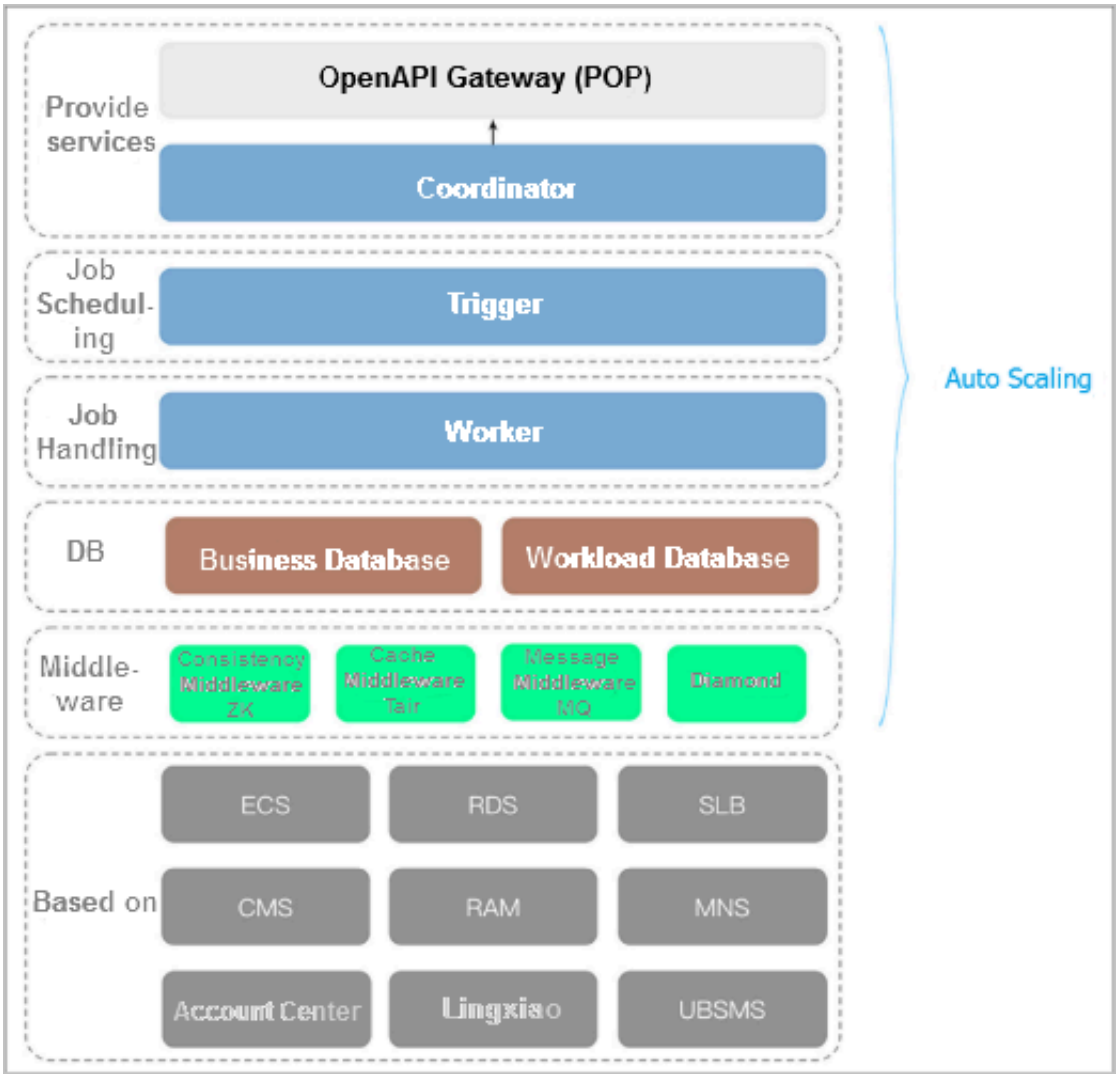


Table 2-1: Architecture description

Layer	Description
Middleware layer	ZooKeeper: ensures consistency by implementing distributed locks for Server Controller.
	Tair: provides caching services for Server Controller
	Message Queue (MQ): provides message queuing services of VM statuses.

Layer	Description
	Diamond: manages persistent configurations.
Database layer: the business database and workload database	Worker: The core of ESS. After receiving a task, it handles the entire life cycle of the task, including splitting, executing, and returning the execution results.
	Trigger: It obtains information from the health checks of instances and scaling groups, scheduled tasks, and CloudMonitor to perform tasks scheduling .
Public-facing services	Coordinator: serves as the ingress of the ESS architecture. It provides external management and control for services, processes API calls, and triggers tasks.
	OpenAPI Gateway: provides basic services such as authentication and parameter passthrough.

2.3 Features

2.3.1 Typical scenarios

2.3.1.1 Overview

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

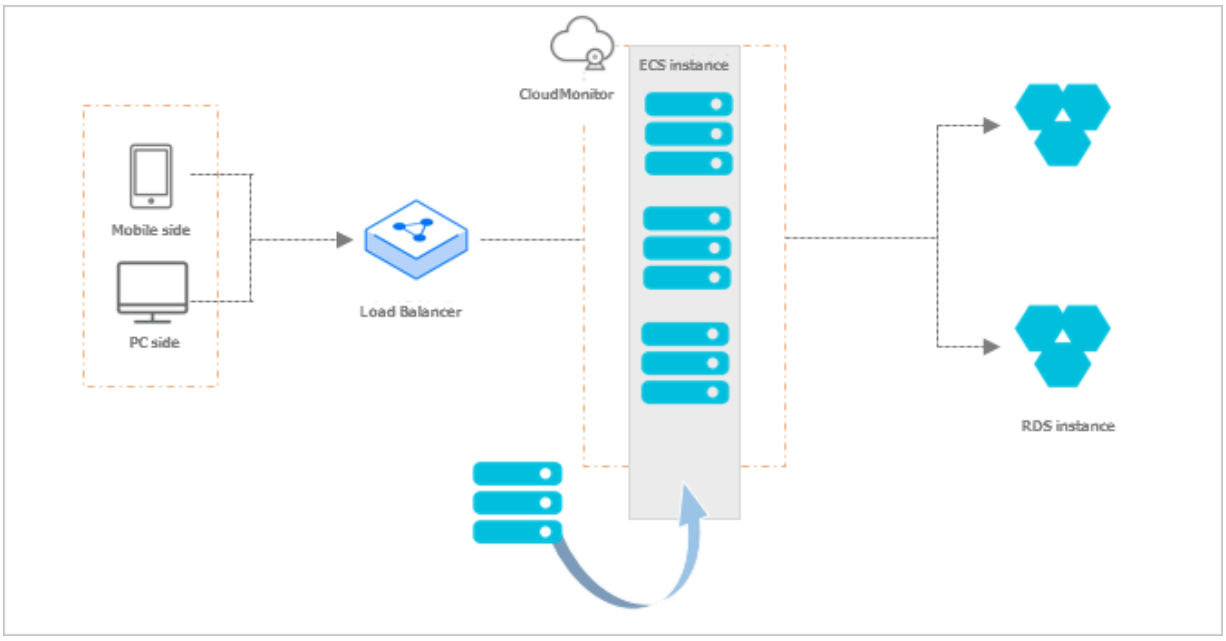
2.3.1.2 Elastic scale-out

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

You can create scheduled tasks to perform automatic scale-out at specified times or configure CloudMonitor to monitor ECS instance usage in real time and perform scale-out based on actual requirements. For example, when CloudMonitor detects that the vCPU utilization of ECS instances in a scaling group exceeds 80%, ESS elastically scales out ECS resources based on user-defined scaling rules. During

the scale-out process, ESS automatically creates ECS instances and adds these ECS instances to the SLB instance and RDS whitelist.

Figure 2-2: Elastic scale-out



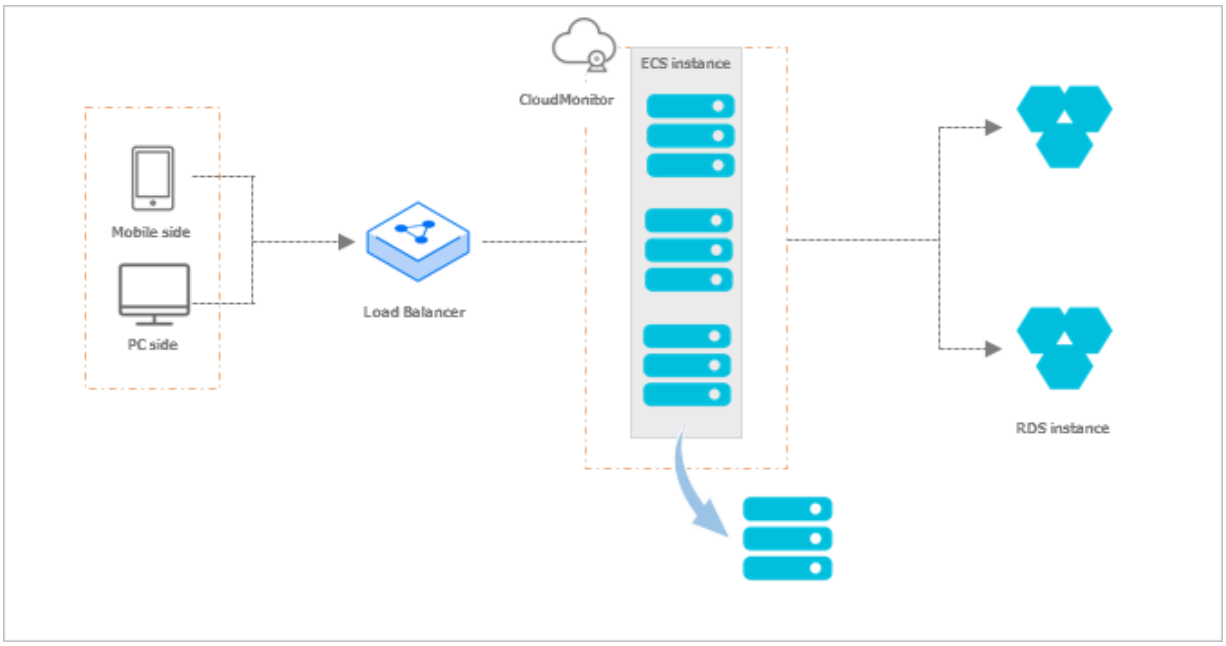
2.3.1.3 Elastic scale-in

When loads on services decrease, ESS automatically releases underlying resources to prevent resource wastage and reduce costs.

You can create scheduled tasks to scale in resources automatically at specified points in time. You can also configure CloudMonitor to monitor ECS instance usage in real time and scale in resources based on actual requirements. For example, when CloudMonitor detects that the vCPU utilization of ECS instances in a scaling group falls below a specified threshold, ESS automatically scales in ECS resources based on user-defined rules. During the scale-in process, ESS releases

ECS instances and removes these ECS instances from the SLB instance and RDS whitelist.

Figure 2-3: Elastic scale-in

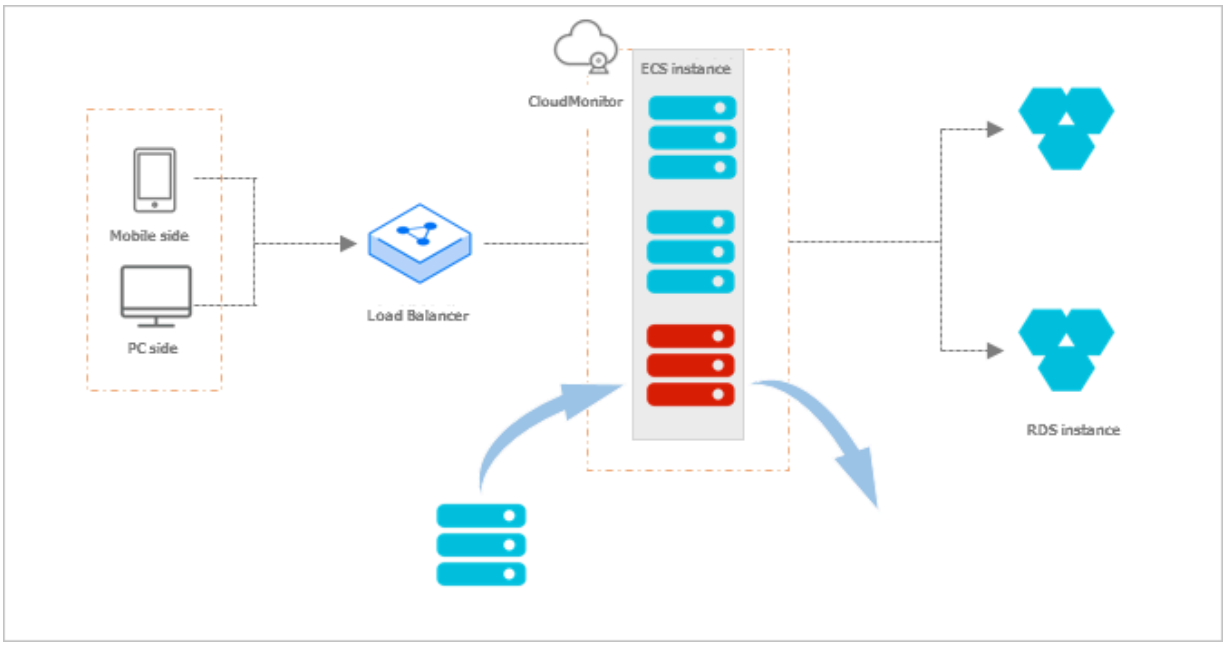


2.3.1.4 Elastic recovery

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

When ESS detects that an ECS instance is not healthy, it automatically releases the unhealthy ECS instance, creates a new ECS instance, and adds the new instance to the SLB instance and RDS whitelist.

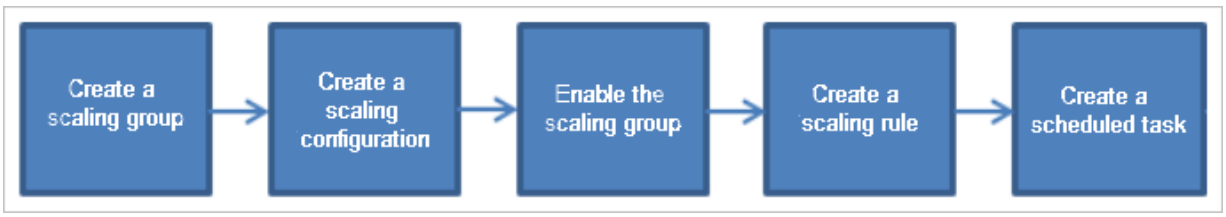
Figure 2-4: Elastic recovery



2.3.2 Function components

To create a complete automatic scaling solution that performs scale-in and scale-out based on actual requirements, you need to create scaling groups, configurations, rules, and scheduled tasks.

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



Scaling group

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

Scaling configuration

A scaling configuration is a template in ESS for creating ECS instances. When creating a scaling configuration, you can specify ECS instance information, such as instance type, image type, storage size, and instance logon key pair. You can also modify an existing scaling configuration as needed.

Scaling rule

A scaling rule defines the specific scaling activity, for example, the number of ECS instances to be added or removed. The following scaling rules are supported:

- **Set to N instances: After this scaling rule is executed, the number of instances in service is changed to N.**
- **Add N instances: After this scaling rule is executed, the number of instances in service is increased by N.**
- **Decrease N instances: After this scaling rule is executed, the number of instances in service is reduced by N.**

Scheduled task

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

3 Object Storage Service (OSS)

3.1 What is OSS?

3.1.1 Basic concepts

Apsara Stack Object Storage Service (OSS) is a secure, cost-effective, and highly reliable storage service that is capable of processing large amounts of data.

It can be considered as immediately available storage solution with unlimited storage capacity. Compared with the user-created server storage, OSS has many outstanding advantages in reliability, security, cost, and data processing capabilities. OSS enables you to store and retrieve a variety of unstructured data objects, such as texts, images, audios, and videos over the network at any time.

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

In OSS, you can:

- **Create a bucket and upload objects to the bucket.**
- **Obtain an object URL from OSS to share or download the object.**
- **Modify properties or metadata of an object, and set the ACL for the object.**
- **Perform basic and advanced OSS tasks through the OSS console.**
- **Perform basic and advanced OSS tasks through the Alibaba Cloud SDKs or by directly calling the RESTful API through your application.**

3.1.2 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 restoration 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**

  Provides the standard RESTful APIs (some compatible with Amazon S3 APIs), a wide range of SDKs, client tools, and console. You can upload, download, retrieve, and manage large amounts of data for websites and mobile apps in the same 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 features such as whitelisting, hotlink protection, and RAM.

3.1.3 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.

3.2 Benefits

Multifunctionality

- **Supports multiple functions: simple upload, form upload, append upload, download, delete, list, and replicate objects, obtain object metadata, and create multipart upload tasks.**
- **Supports bucket-based functions: create, delete, and list objects in a bucket as well as obtain bucket metadata.**
- **Creates a globally unique bucket and supports cross-region replication (CRR) for buckets.**
- **Supports lifecycle management, defines and manages lifecycle rules for all or a subset of objects in a bucket, and changes capacities and ownership.**
- **Supports zone-disaster recovery. In zone-disaster recovery mode, buckets with the same name are replicated. Cluster-based disaster recovery is automatica lly enabled based on configurations made when the cluster is created. In other words, after a primary bucket is created, a secondary bucket with the same name**

is automatically created. Information stored in the primary bucket is automatically synchronized to the secondary bucket.

- Configures static website hosting for your bucket and allows you to use the bucket domain name to access the static website.
- Supports hotlink protection based on the HTTP Referer fields in HTTP headers.
- Supports cross-origin resource sharing (CORS). Supports logging and log analytics in multiple dimensions. You can view access source information.
- Uses the architecture that features redundancy to prevent single point of failures (SPOFs).
- Uploads and downloads large objects, supports multipart upload and range download of large objects, and supports resumable upload, download, and replication.

High performance

Supports the throughput of a cluster that contains tens of thousands of nodes.

Security

Supports ACL for access control. You can configure ACL when creating a bucket and modify the ACL after the bucket is created. The following ACLs are supported: private, public read, and public read/write.

Supports Resource Access Management (RAM) for employees, applications, and systems based on the department architecture. A separate logon password or AccessKey pair is created for each employee, application, or system. By default, RAM users do not have any permissions on OSS resources. You can use RAM to grant permissions to RAM users or use Security Token Service (STS) for temporary access authorization. HTTPS and encryption on the server and client are supported.

Supports the APIs, SDKs, and migration tool to migrate large amounts of data to or from Alibaba Cloud.

Allows multiple types of terminals, web applications, and mobile apps to write data to or read data from OSS directly. Stream input and object input are supported. You can manage static resources such as images, scripts, and videos on websites in the way you manage folders. After objects are uploaded to OSS, you can apply the features provided by the services in the cloud OS to the objects. You can use

services such as audio and video processing, IMG, Batch Compute, and offline processing. This way, you can maximize data values.

Supports hotlink protection to prevent unauthorized access.

Supports Secure Sockets Layer (SSL) to control the read and write permissions on each object.

Combines with the intrusion prevention system to prevent DDoS attacks and HTTP flood attacks to ensure that business works properly.

Supports cross-region replication (CRR) to synchronize data to a specified region in real time for geo-disaster recovery. This way, OSS can protect important data from the impact of extreme disasters and ensures service stability.

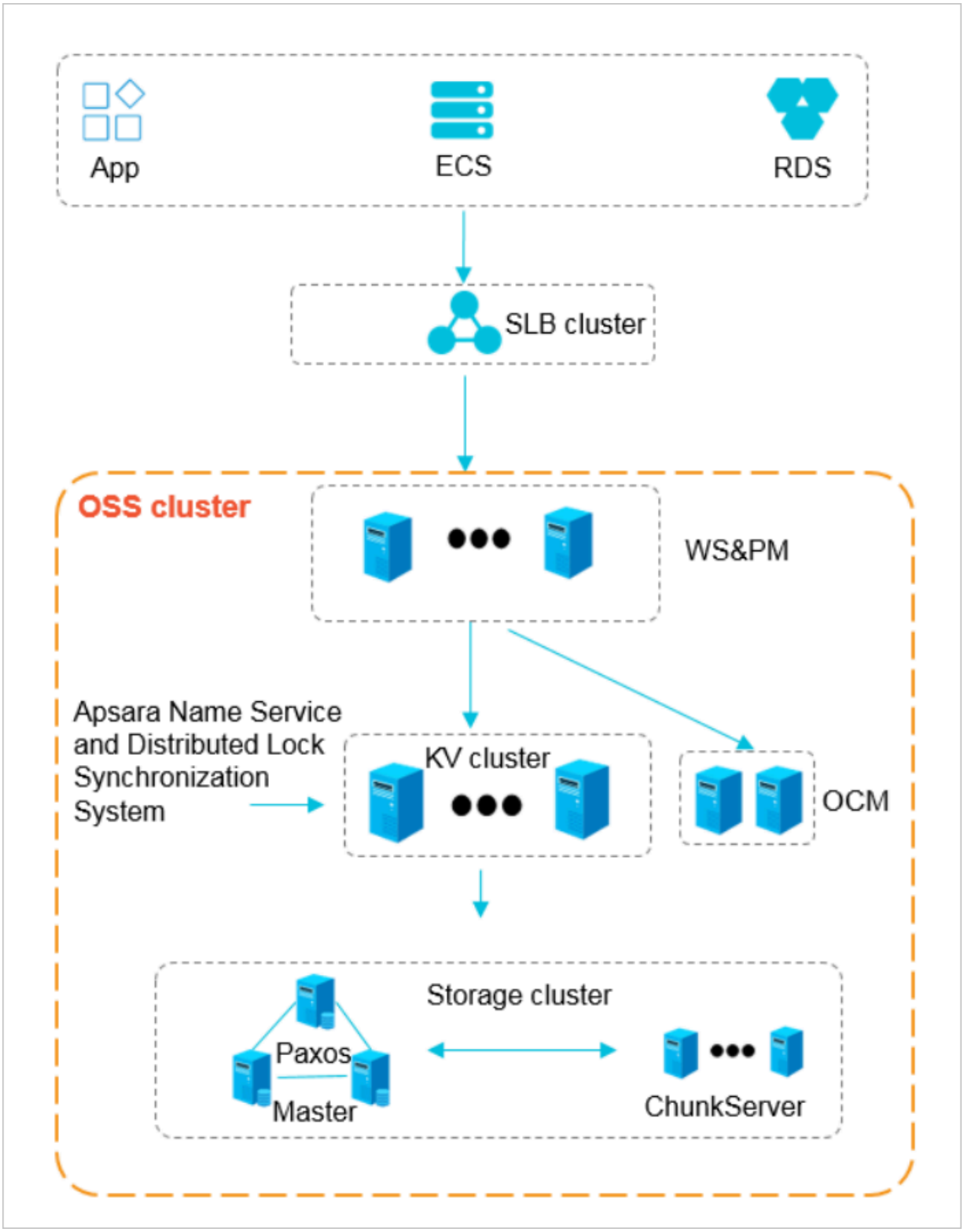
3.3 Architecture

3.3.1 System architecture

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

The following figure shows the architecture of OSS.

Figure 3-1: OSS architecture



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

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

  The partition layer uses keys to query and store structured data. This layer also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point. The partition layer manages indexes of objects, and converts objects to the persistent data objects at the persistent storage layer.
  - **SchedulerX is responsible for naming services and is based on Apsara Name Service and Distributed Lock Synchronization System.**
  - **KV consists of KVMaster and KVServer. KVMaster manages and schedules partitions. KVServer stores indexes and actual data of partitions.**
- **Persistent layer**

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

3.3.2 Data forwarding procedure

The data forwarding procedure from the perspective of user access is as follows:

```
User → RESTful API → SLB-Web server (WS) → Protocol module (PM) → KV
Engine → Distributed storage
```

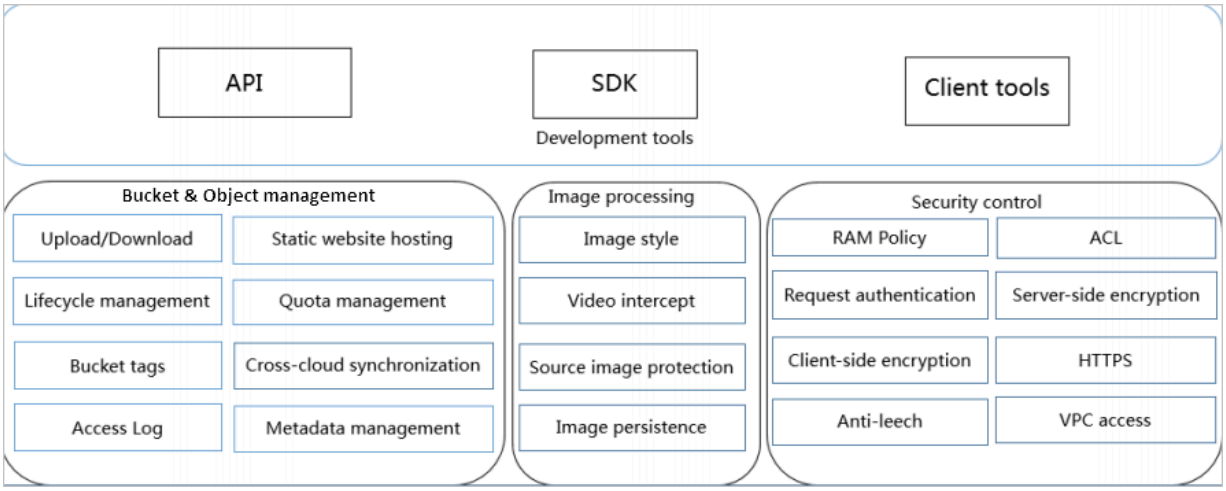
- **A user uses different clients such as browsers or SDKs to initiate a request that complies with the convention of the OSS API to the OSS endpoint. The endpoint**

parses the request and sends it to the LVS VIP of SLB. The back end of the LVS VIP is bound to the actual WS. The request is forwarded to one of the WSs.

- The PM parses the user request. The specific process is as follows: First, the request is authenticated. If the request fails the authentication, the correspond ing error code is returned.
- If the authentication succeeds, the request is parsed as the read/write operation on KV Engine and enters the partition layer.
- The partition layer uses keys to query and store structured data. This layer also supports sporadic bursts of requests. When a service has to run on a different physical server due to a change to the service coordination cluster, the KV cluster can coordinate and find the access point.
- The data stored in KV Engine of the partition layer is written to the persistent storage layer.
- The large-scale distributed file system is deployed at the persistent storage layer . Metadata is stored in masters. A distributed message consistency protocol (or Paxos) is adopted between masters to ensure the metadata consistency. This way , efficient distributed file storage and access are achieved. This method ensures that three copies of data are stored in the system and that the system can recover from any software or hardware faults.

3.4 Features and principles

3.4.1 Components



OSS is a storage solution that is built on the Apsara system.

OSS consists of three modules: access layer, application layer, and infrastructure layer.

- **Access layer: the API, SDKs, and Apsara Stack console**
- **Application layer: bucket and object management, IMG, and security modules**
- **Infrastructure layer: Apsara Distributed File System, Job Scheduler, and Apsara Name Service and Distributed Lock Synchronization System**

3.4.2 Features

Bucket and object management

- **Bucket overview**

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

- **Create or delete buckets**

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

  The following scenarios may exist when you create a bucket:

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

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

  - **The bucket exists.**
  - **You have the permissions to delete the bucket.**
  - **The bucket contains no data.**

- **List all objects in a bucket**

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

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

- **Upload or delete objects**

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

- **Obtain the content or metadata of objects**

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

- **Access objects**

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

Security control

- **Set and query the ACL of a bucket**

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

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

- **Logging and monitoring**

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

- **Hotlink protection**

  OSS provides hotlink protection to prevent unauthorized domain names from accessing your OSS data. You can use the OSS console to configure the Referer whitelist and specify whether to allow an request that includes an empty Referer field. For example, you can add http://www.aliyun.com to the Referer whitelist for a bucket named oss-example. Then, only requests with a Referer of http://www.aliyun.com can access the objects in the oss-example bucket.

3.4.3 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.

4 Table Store

4.1 What is Table Store?

4.1.1 Technical background

Data features in the data technology (DT) era

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

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

Challenges of traditional IT software solutions

Traditional IT software solutions present the following trends and challenges:

- **Scalability**

  Traditional software, such as relational databases, is incapable of handling such fast-growing data. It bottlenecks data write throughput and access efficiency. With traditional database solutions, the whole process is complex. Databases and tables are partitioned manually and statically. This method requires large amounts of maintenance. In particular scenarios where nodes are added to increase the storage capacity, there is a need to repartition and migrate existing

data. During this process, it is difficult to guarantee service performance, stability, and availability. The whole process is complex.

- Data model changes

  Data in traditional databases is processed based on a schema. The number of columns in data is fixed and not changed often. Frequent changes to the table schema and column count affect service availability. Therefore, traditional solutions are incapable of handling the increasing volumes of loosely structured data from Internet applications.

- Quick scaling

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

- O&M guarantees

  With traditional software solutions, services are recovered when hardware (network devices or disks) failures occur. Hardware replacement, software upgrades, and configuration tuning and updates need to be performed manually . To ensure that applications are not aware of these processes and avoid deterioration of service availability, users need a special engineering team to achieve system O&M. Therefore, workloads caused from recruitment and fund investment bring a huge challenge to fast-developing enterprises.

- Computing bottlenecks

  The current business system uses Online Transaction Processing (OLTP) to process and analyze data in relational databases such as MySQL and Microsoft SQL Server. These relational databases are adept at transaction processing. They maintain high consistency and atomicity in data operations, and support frequent data insertion and modification. However, when the volume of data exceeds the processing capabilities of the system, such as when the number of data records reaches tens of millions, or complex calculations are required, OLTP database systems are no longer sufficient.

4.1.2 Table Store technologies

Table Store is a NoSQL data storage service built on the Apsara system that is developed by Alibaba Cloud. Table Store partitions tables and schedules data partitions to different nodes to improve scalability. When a hardware failure occurs, Table Store quickly detects the faulty node by using the heartbeat mechanism and migrates data partitions from the defective node to a healthy node to continue services, achieving rapid service recovery.

Data partitioning and load balancing

The first primary key column in each row of a table is called the partition key. The system partitions a table into multiple partitions based on the range of the partition key. When the data in a partition exceeds a certain size, the partition is automatically split into two smaller partitions. The data and access loads are distributed to two partitions. The partitions are scheduled to different nodes. Eventually, the linear scalability of the single-table data scale and access loads is achieved.

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

Automatic recovery from single point of failures (SPOFs)

In the storage engine of Table Store, each node serves a number of data partitions in different tables. The Master service role monitors partition distribution and scheduling, and the health of each service node. If a service node fails, the Master service role migrates data partitions from this faulty node to other healthy nodes. The migration is logically performed, and does not involve physical entities, so services can rapidly recover from SPOFs.

Technical indicator: SPOFs affect services of some data partitions only and services can recover within several minutes.

Zone-disaster recovery and geo-disaster recovery

To meet business security and availability requirements, Table Store provides active-standby cluster-based zone-disaster recovery and geo-disaster recovery. Disaster recovery supports instance-based recovery. Any table operation on the primary instance, including insertion, update, or deletion, is synchronized to the table of the same name in the secondary instance. The duration of data synchronization between the primary and secondary instances depends on the network environment of the primary and secondary clusters. In the ideal network

environment, the synchronization latency is in milliseconds. Before the manual failover, you must stop resource access to the primary cluster and wait for all data to be completely backed up. Do not perform any failover operations in the hour after a recent failover.

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

Technical indicator: The RTO of Table Store is less than 2 minutes, the RPO is less than 5 minutes, and the RCO is 1.

4.2 Benefits

Scalability

There is no upper limit to the amount of data that can be stored in Table Store tables. As data increases, Table Store adjusts partitions to provide more storage space for tables and improve the capability of handling access request bursts.

High performance

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

Reliability

Table Store provides high data reliability. It stores multiple data copies and restores data when some copies become invalid.

High availability

Through automatic failure detection and data migration, Table Store shields applications from host- and network-relevant hardware faults to achieve high availability.

Ease of management

Table Store automatically performs complex O&M tasks, such as the management of data partitions, software and hardware upgrades, configuration updates, and cluster scale-out.

Access security

Table Store provides multiple permission management mechanisms. It verifies and authenticates the identity of the request to prevent unauthorized data access, improving the data security.

Strong consistency

Table Store ensures strong data consistency for data writes. A successful write operation indicates that the data is written to three copies and stored in disks. Applications can read the latest data immediately.

Flexible data models

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

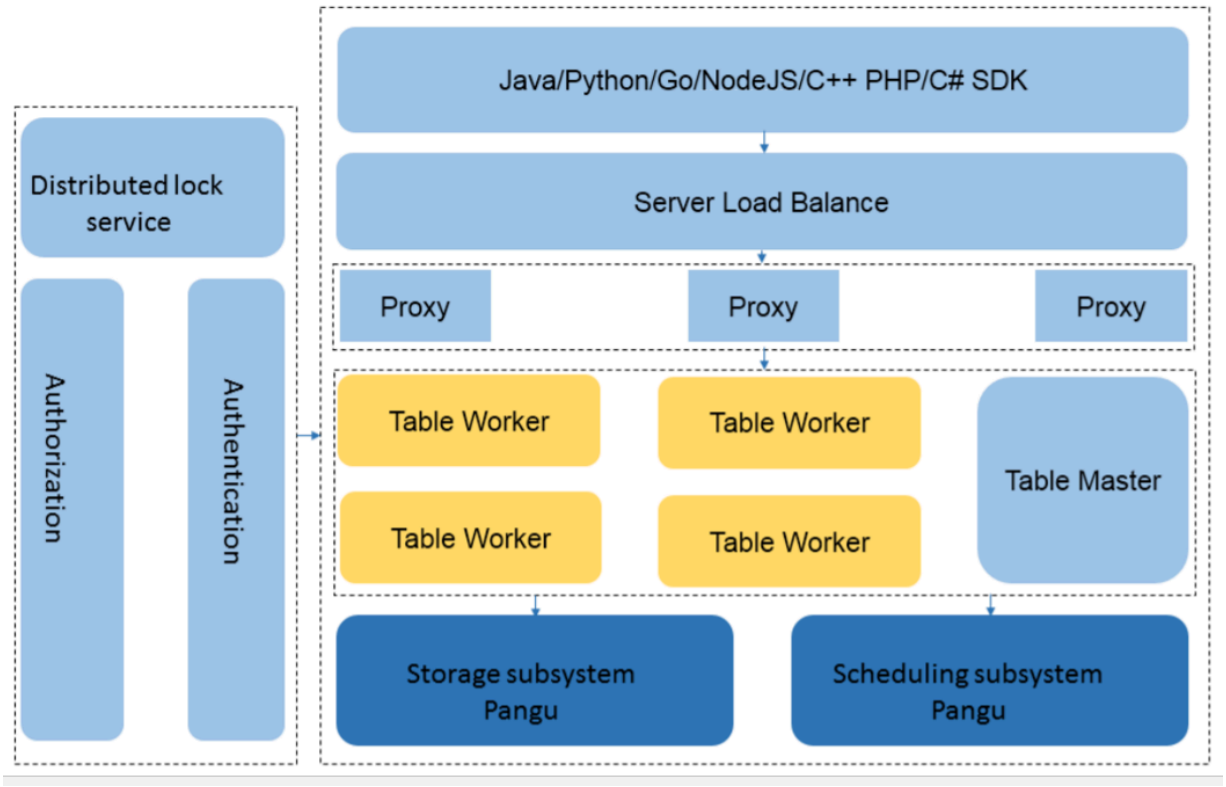
Monitoring integration

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

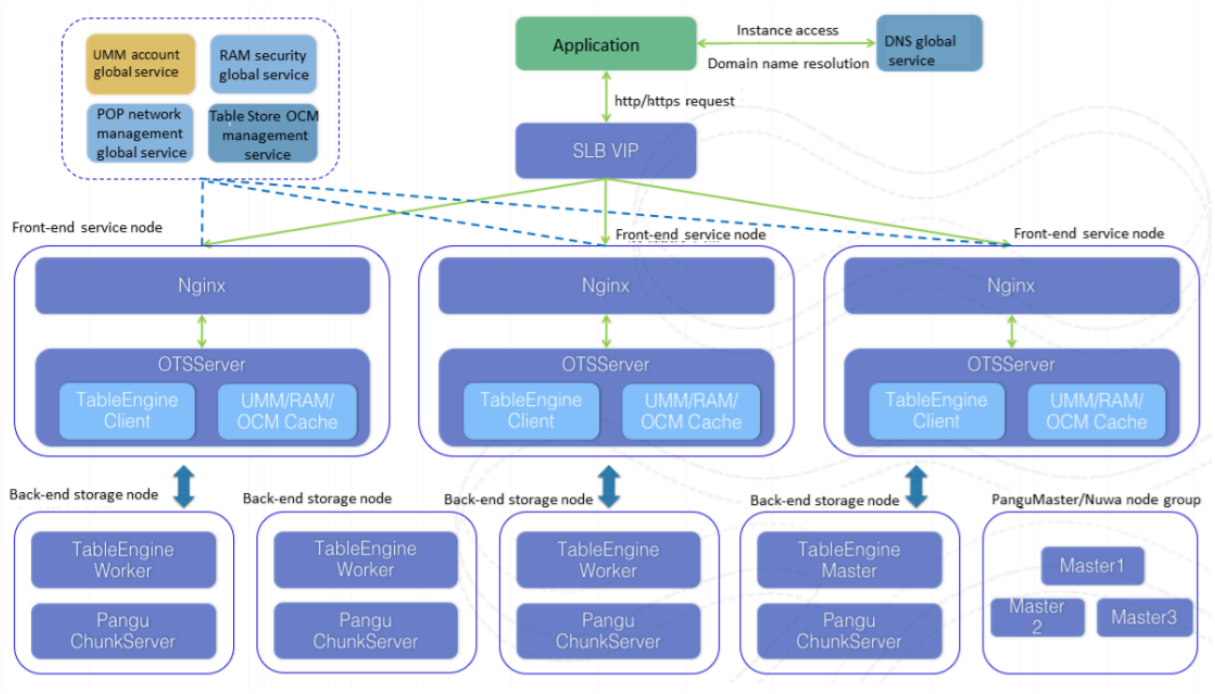
4.3 Architecture

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

The following figure shows the basic architecture of Table Store.



The following figure shows the detailed architecture of Table Store.



- **The top layer is the protocol access layer. SLB distributes user requests to various proxy nodes. The proxy nodes receive requests that are sent through the RESTful protocol and implement security authentication. If the authentica tion succeeds, the user requests are forwarded to the corresponding data engine**

based on the value of the first primary key column for further operations. If the authentication fails, an error message is directly returned to the user.

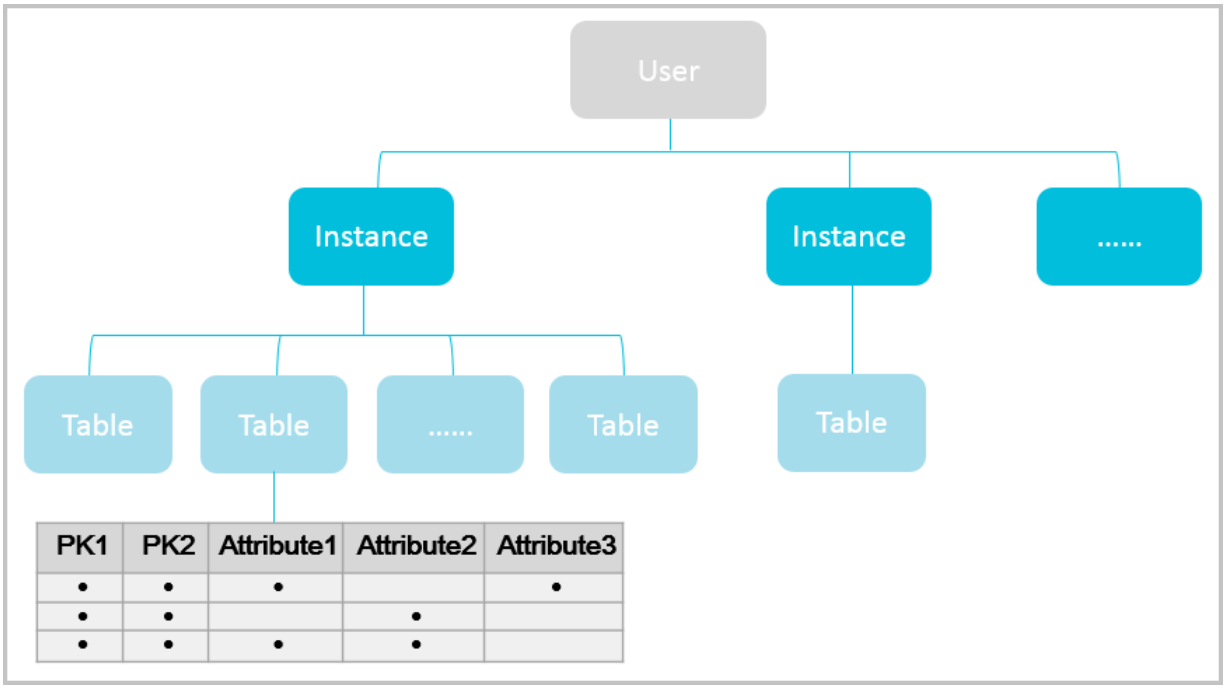
- Table Worker is the data engine layer that processes structured data. It uses a primary key to search for or store data. Table Worker supports large-scale access request bursts.
- The bottom layer is the storage layer. Apsara Distributed File System is deployed at this layer. Metadata is stored in Master server roles. The distributed message consistency protocol Paxos is adopted between Master service roles to ensure metadata consistency. In this scenario, efficient distributed file storage and access are achieved. This method guarantees three copies of data stored in the system and system recovery from any hardware or software faults.

4.4 Features

4.4.1 Users and instances

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

Figure 4-1: User and instance architecture



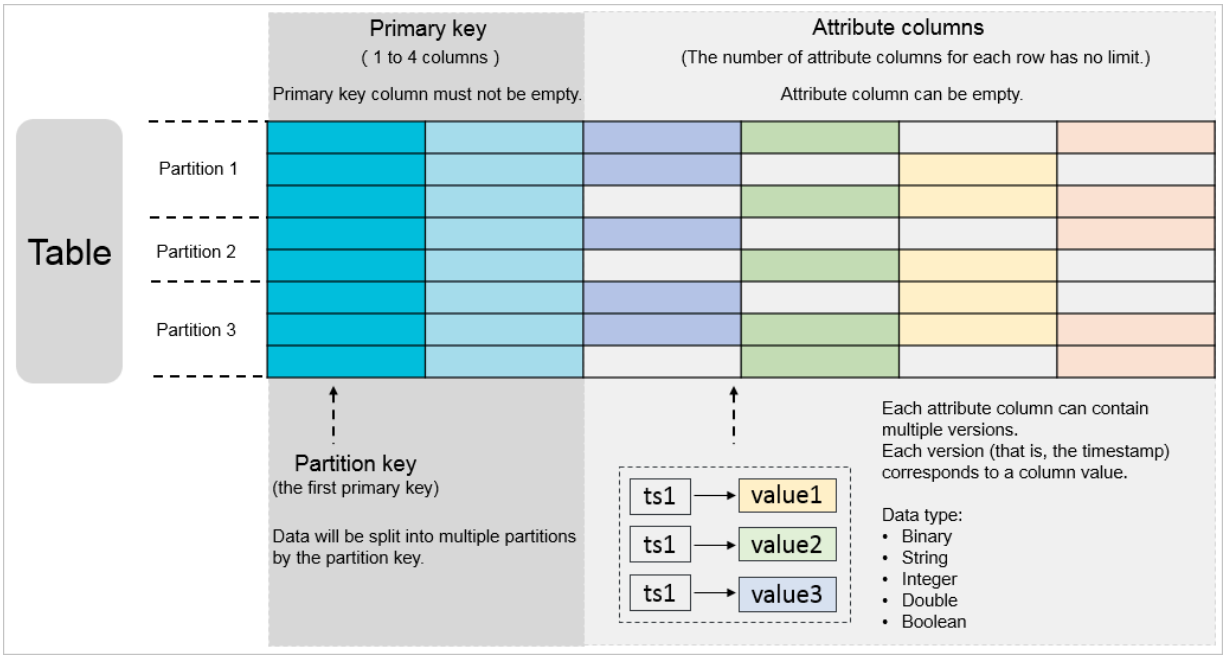
- Users can log on with an Apsara Stack tenant account.
- User operations can be audited in fine granularity.

- **Users organize resources through instances. A user can create multiple instances and use each instance to create and manage multiple data tables.**
- **An instance is the basic unit of multi-tenant isolation.**
- **User permissions can vary with their roles.**

4.4.2 Data tables

The following figure shows the data table structure.

Figure 4-2: Data table structure



- **A data table is the basic unit of resource allocation.**
- **A table is a set of rows. A row consists of primary key columns and attribute columns.**
- **A table partitions data based on the size of the first primary key column.**
- **All rows in a table must have the same quantity of primary key columns with the same names.**
- **The quantity of attribute columns in a row is variable. So are the names and data types of the attribute columns.**
- **There is no limit to the number of attribute columns contained in a row. However, the maximum number of attribute columns where each request can write data to is 1,024.**
- **A table can contain over hundreds of billions of rows of data.**
- **A table can store PBs of data.**

4.4.3 Data partitioning

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

4.4.4 Common commands and functions

Commands

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

Functions

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

4.4.5 Authorization and access control

Table Store permissions

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

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

Apsara Stack Management Console-based permissions

- **Account logons and authentication through Apsara Stack Management Console**
- **Instance creation, management, and deletion functions through GUI**
- **Table creation, management, deletion, and reserved read/write throughput adjustment functions through GUI**
- **Display of table-level monitoring information**

5 ApsaraDB for RDS

5.1 What is ApsaraDB for RDS?

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

ApsaraDB RDS for MySQL

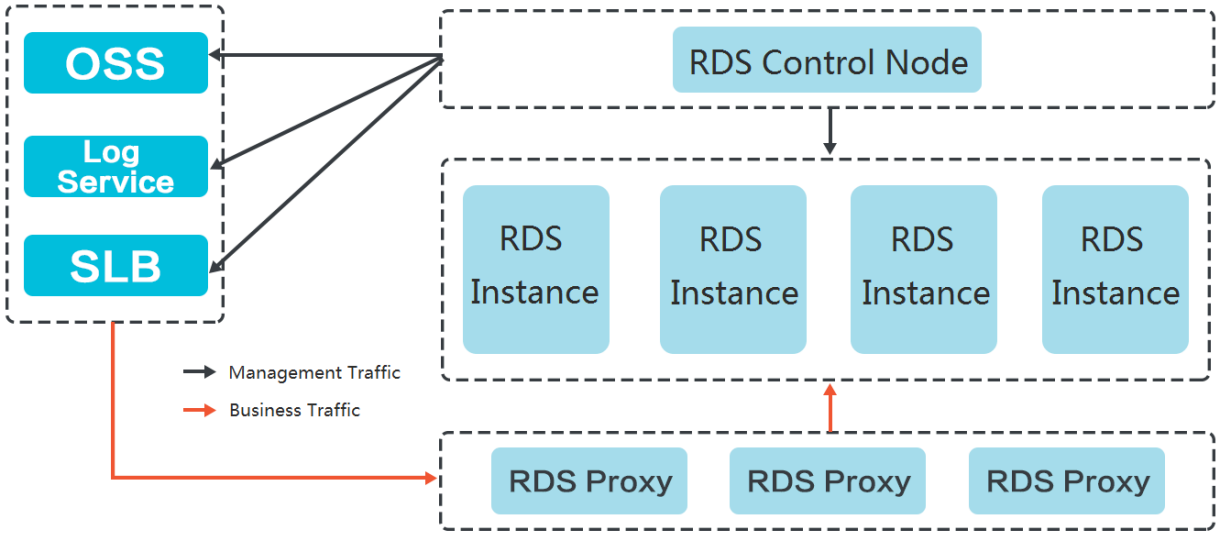
Originally based on a branch of MySQL, ApsaraDB RDS for MySQL is a tried and tested solution for handling high-volume concurrent traffic during Double 11, providing excellent performance. ApsaraDB RDS for MySQL provides whitelist configuration, backup and restoration, transparent data encryption, data migration , and management for instances, accounts, and databases.

ApsaraDB RDS for MySQL also provides read-only instances. In scenarios where RDS has a small number of write requests but a large number of read requests, you can create read-only instances to scale the reading capability and increase the application throughput.

5.2 Architecture

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

Figure 5-1: RDS system architecture

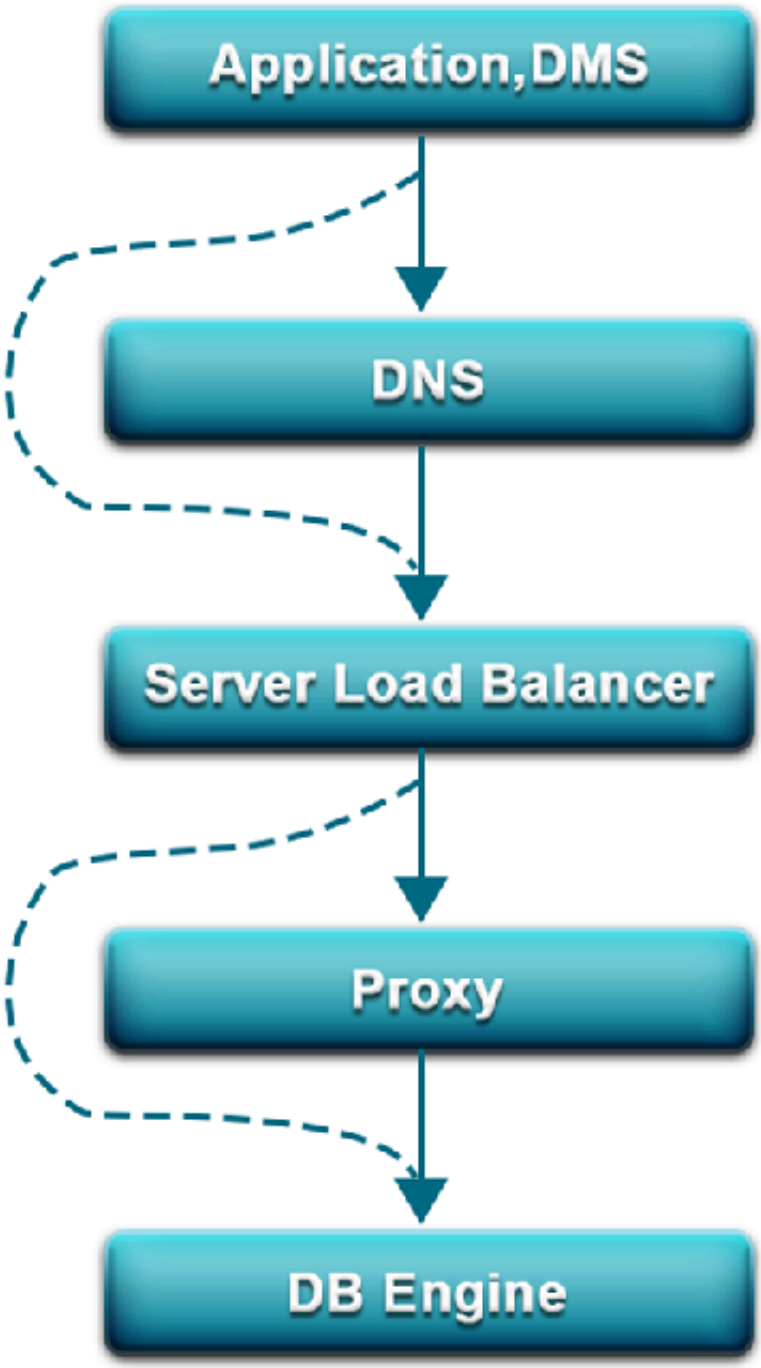


5.3 Features

5.3.1 Data link service

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

Figure 5-2: RDS data link service



DNS

The DNS module can dynamically resolve domain names to IP addresses. Therefore , IP address changes do not affect the performance of RDS instances.

For example, the domain name of an RDS instance is test.rds.aliyun.com, and its corresponding IP address is 10.1.1.1. The instance can be accessed when either test.rds.aliyun.com or 10.1.1.1 is configured in the connection pool of a program.

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

SLB

The SLB module provides both the internal IP address and public IP address of an RDS instance. Therefore, server changes do not affect the performance 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 the hot standby state with the IP address 192.168.0.2 will take over for the initial server. In this case, the SLB module will redirect all traffic destined for 10.1.1.1 to 192.168.0.2, and the RDS instance will continue to provide services normally.

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.**

DB Engine

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

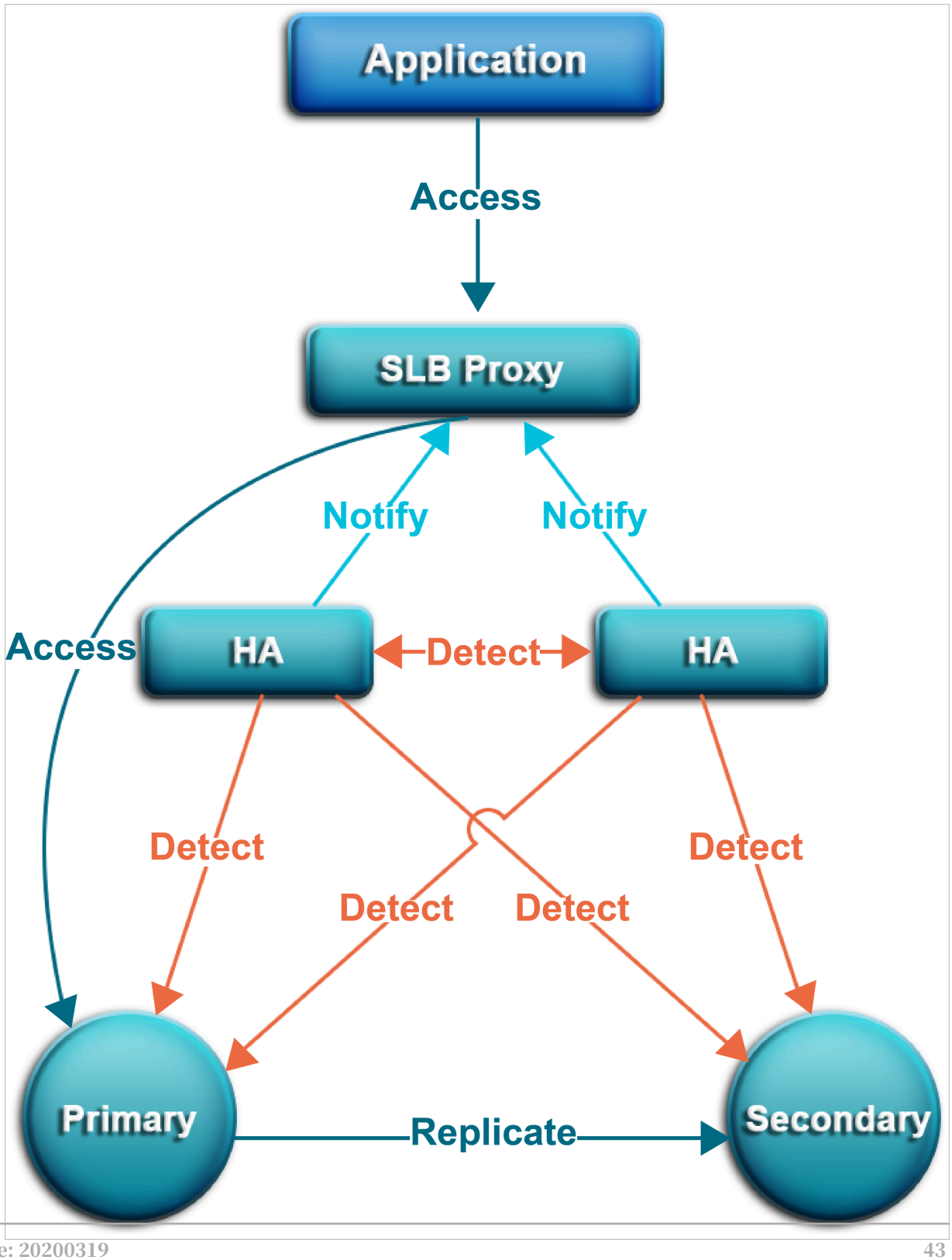
Table 5-1: RDS database protocols

RDBMS	Version
MySQL	5.6 or 5.7 (including read-only instances)

5.3.2 High-availability service

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

Figure 5-3: RDS HA service



Detection

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

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

Repair

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

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

Notice

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

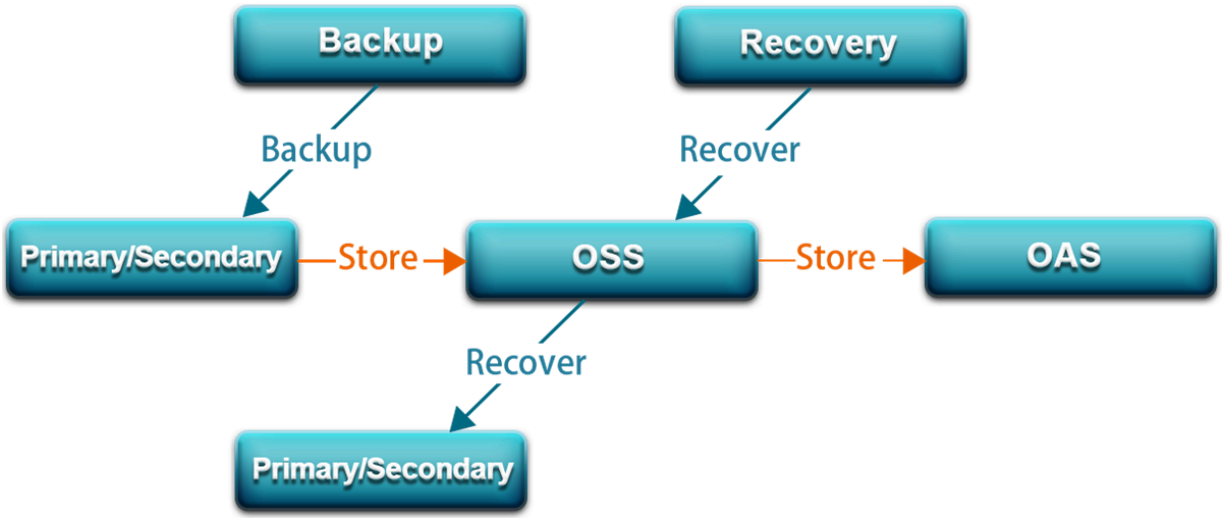
For example, the Detection module discovers problems with the primary node and instructs the Repair module to resolve these problems. If the Repair module fails to resolve a problem, it instructs the Notice module to perform traffic switchover. The Notice module forwards the switching request to the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

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

5.3.3 Backup service

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

Figure 5-4: RDS backup service



Backup

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

Recovery

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

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

Storage

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

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

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

5.3.4 Monitoring service

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

Service

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

Network

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

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

OS

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

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

Instance

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

- **Instance availability information**
- **Instance capacity and performance metrics**
- **Instance SQL execution records**

5.3.5 Scheduling service

The scheduling service allocates resources and manages instance versions.

Resource

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

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

5.3.6 Migration service

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

DTS

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

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

6 AnalyticDB for PostgreSQL

6.1 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.**

6.1.1 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.**

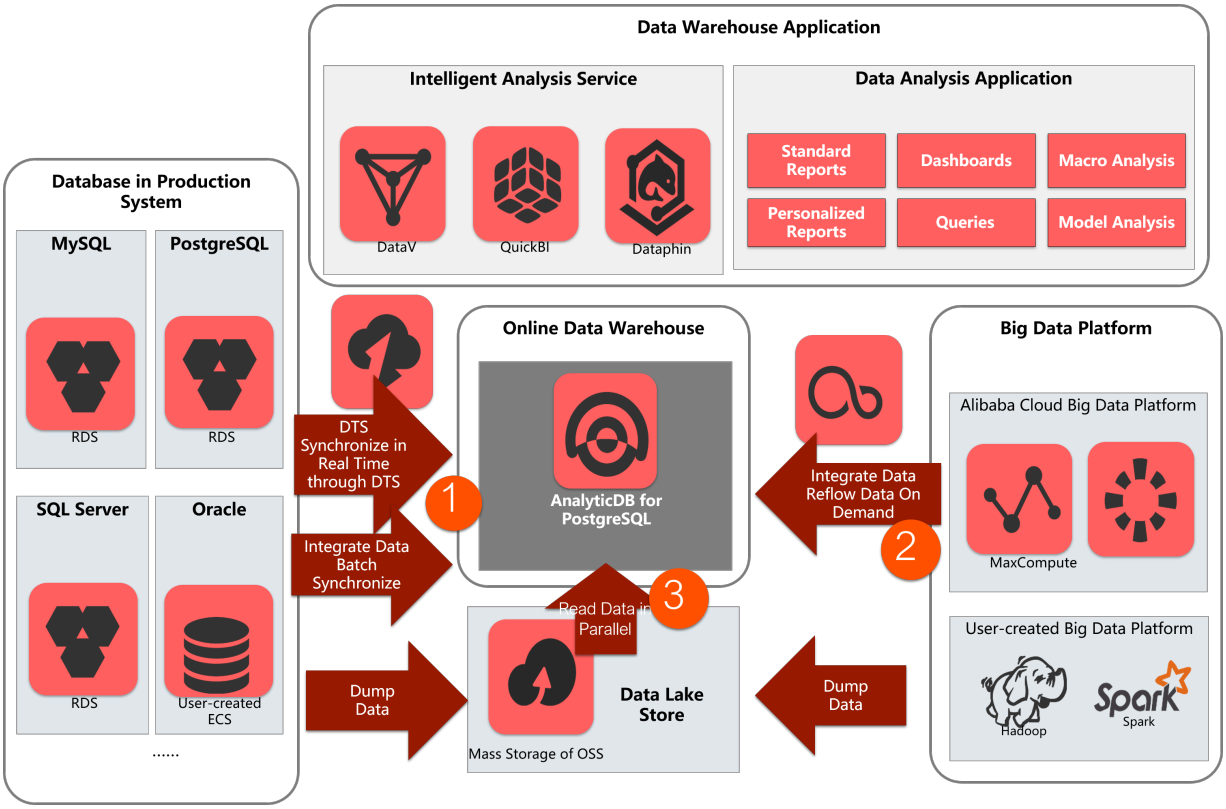
- **Multi-model data analysis**

  AnalyticDB for PostgreSQL provides the following benefits for processing 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. It can also process and analyze semi-structured data such as logs.**

Typical scenarios

AnalyticDB for PostgreSQL is applicable to the three following scenarios:



- **Data warehousing service**

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

- **Big data analytics platform**

  You can 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.

6.2 Benefits

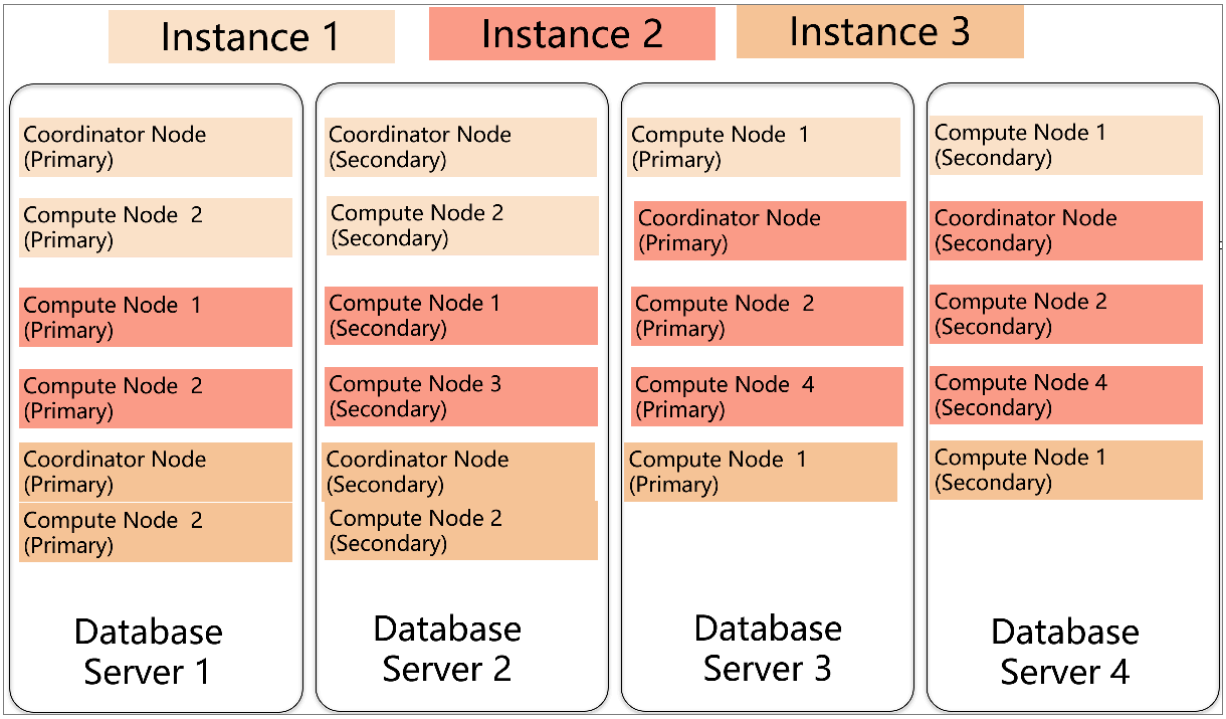
Real-time analysis	**Built on the MPP architecture for horizontal scaling and PB /s data processing. AnalyticDB for PostgreSQL supports the leading vector computing feature and intelligent indexes of column store. It also supports the CASCADE-based SQL optimizer to make complex queries without the need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

6.3 Architecture

Physical cluster architecture

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

Figure 6-1: Physical cluster architecture



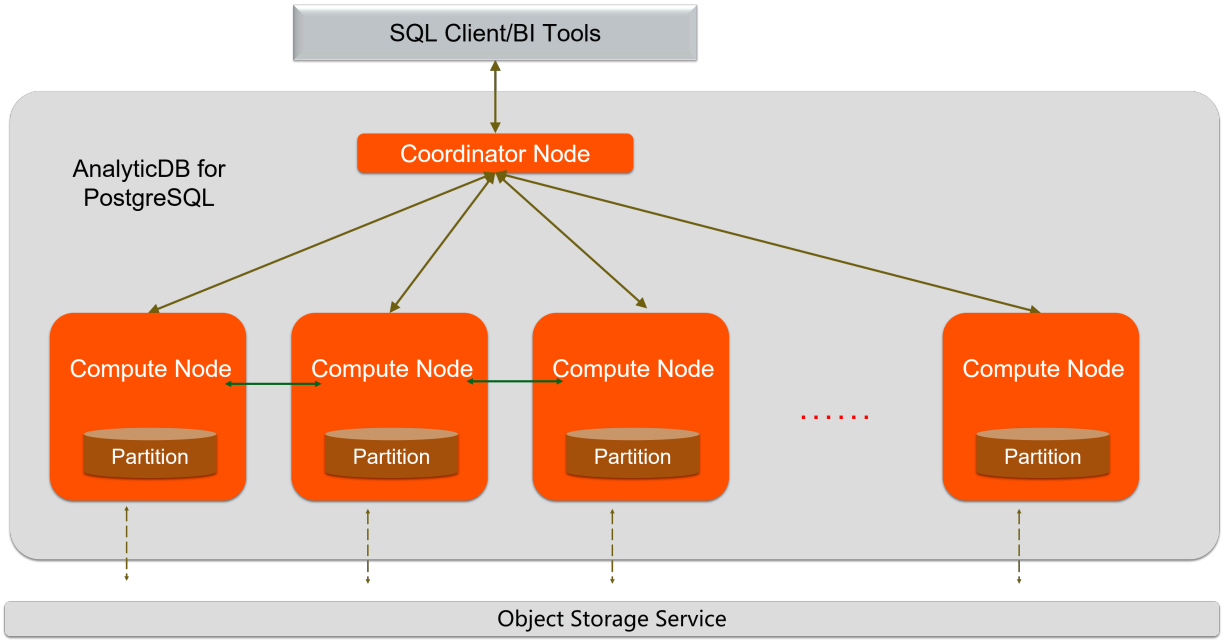
You can create multiple instances within a physical cluster of AnalyticDB for PostgreSQL. Each cluster includes two components: the coordinator node and the compute node.

- **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 6-2: 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 an application, the coordinator node parses and optimizes SQL statements to generate a distributed execution plan. After the coordinator node sends the execution plan to the compute nodes, the compute nodes will perform an MPP execution of the plan.

6.4 Features

6.4.1 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.

6.4.2 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.

6.4.3 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.

6.4.4 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.

6.4.5 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.

6.4.6 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.

Data type	Alias	Storage size	Range	Description
bigint	int8	8 bytes	-9223372036854775808 to 9223372036854775807	Large-range integer
bigserial	serial8	8 bytes	1 to 9223372036854775807	Large auto-increment integer

Data type	Alias	Storage size	Range	Description
bit [(n)]	None	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	None	32 bytes	((x1,y1),(x2,y2))	A rectangular box on a plane, not allowed in distribution key columns
bytea	None	1 byte + binary string	Sequence of octets	Variable-length binary string
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	None	12 or 24 bytes	None	IPv4 and IPv6 networks
circle	None	24 bytes	<(x,y),r> (center and radius)	A circle on a plane , not allowed in distribution key columns
date	None	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	8 bytes	Precise to 15 decimal digits	Variable precision, inexact
	float			
inet	None	12 or 24 bytes	None	IPv4 and IPv6 hosts and networks
Integer	int or int4	4 bytes	-2.1E+09 to + 2147483647	Typical choice for integer

Data type	Alias	Storage size	Range	Description
interval [(p)]	None	12 bytes	-178000000 years to 178000000 years	Time span
json	None	1 byte + JSON size	JSON string	Unlimited variable length
lseg	None	32 bytes	((x1,y1),(x2,y2))	A line segment on a plane, not allowed in distribution key columns
macaddr	None	6 bytes	None	Media Access Control (MAC) address
money	None	8 bytes	-92233720368547758.08 to +92233720368547758.07	Currency amount
path	None	16+16n bytes	[(x1,y1),...]	A geometric path on a plane, not allowed in distribution key columns
point	None	16 bytes	(x,y)	A geometric point on a plane, not allowed in distribution key columns
polygon	None	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
text	None	1 byte + string size	Variable-length string	Unlimited variable length

Data type	Alias	Storage size	Range	Description
time [(p)] [without time zone]	None	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]	None	8 bytes	4,713 BC to 294,277 AD	Date and time
timestamp [(p)] with time zone	timestamptz	8 bytes	4,713 BC to 294,277 AD	Date and time (with time zone)
xml	None	1 byte + XML size	Variable-length XML string	Unlimited variable length

- For more information about the supported standard SQL syntax, see *SQL syntax*.

7 KVStore for Redis

7.1 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.

7.1.1 Scenarios

Game industry applications

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

Scenario 1: KVStore for Redis works as a storage database

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

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

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

Reliability of the KVStore for Redis service is vital to your business. If the KVStore for Redis service is unavailable, the backend database is overloaded when processing connections to your application. KVStore for Redis provides a two-node

hot standby architecture to ensure high availability and reliability of services. The primary node provides services for your business. If this node fails, the system automatically switches services to the secondary node. The complete failover process is transparent.

Live video applications

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

Two-node hot standby ensures high availability

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

Cluster editions eliminate the performance bottleneck

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

Easy scaling relieves pressure at peak hours

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

E-commerce industry applications

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

Scenario 1: rapid online sales promotion systems

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

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

Scenario 2: counter-based inventory management systems

In this scenario, you can store inventory data in an RDS database and save count data to corresponding fields in the database. In this way, the KVStore for Redis

instance reads count data, and the RDS database stores count data. KVStore for Redis is deployed on a physical server. Based on solid-state drive (SSD) high-performance storage, the system can provide a high-level data storage capacity.

7.2 Benefits

High performance

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

Elastic scaling

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

Resource isolation

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

Data security

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

High availability

- **Primary/secondary structure: each instance runs in this structure to eliminate the possibility of single points of failure (SPOFs) and guarantee high availability.**
- **Automatic detection and recovery of hardware faults: the system automatica lly 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.**

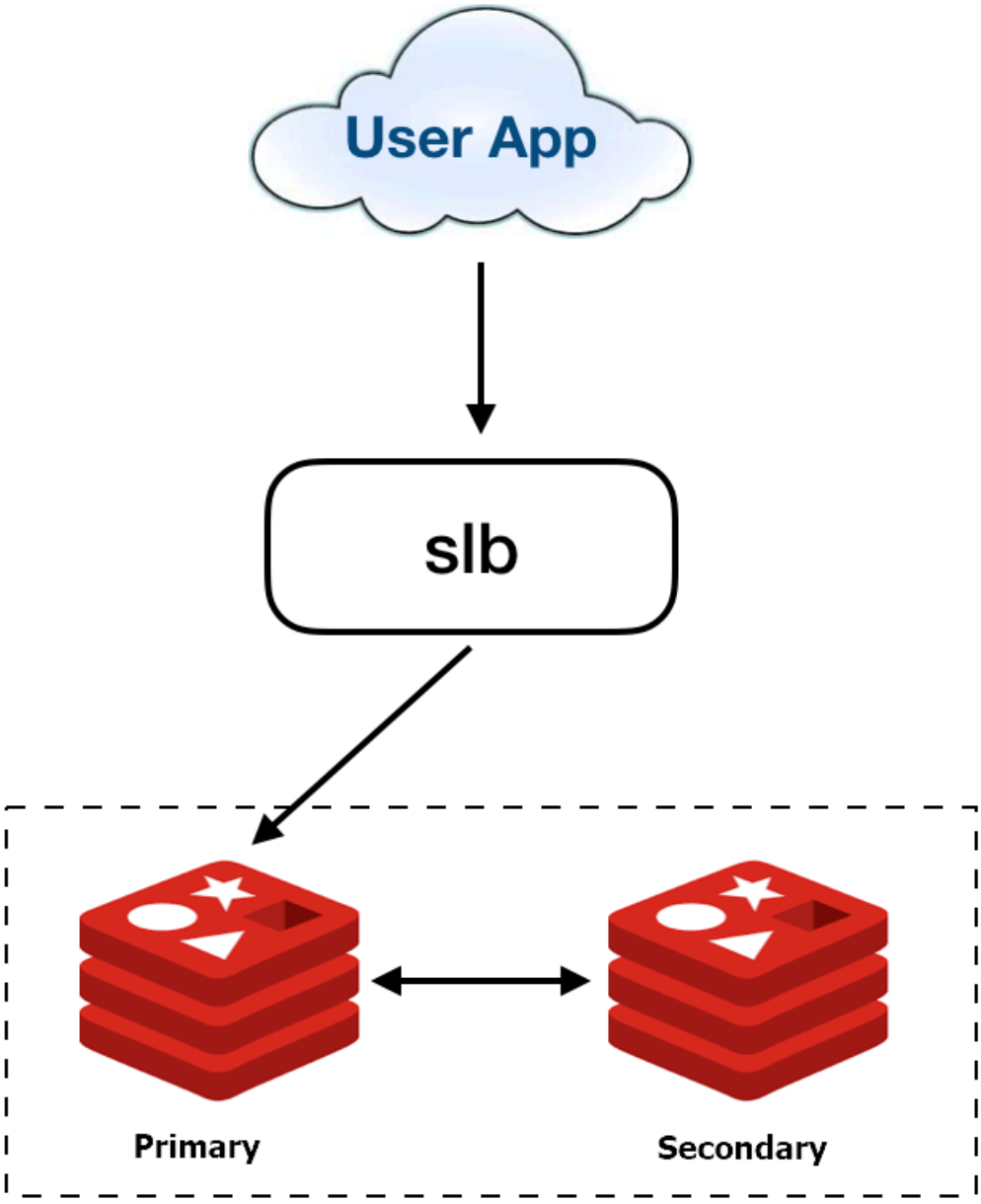
7.3 Architectures

7.3.1 Overall system architecture

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

Primary/secondary architecture

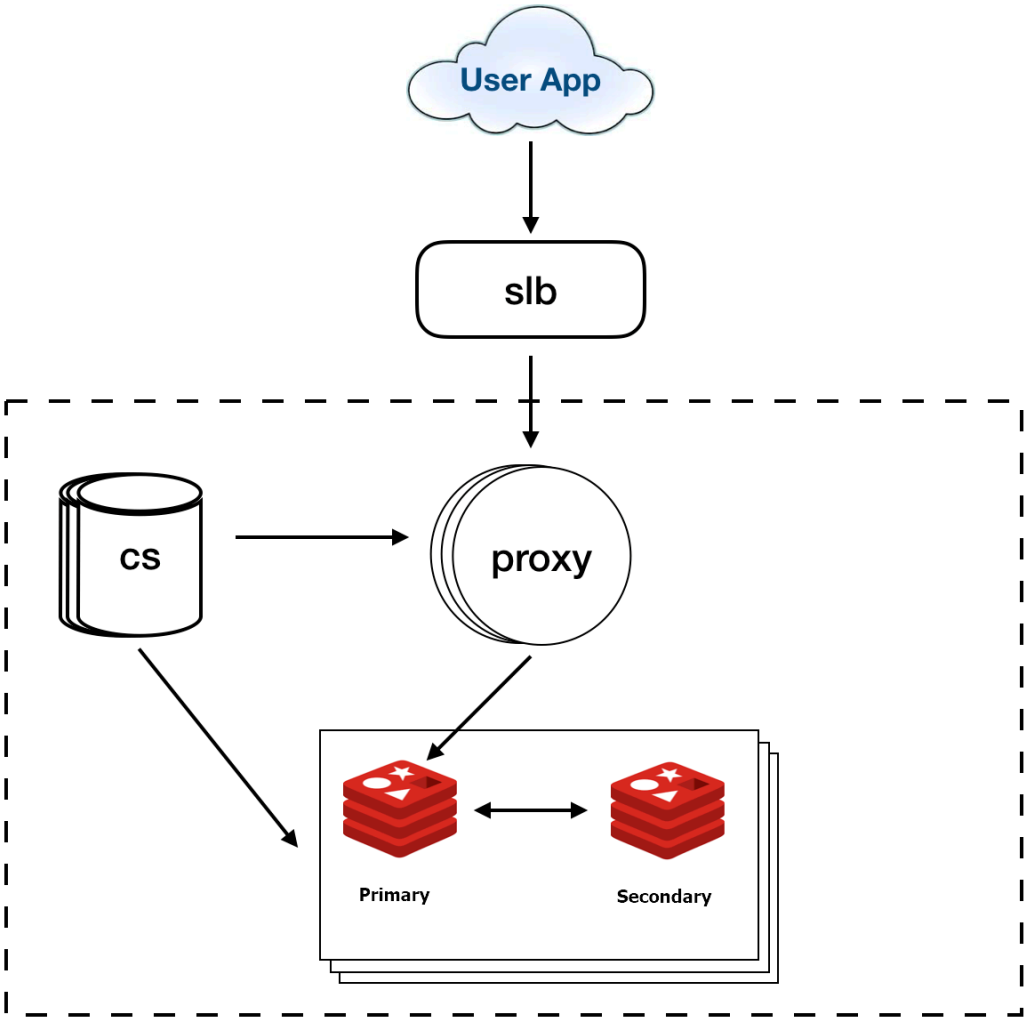
The following figure shows the primary/secondary architecture.



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

Cluster architecture

The following figure shows the cluster architecture.



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

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

7.3.2 Components

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

redis-config

The redis-config (cs) component stores the metadata and topology information of the cluster, and performs cluster operations and maintenance. The cs component

keeps checking heartbeat messages with the Redis and proxy components, and synchronizes metadata and topology information of clusters to redis and proxy.

redis-proxy

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

Redis kernel

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

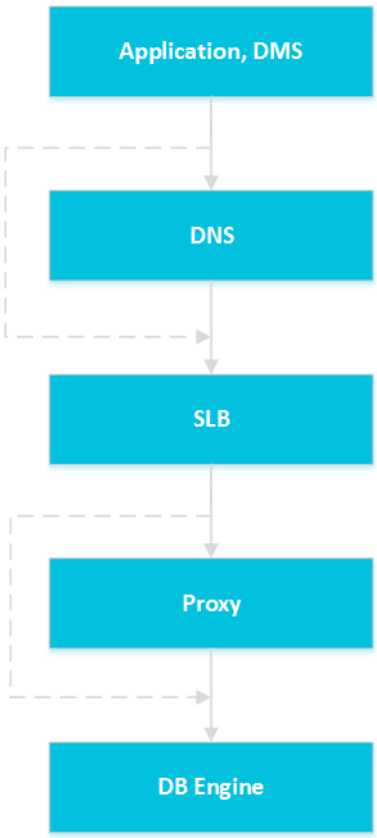
7.4 Features

7.4.1 Data link service

7.4.1.1 Overview

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

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



7.4.1.2 DNS

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

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

7.4.1.3 SLB

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

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

7.4.1.4 Proxy

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

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

7.4.1.5 DB Engine

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

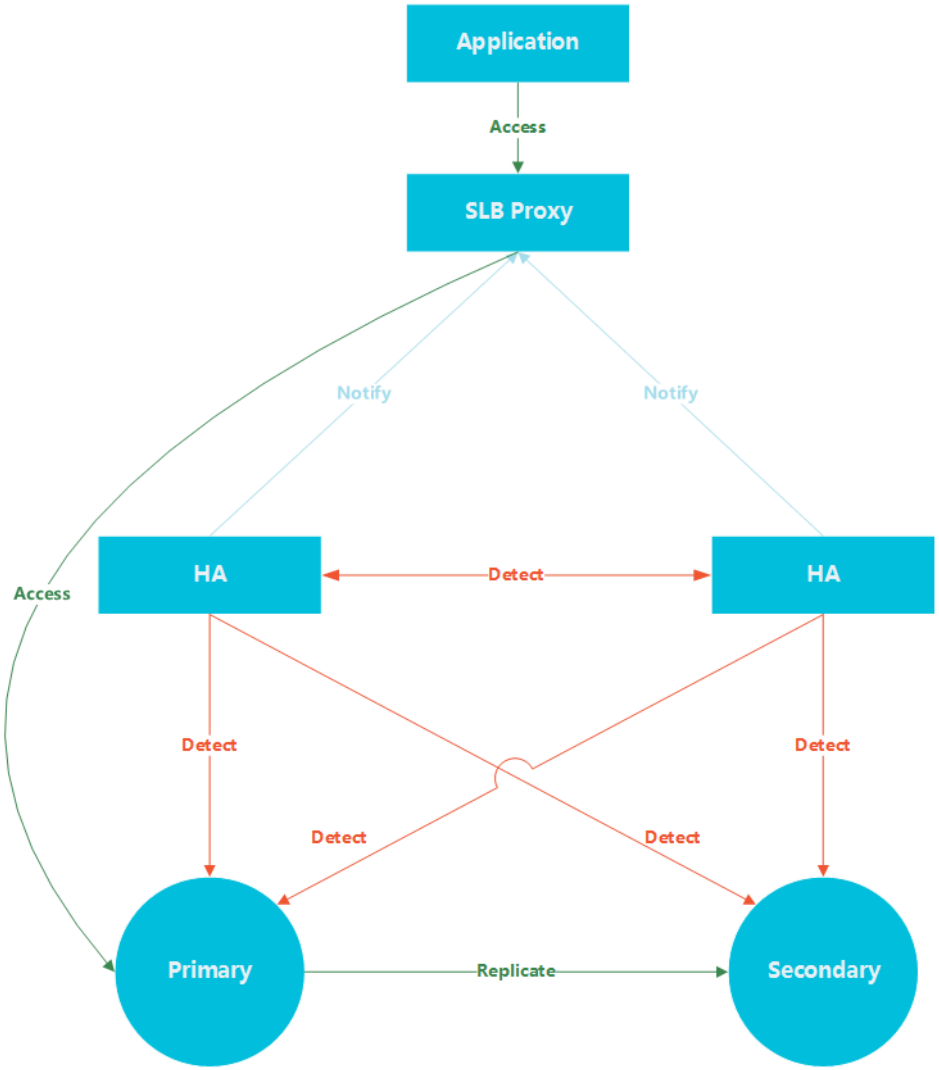
Engine	Version
Redis	Compatible with Redis 2.8 and Redis 3.0 GEO.
Redis	Redis 4.0

7.4.2 HA service

7.4.2.1 Overview

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

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



7.4.2.2 Detection

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

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

7.4.2.3 Repair

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

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

7.4.2.4 Notice

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

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

7.4.3 Monitoring service

7.4.3.1 Service-level monitoring

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

7.4.3.2 Network-level monitoring

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

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

7.4.3.3 OS-level monitoring

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

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

7.4.3.4 Instance-level monitoring

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

- **Instance availability.**
- **Instance capacity.**

7.4.4 Scheduling service

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

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

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

8 Data Transmission Service (DTS)

8.1 What is DTS?

Data Transmission Service (DTS) is a data service provided by Alibaba Cloud. It supports data exchanges between databases of various types such as relational databases and big data systems. DTS supports multiple data transmission methods such as data migration, real-time data subscription, and real-time data synchronization. These data transmission methods are used to achieve data migration with zero downtime, geo-disaster recovery, cache updates, online and offline real-time data synchronization, and asynchronous notifications.

DTS applies to multiple business scenarios. It enables you to build a secure, scalable, and highly available architecture.

8.2 Benefits

DTS supports data transmission between data sources such as relational databases and OLAP databases. It provides multiple data transmission methods such as data migration, change tracking, and data synchronization. Compared with other data migration and synchronization tools, DTS provides better transmission channels because it has high compatibility, high performance, security, and reliability. DTS also provides a variety of features to help you easily 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 supports multiple data transmission methods including data migration, change tracking, and data synchronization. In change tracking and data synchronization, data is transmitted in real time.

Data migration enables you to migrate data between databases with little downtime, which minimizes the impact of data migration on applications. 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 transmission optimization measures for data migration.

Compared with traditional data synchronization tools, the real-time synchroniz ation function of DTS refines the granularity of concurrency to the transaction level. This feature allows you to synchronize the incremental data in one table with multiple concurrent channels, improving synchronization performance.

Security and reliability

DTS is implemented based on clusters. If a node in a cluster is down or faulty, the control center quickly moves 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

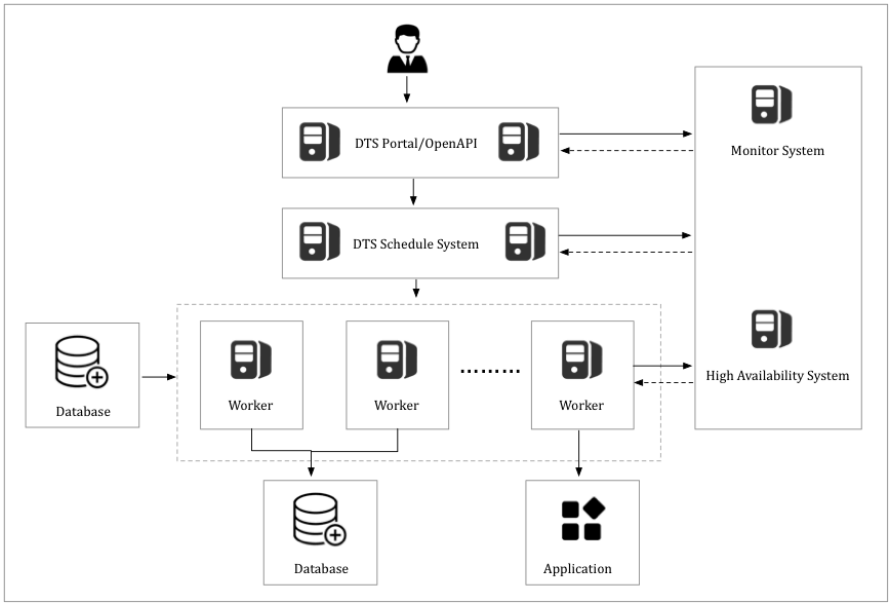
The DTS console provides a codeless wizard for creating and managing channels.

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

DTS supports resumable transmission, and regularly monitors channel status. If DTS detects an error such as network failure or system exception, it automatically fixes the error and restarts the channel. If the error persists, you need to manually repair the channel and restart it in the DTS console.

8.3 Architecture

The following figure shows the system architecture of DTS.



8.4 Environment requirements

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

- **PF51. ***
- **PV52P2M1. ***
- **DTS_E. ***
- **PF61. ***
- **PF61P1. ***
- **PV62P2M1. ***
- **PV52P1. ***
- **Q5F53M1. ***
- **PF52M2. ***
- **Q41. ***
- **Q5N1.22**
- **Q5N1.2B**
- **Q46.22**
- **Q46.2B**
- **W41.22**
- **W41.2B**
- **W1.22**
- **W1.2B**

- **W1.2C**
- **D13.12**

Use the following operating system:

AliOS7U2-x86-64

Notice:

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

  If the space of the hard disk where the `/apsara` directory resides is smaller than 2 TB, tasks may fail to run and errors may occur. In this case, DTS cannot restore failed tasks or pull data properly.

8.5 Features

8.5.1 Data migration

8.5.1.1 Data migration

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

8.5.1.2 Data sources

DTS supports migrating data between the following data sources.

Table 8-1: Data migration between different data sources

Data source	Schema migration	Full data migration	Incremental data migration
MySQL > RDS for MySQL	Supported	Supported	Supported

Data source	Schema migration	Full data migration	Incremental data migration
MySQL > Oracle	Not supported	Supported	Supported
Oracle > RDS for MySQL	Supported	Supported	Supported

8.5.1.3 Online migration

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

8.5.1.4 Migration modes

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

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

8.5.1.5 ETL features

Data migration supports the following ETL features:

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

8.5.1.6 Migration task

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

8.5.2 Data synchronization

8.5.2.1 Overview

Real-time data synchronization enables you to synchronize data between two data sources in real time. This feature applies to multiple scenarios, such as zone-disaster recovery, geo-disaster recovery, and data synchronization between OLTP and OLAP databases.

The following table describes synchronization features.

Table 8-2: Synchronization features

Source instance	Destination instance	One-way/two-way synchronization
MySQL	MySQL	· One-way synchronization · Two-way synchronization
MySQL	MaxCompute	One-way synchronization
MySQL	DataHub	One-way synchronization

8.5.2.2 Synchronization tasks

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

Table 8-3: Synchronization task statuses and descriptions **shows the statuses of a synchronization task during creation and running.**

Table 8-3: Synchronization task statuses and descriptions

Status	Description	Available operation
Pre-checking	**The synchronization task is performing a pre-check before the task is started.**	**· View synchronization configurations. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.**
Pre-check failed	**The synchronization task has failed the pre-check.**	**· Perform the pre-check. · View synchronization configurations. · Modify synchronization objects. · Modify synchronization speed. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.**
Not started	**The synchronization task that has passed the pre-check is not started.**	**· Perform the pre-check. · Start the synchronization task. · Modify synchronization objects. · Modify synchronization speed. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.**

Status	Description	Available operation
Initializing	The synchronization task is being initialized.	· View synchronization configurations. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.
Initialization failed	Data migration has failed during the synchronization task initialization.	· View synchronization configurations. · Modify synchronization objects. · Modify synchronization speed. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.
Synchronizing	The task is synchronizing data.	· View synchronization configurations. · Modify synchronization objects. · Modify synchronization speed. · Pause the synchronization task. · Delete the synchronization task. · Replicate synchronization configurations. · Configure monitors and alarms.

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

8.5.2.3 Synchronization objects

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

8.5.2.4 Advanced features

The following advanced features are used to facilitate data synchronization:

- **Dynamically add and remove synchronization objects**

  You can add and remove synchronization objects during data synchronization.
- **Improve the performance query system**

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

8.5.3 Data subscription

8.5.3.1 Real-time data subscription

Real-time data subscription can help you obtain the incremental data of RDS in real time. You can migrate incremental data based on your business requirements, such as cache updates, asynchronous business decoupling, real-time heterogeneous data synchronization, and real-time complex ETL data synchronization.

Real-time data subscription supports RDS for MySQL instances in classic networks and VPCs.

Real-time data subscription supports the following data sources:

- **RDS for MySQL**

8.5.3.2 Subscription channels and objects

Subscription channels

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

A subscription channel can only be subscribed and consumed by one downstream SDK. To subscribe to an RDS instance for multiple downstream SDKs, you must

create an equivalent number of subscription channels. RDS instances subscribed to with these subscription channels share the same instance ID.

Table 8-4: Subscription channel statuses and descriptions **shows the statuses of a subscription channel during creation and running.**

Table 8-4: Subscription channel statuses and descriptions

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

Subscription objects

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

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

8.5.3.3 Advanced features

The following advanced features are used to facilitate data subscription:

- **Dynamically add and remove subscription objects**

  You can add and remove subscription objects during data subscription.
- **View the subscribed data online**

  You can view the incremental data that has been subscribed to in the DTS console.
- **Modify data consumption time**

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

9 Data Management (DMS)

9.1 What is DMS?

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

9.1.1 Product value

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

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

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

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

DMS provides the following benefits:

- **Simple data operations**
  - **Pain point: You need a convenient and all-in-one product to complete SQL operations, save common operations, and apply common operations to specific services.**
  - **Solution: You can create a table in DMS and perform operations on table data just as you would in an Excel worksheet. You can add, delete, change, query, and make statistical analysis of table data without understanding SQL. You can customize SQL operations and save common business-related SQL operations in DMS. Then you can apply these operations directly when managing other databases or instances.**
- **Visualization of database table structures**
  - **Pain point: When you design a new business table or perform operations on an existing business table, you often need to understand the structures of all tables in a database. You can execute SQL commands one by one to display the table structures, but this method is neither intuitive nor convenient.**
  - **Solution: Through the document generation function of DMS, you can generate the table structures of an entire database with a single click. Then you can browse these structures online or export them to other formats such as Word, Excel, and PDF.**
- **Real-time optimization of database performance**
  - **Pain point: Detailed monitoring logs over a long period of time are required for database performance optimization. You need to make a detailed analysis of the logs and locate exceptions to better improve the database performance.**
  - **Solution: DMS provides second-level monitoring of database performanc e metrics, such as SELECT, INSERT, UPDATE, and DELETE operations, the number of active connections, and network traffic volume, and helps keep you informed of any performance variations. DMS allows you to view and terminate database sessions.**
- **Chart-based presentation of SQL result sets**
  - **Pain point: Users used to use SQL statements to find data, and import the data into Excel to create static charts such as line charts and pie charts. This process takes a lot of time.**
  - **Solution: With DMS, you can directly create charts from SQL result sets. You can also create many advanced charts, such as dynamic charts, period-over-**

period comparison charts, and personalized tooltips. This helps you produce high-quality work.

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

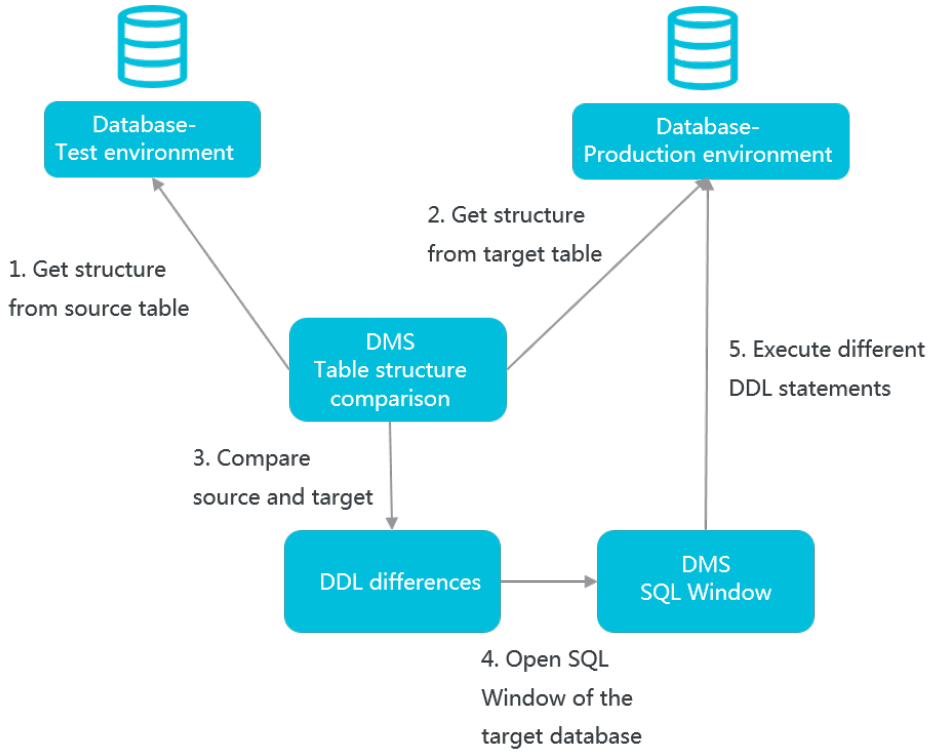
  Big data is the latest trend in data analysis and is widely discussed. However, taking full advantage of the values provided by big data analysis is not an easy task. The core idea of DMS is to start analyzing data when data is available.

  DMS monitors changes to table data volume through a custom RDS kernel, which allows it to quickly collect row count changes of each instance, database , and table. DMS provides real-time monitoring, data trends, and detailed data through professional data analysis and interaction.
- **Table structure synchronization**
  - **Pain point: Within enterprises, database environments are divided into production environment and test environment. A database will be released in the production environment after it is verified in the test environment. If table structures in the test environment are not completely synchronized to the production environment, major faults can occur during the release.**
  - **Solution: You can use the table structural comparison function of DMS to detect inconsistencies in database table structures between the production and test environments. You can also obtain a DDL statement for table structure**

correction to ensure table structure consistency between the production and test environments.

Figure 9-1: DMS - Table structure comparison - Table structure synchronization **shows how to use the table structure comparison function to synchronize table structures.**

Figure 9-1: DMS - Table structure comparison - Table structure synchronization



9.2 Technical advantages

Improved R&D efficiency

- **Schema comparison**
- **Smart SQL completion**
- **Convenient reuse of custom SQL statements and SQL templates**
- **Automatic recovery of working environments**
- **Export of dictionary files**

Real-time optimization of database performance

- **Effective session management**
- **Monitoring of core metrics in seconds**
- **Graphical lock management**

- **Real-time SQL index recommendations**
- **Reports on overall performance**

Comprehensive access security protection

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

Extensive options for data sources

- **Relational databases such as MySQL.**

9.3 Architecture

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

Business layer

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

Scheduling layer

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

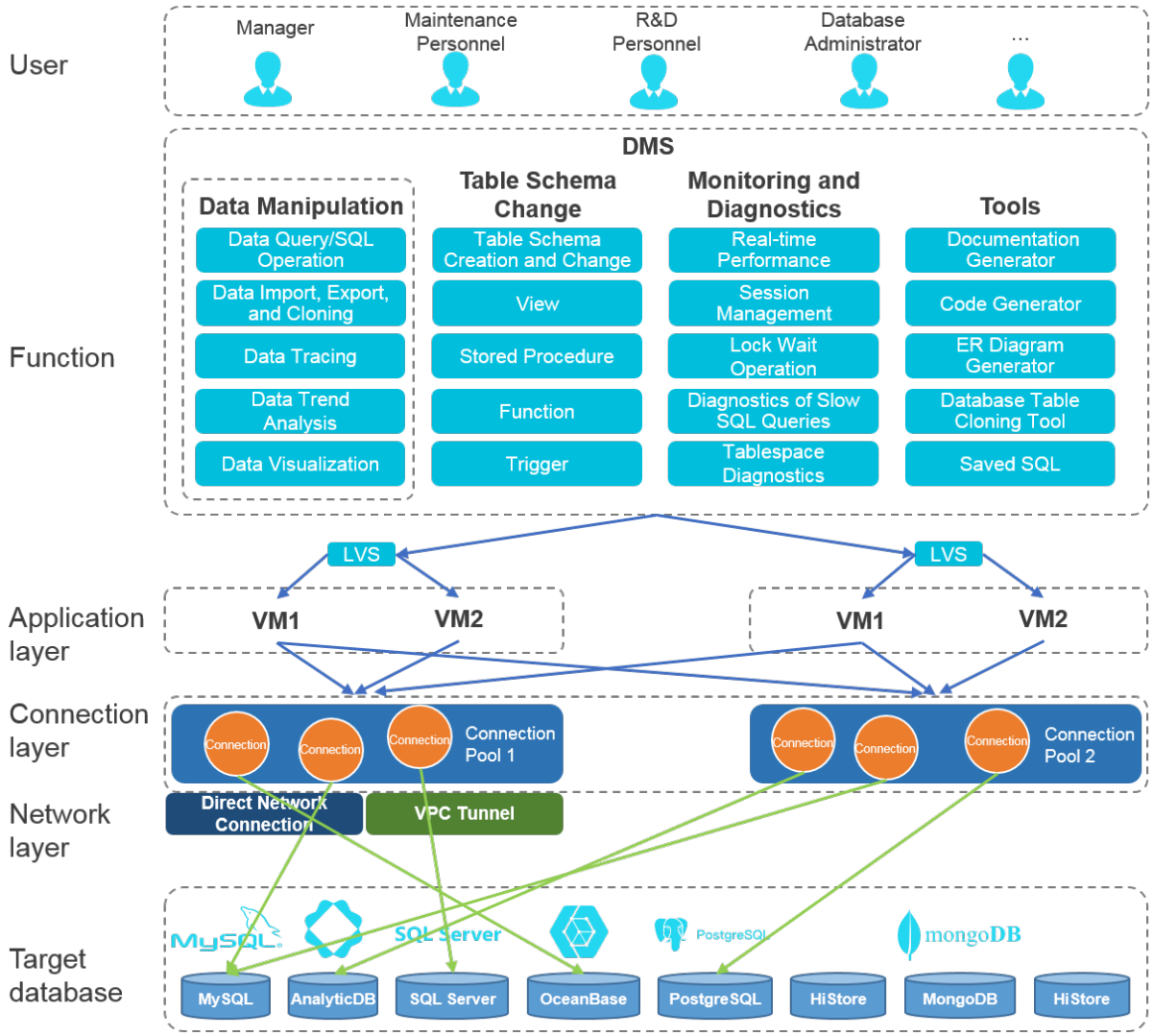
Connection layer

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

- **Processes requests from MySQL.**

- **Supports session isolation and persistence. SQL windows opened in DMS are isolated from each other and the sessions in each SQL window are persistent to simulate the client experience.**
- **Controls the number of instance sessions to avoid establishing a large number of connections to a single instance.**
- **Provides different connection release policies for different features. This improves user experience and reduces the number of connections to the databases.**

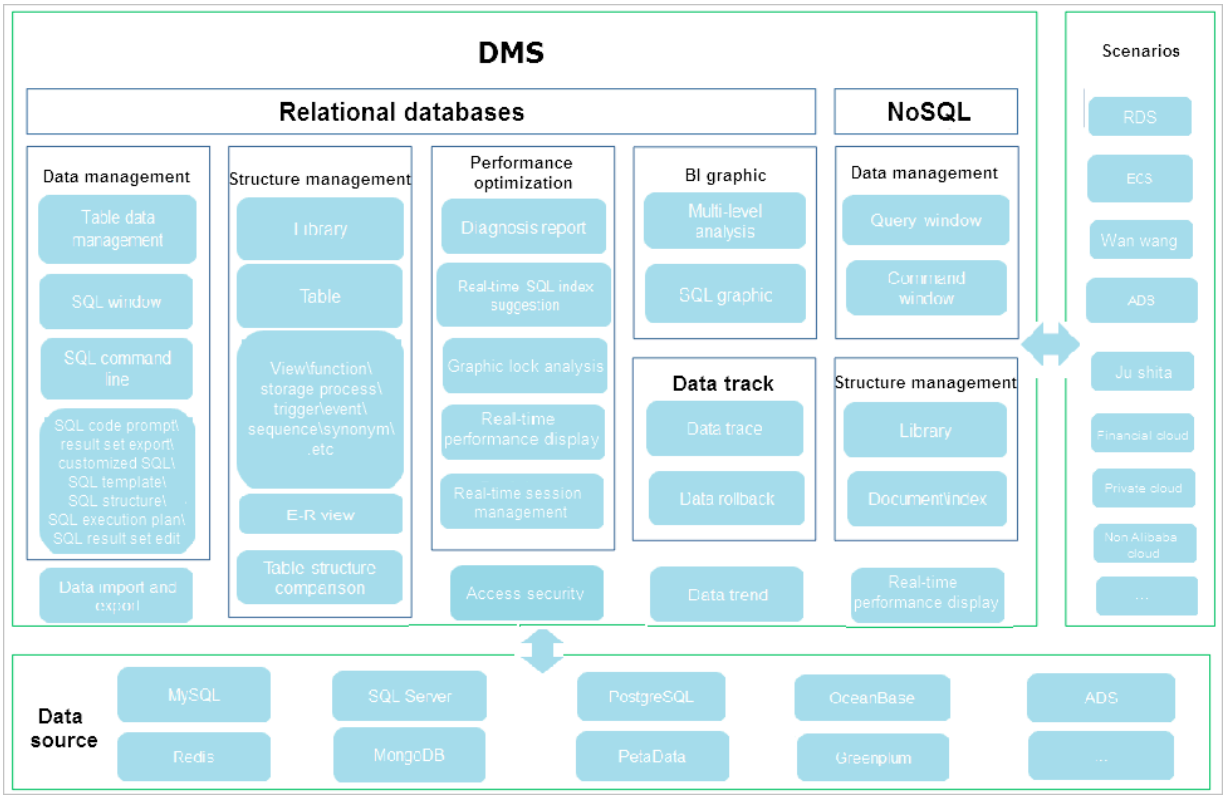
Figure 9-2: DMS system architecture



9.4 Features

The following figure shows the features of DMS.

Figure 9-3: DMS features



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

- **Features for NoSQL databases**
  - **Data management: query editor and command-line interface.**
  - **Schema management: management of objects such as databases, documents, and indexes.**
  - **Real-time performance monitoring: real-time display of key performance indicators.**

10 Server Load Balancer (SLB)

10.1 What is Server Load Balancer?

Server Load Balancer (SLB) is a type of traffic control service that distributes inbound traffic across multiple ECS instances based on forwarding rules. SLB improves the service capability and availability of applications.

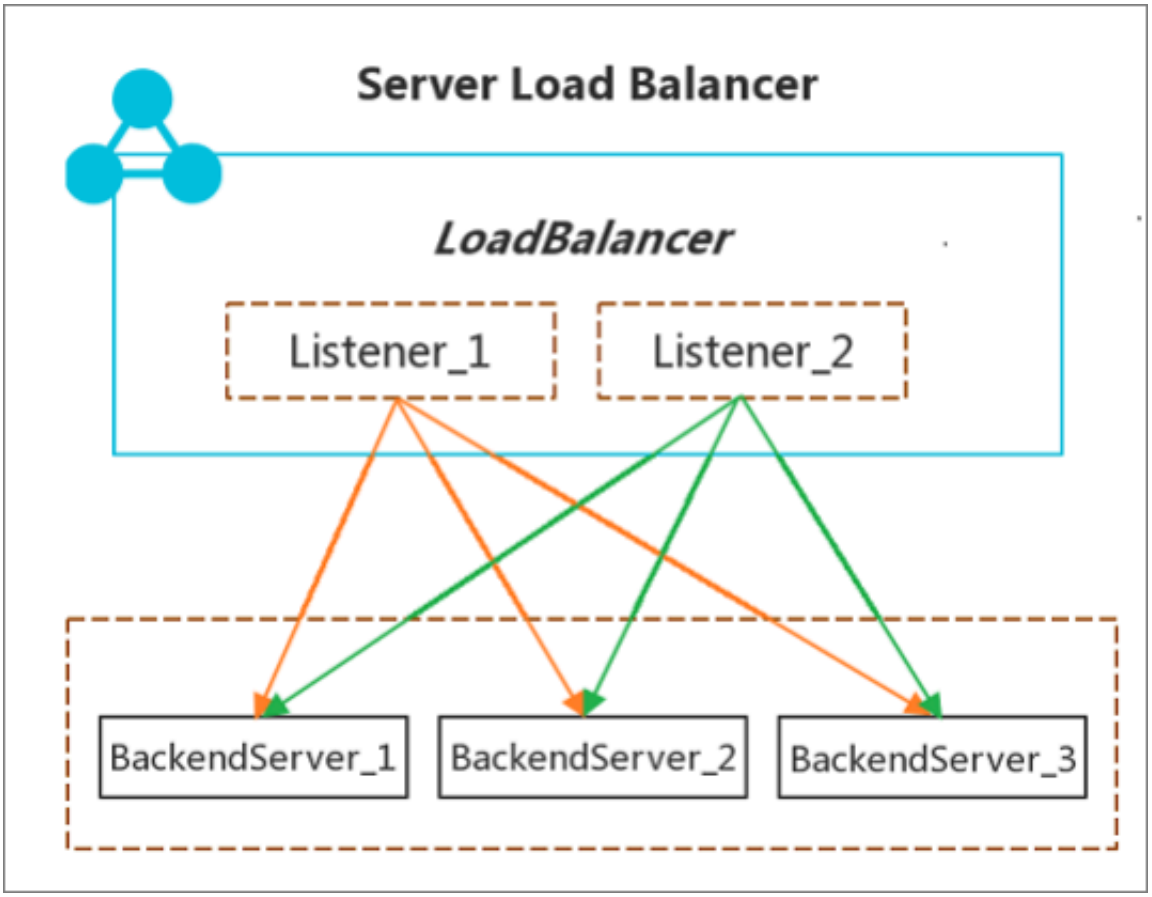
SLB consists of three components:

- **SLB instances: An SLB instance is a running load-balancing service that receives and distributes inbound traffic to back-end servers.**

  To use the SLB service, you must create an SLB instance with at least one listener and two ECS instances configured.
- **Listeners: A listener checks client requests and forwards them to back-end servers. It also performs health check on the back-end servers.**

  You can create Layer-4 (TPC/UDP) or Layer-7 (HTTP/HTTPS) listeners as needed. You can create domain- and URL- based forwarding rules for Layer-7 listeners.
- **Backend servers: Backend servers are ECS instances attached to an SLB instance to receive and process the distributed requests. You can divide ECS instances running different applications or functioning different roles into different server groups.**

  As shown in the following figure, after the SLB instance receives a client request , the listener forwards the request to the corresponding back-end ECS instances based on the configured forwarding rules.



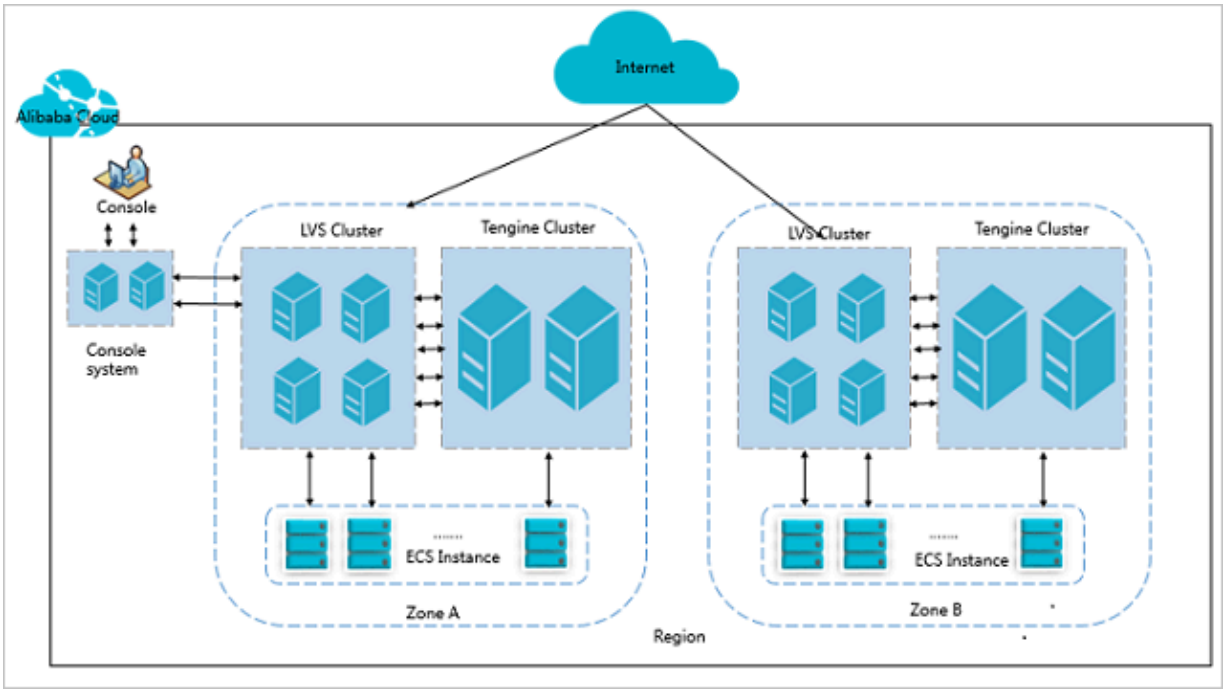
10.2 Architecture

SLB is deployed in clusters to achieve session synchronization. This can eliminate SPOFs of back-end servers, improve redundancy, and ensure service stability. Apsara Stack provides Layer-4 (TCP and UDP) and Layer-7 (HTTP and HTTPS) load-balancing services.

- **Layer-4 SLB combines the open-source Linux Virtual Server (LVS) with Keepalived to balance loads, and implements customized optimizations to meet cloud computing requirements.**

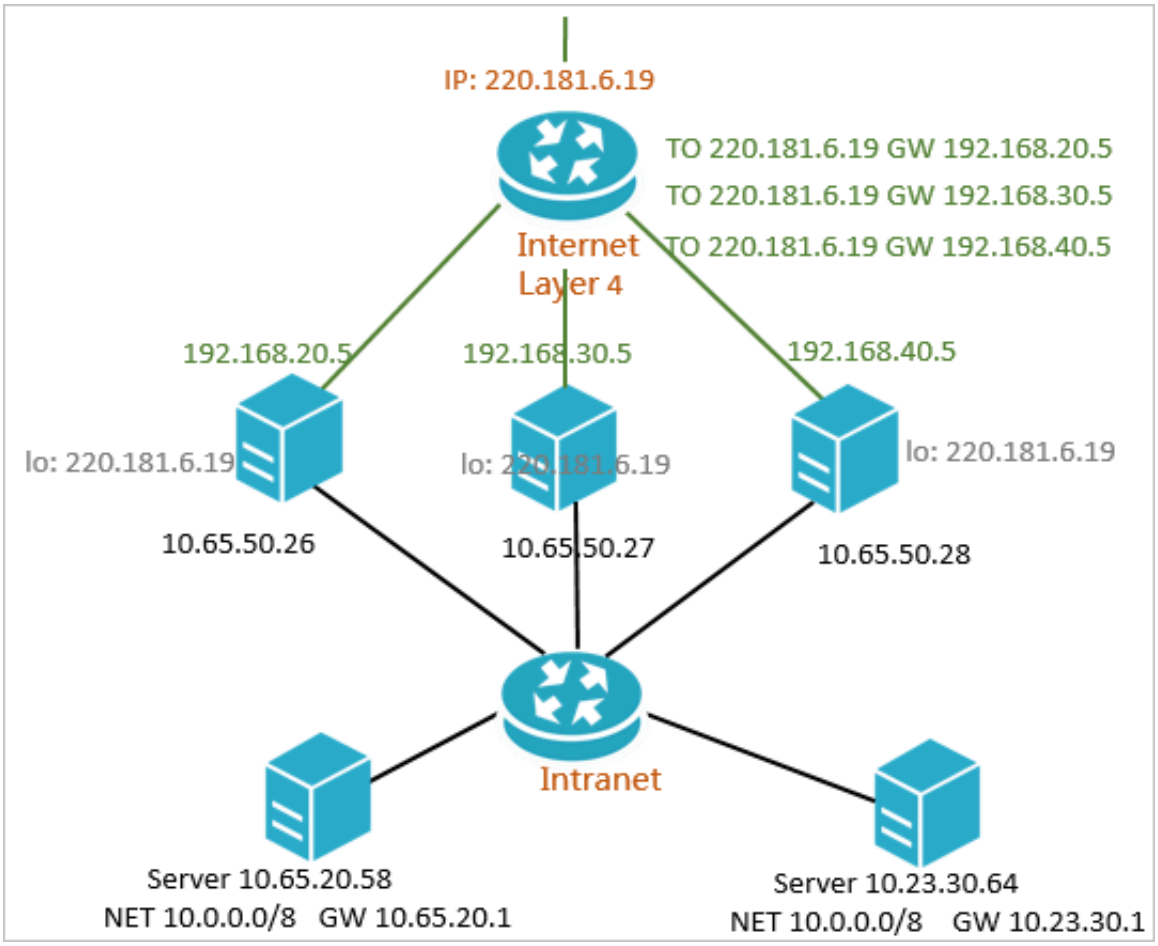
- **Layer-7 SLB uses Tengine to balance loads. Tengine is a Web server project launched by Taobao. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites.**

Figure 10-1: SLB architecture



As shown in the following figure, Layer-4 SLB runs in a cluster of LVS machines. The cluster deployment model strengthens the availability, stability, and scalability of load-balancing services in abnormal circumstances.

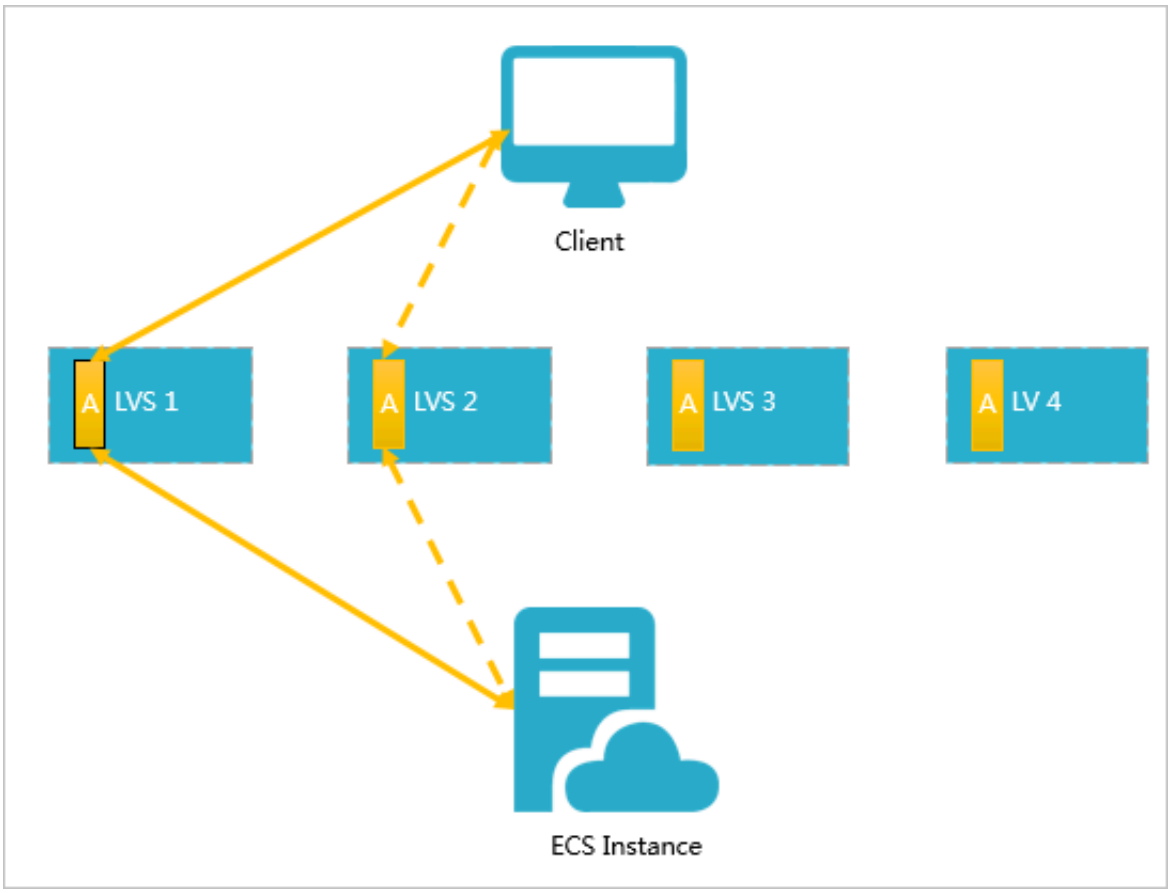
Figure 10-2: Cluster deployment



Each machine in the LVS cluster uses multicast packets to synchronize sessions with the other LVS machines. As shown in the following figure, Session A established on LVS1 is synchronized to other LVS machines after the client transfers three data packets to the server. Solid lines indicate the current active connections, while dotted lines indicate that the session requests will be sent to other normally working machines if LVS1 fails or is being maintained. In this way, you can perform

hot updates, machine failure maintenance, and cluster maintenance without affecting business applications.

Figure 10-3: Session synchronization



10.3 Features

Server Load Balancer (SLB) is a type of traffic control service that distributes inbound traffic across multiple ECS instances based on forwarding rules. SLB improves the service capability and availability of applications.

You can use SLB to virtualize multiple ECS instances in the same region into an application server pool with high performance and high availability by configurin g virtual IP addresses (VIPs). Then, you can distribute client requests to the ECS instances based on forwarding rules.

SLB checks the health status of the ECS instances and automatically isolates abnormal ones in the server pool to eliminate single points of failure (SPOFs), improving the overall service capability of applications. SLB is also well equipped to defend against DDoS attacks.

10.4 Benefits

10.4.1 LVS in Layer-4 SLB

This topic describes the customized technical improvements on the standard LVS performed by Alibaba Cloud.

Drawbacks of the standard LVS

Linux Virtual Server (LVS) is the world's most popular Layer-4 load-balancing software. LVS was developed by Dr. Zhang Wensong in May 1998 for Linux systems . LVS is a kernel module based on a linux netfilter framework named IP Virtual Server (IPVS), which is similar to iptables. LVS is hooked into LOCAL_IN and FORWARD.

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

- **Drawback 1: LVS supports three packet forwarding modes: NAT, DR, and TUNNEL. When these forwarding modes are deployed in a network with multiple VLANs, the network topology becomes complex and incurs high O&M costs.**
- **Drawback 2: Compared with commercial load-balancing devices such as F5, LVS lacks defense against DDoS attacks.**
- **Drawback 3: LVS uses PC servers and the Virtual Router Redundancy Protocol (VRRP) of Keepalived to deploy primary and secondary nodes for high availability . Therefore, its performance cannot be extended.**
- **Drawback 4: The configurations and health check performance of the Keepalived software are insufficient.**

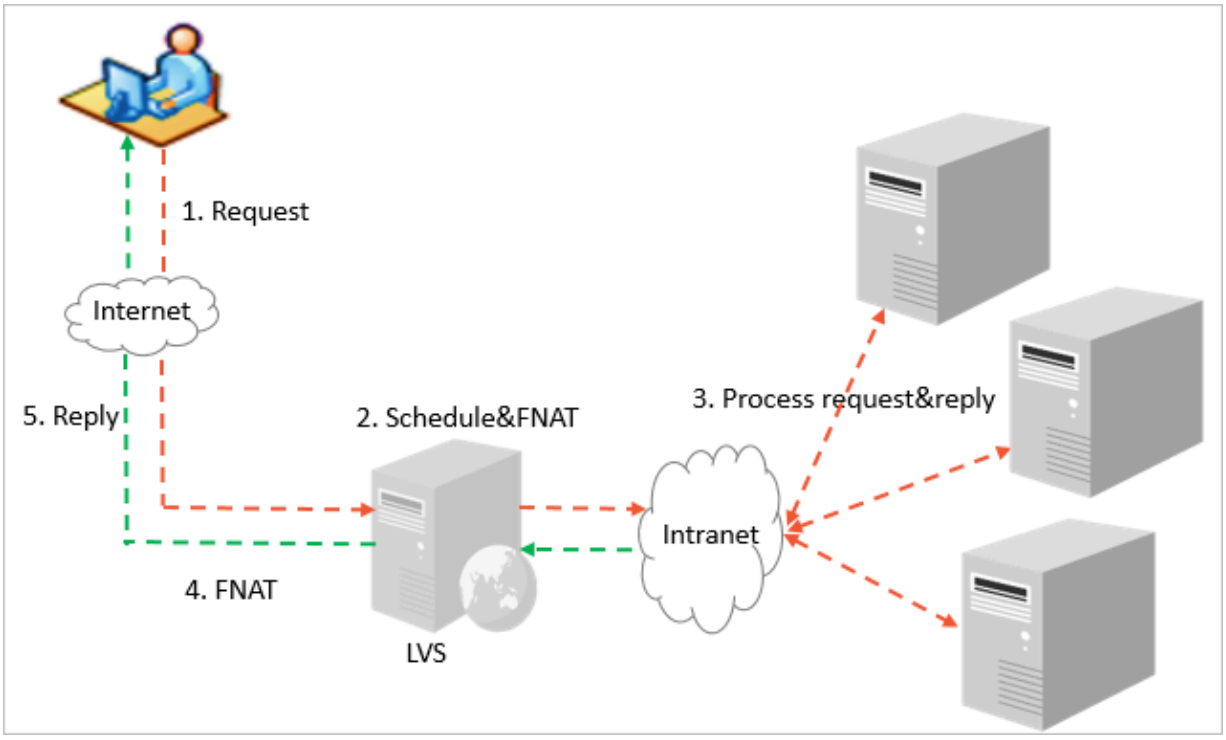
LVS customized features

To solve these problems, Alibaba Cloud added the following customized features to LVS. For more information about Ali-LVS, visit *https://github.com/alibaba/LVS*.

- **Customization 1: FULLNAT, a new forwarding mode that enables inter-VLAN communication between LVS load balancers and back-end servers.**
- **Customization 2: Defense modules such as SYNPROXY against TCP flag-targeted DDoS attacks.**
- **Customization 3: Support for LVS cluster deployment.**
- **Customization 4: Improved Keepalived performance.**

FULLNAT technology

- **Principles: The module introduces local addresses (internal IP addresses). IPVS translates cip-vip to and from lip-rip, in which both lip and rip are internal IP addresses. This means that the load balancers and back-end servers can communicate across VLANs.**
- **All inbound and outbound data flows traverse LVS. 10-GE Network Interface Cards (NICs) are used to ensure adequate bandwidth.**
- **Currently, FULLNAT supports only TCP.**



SYNPROXY technology

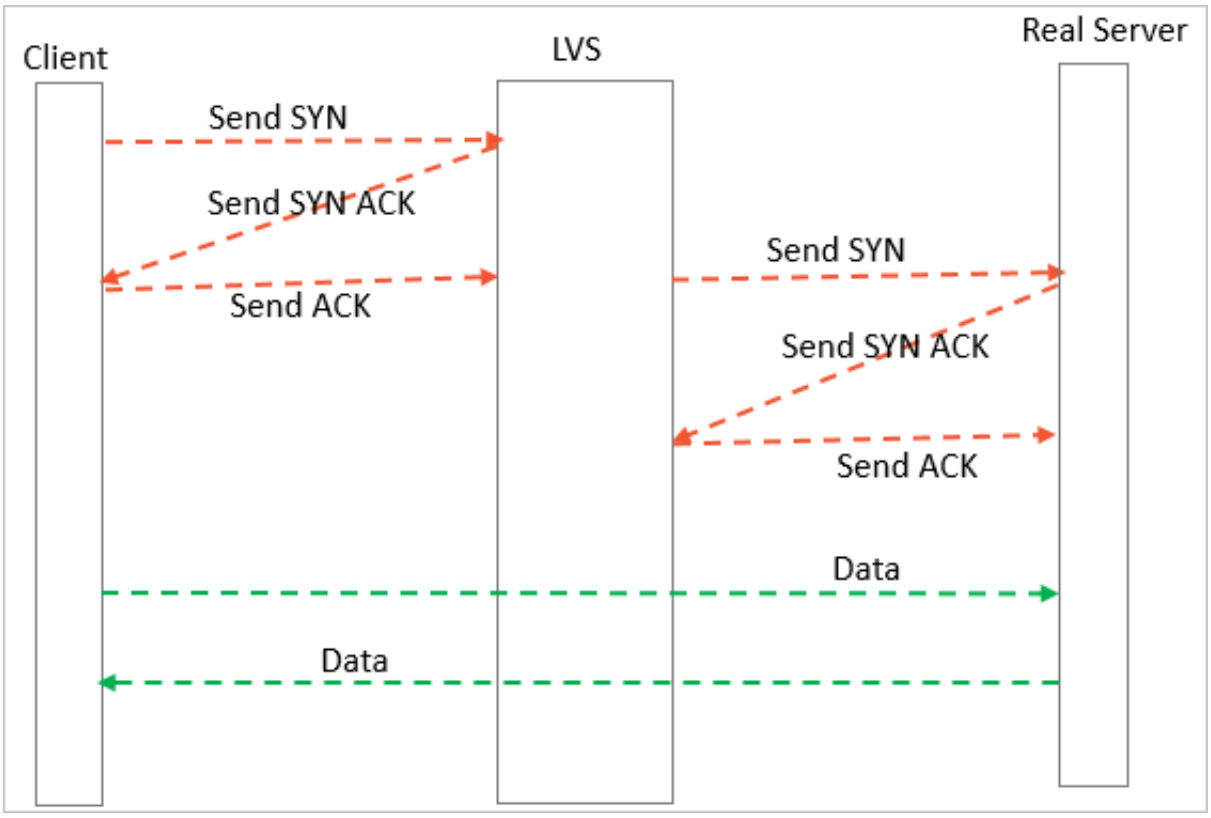
LVS uses the SYNPROXY module to defend against TCP flag-targeted attacks and Synflood attacks (both are DDoS attacks). According to the principle of SYN cookies in the Linux TCP protocol stack, LVS acts as a proxy for TCP three-way handshakes.

The process is as follows:

1. **A client sends an SYN packet to LVS.**
2. **LVS constructs an SYN-ACK packet with a unique sequence number and sends this packet to the client. The client returns an ACK response to LVS.**

3. **LVS checks whether the ack_seq value in the ACK response is valid. If the value is valid, LVS establishes a three-way handshake with a back-end server.**

Figure 10-4: LVS proxy of a three-way handshake



To defend against ACK, FIN, and RST flood attacks, LVS checks the connection table and discards any requests for connections which are undefined in the table.

Cluster deployment

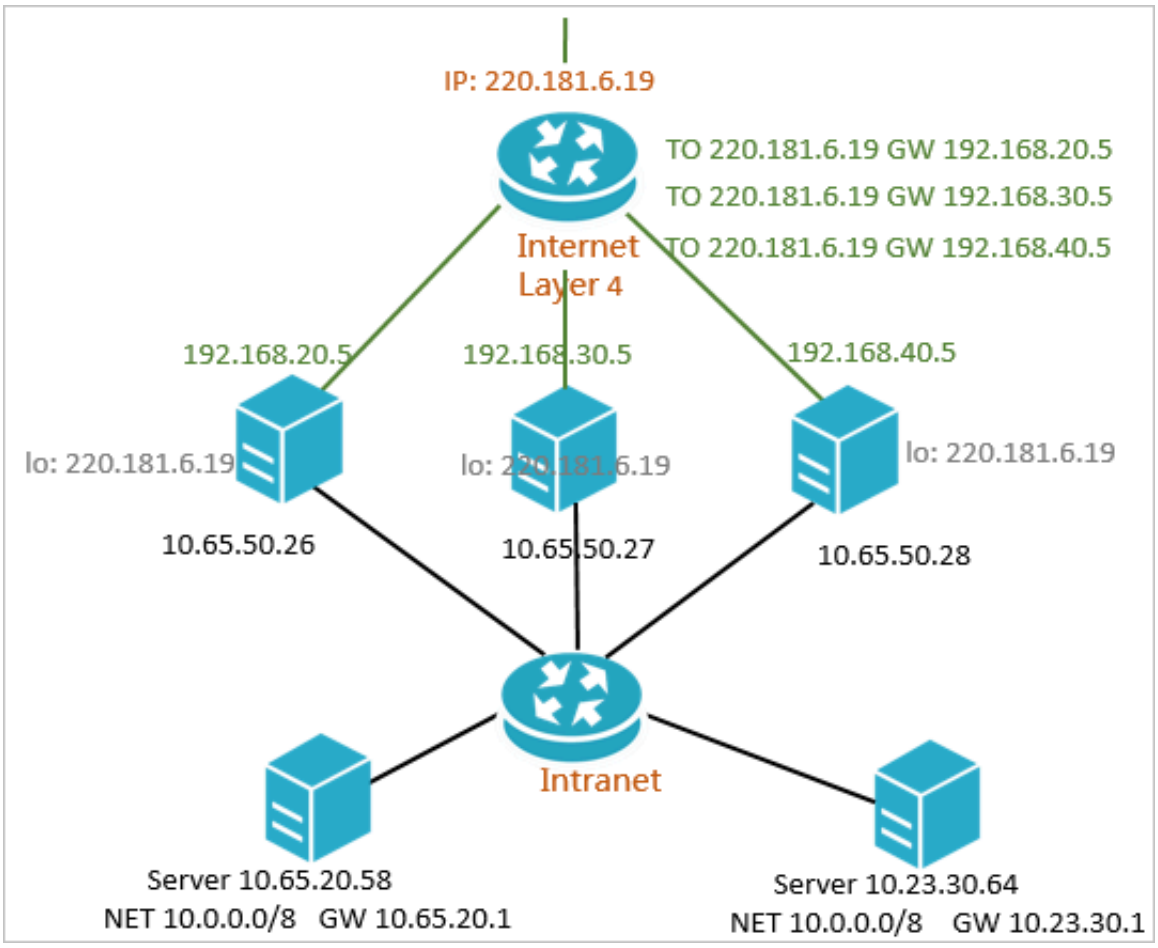
An LVS cluster communicates with uplink switches through OSPF. The uplink switches use equal-cost multi-path (ECMP) routes to distribute traffic to the LVS cluster. Then, the LVS cluster forwards the traffic to your servers.

The cluster deployment model ensures the stability of Layer-4 SLB with the following features:

- **Robustness: LVS and uplink switches use OSPF as the heartbeat protocol. A virtual IP address (VIP) is configured on all LVS nodes in the cluster. The switches can locate the failure of any LVS node and remove it from the ECMP route list.**

- **Scalability: You can scale out an LVS cluster if traffic from a VIP exceeds the cluster capacity.**

Figure 10-5: Cluster deployment



Keepalived optimization

Improvements made to Keepalived include:

- **Changing the asynchronous network model from select to epoll.**
- **Optimizing the reload process.**

Features of Layer-4 SLB

In conclusion, Layer-4 SLB has the following features:

- **High availability: The LVS cluster ensures redundancy and prevents SPOFs.**
- **Security: Together with Apsara Stack Security, LVS provides near real-time defense.**
- **Health check: LVS performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.**

10.4.2 Tengine in Layer-7 SLB

Tengine is a Web server project launched by Alibaba. Based on NGINX, Tengine has a wide range of advanced features enabled for high-traffic websites. NGINX is one of the most popular open-source Layer-7 load-balancing software.

For more information about Tengine, visit *http://tengine.taobao.org/***.**

Customized features

Tengine is customized for cloud computing scenarios:

- **Inherits all features of NGINX 1.4.6 and is fully compatible with NGINX configurations.**
- **Supports the dynamic shared object (DSO) module. This means you do not need to recompile Tengine to add a module.**
- **Provides enhanced load balancing capabilities, including a consistent hash module and a session persistence module. It can also actively perform health checks on back-end servers and automatically enable or disable servers based on their status.**
- **Monitors system loads and resource usage to protect the system.**
- **Provides error messages to help locate abnormal servers.**
- **Provides an enhanced protection module (by limiting the access speed).**

Features of Layer-7 SLB combined with Tengine

Layer-7 Server Load Balancer (SLB) is based on Tengine, and has the following features:

- **High availability: The Tengine cluster ensures redundancy and prevents single points of failure (SPOFs).**
- **Security: Tengine provides multi-dimensional protection against CC attacks.**
- **Health check: Tengine performs health check on back-end ECS instances and automatically isolates abnormal instances until they recover.**
- **Supports Layer-7 session persistence.**
- **Supports consistent hash scheduling.**

11 Virtual Private Cloud (VPC)

11.1 What is VPC?

A Virtual Private Cloud (VPC) is a private network established in Apsara Stack. VPCs are logically isolated from each other.

Background information

The continuous development of cloud computing technologies leads to increasing virtual network requirements such as scalability, security, reliability, privacy, and performance. This scenario has hastened the birth of a variety of network virtualization technologies.

Earlier solutions combined virtual and physical networks to form a flat network architecture, such as large layer-2 networks. As the scale of virtual networks grew, earlier solutions faced more serious problems. A few notable problems include ARP spoofing, broadcast storms, and host scanning. Various network isolation technologies emerged to resolve these problems by completely isolating the physical networks from the virtual networks. One of the technologies utilized VLAN to isolate users, but due to VLAN limitations, it could only support up to 4096 nodes. It is insufficient to support the huge amount of users in the cloud.

Benefits

A VPC has the following benefits:

- **Security**

  Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs.
- **Ease of use**

  You can create and manage a VPC in the VPC console. After a VPC is created, the system automatically creates a VRouter and a routing table for it.
- **Scalability**

  You can create multiple subnets in a VPC to deploy different services. Additionally, you can connect a VPC to a local IDC or another VPC to extend the network architecture.

Scenarios

VPC applies to scenarios with high requirements on communication security and service availability.

- **Host applications**

  You can host applications that provide external services in a VPC and control access to these applications from the public network by creating security group rules and access control whitelists. You can also isolate application servers from databases to implement access control. For example, you can deploy Web servers in a subnet that can access the public network, and deploy its application databases in a subnet that cannot access the public network.

- **Host applications that require public network access**

  You can host an application that requires access to the public network in a subnet of a VPC and route the traffic through NAT. After you configure SNAT rules, instances in the subnet can access the public network without exposing their private IP addresses, which can be changed to public IP addresses anytime to avoid external attacks.

- **Zone-disaster recovery**

  You can divide a VPC into one or multiple subnets by creating VSwitches. Different VSwitches within the same VPC can communicate with each other . Resources can be deployed to VSwitches of different zones to achieve zone-disaster recovery.

- **Isolate business systems**

  VPCs are logically isolated from each other. To isolate multiple business systems , such as the production and test environments, you can create a VPC for each environment. When the VPCs need to communicate with each other, you can create a peering connection between them.

- **Extend the local network architecture**

  To extend the local network architecture, you can connect the local IDC to a VPC. You can also seamlessly migrate local applications to the cloud without changing the application access method.

11.2 Benefits

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. Different VPCs are isolated by tunnel IDs:

- **Similar to traditional networks, VPCs can also be divided into subnets. ECS instances in the same subnet use the same VSwitch to communicate with each other, whereas ECS instances in different subnets use VRouters to communicate with each other.**
- **VPCs are completely isolated from each other and can only be interconnected by mapping an external IP address (EIP or NAT IP address).**
- **The IP packets of an ECS instance are encapsulated by using the tunneling technology. Therefore, information about the data link layer (the MAC address) of the ECS instance is not transferred to the physical network. This way, ECS instances in different VPCs are isolated at Layer 2.**
- **ECS instances in VPCs use security groups as firewalls to control the traffic to and from ECS instances. This way, ECS instances in different VPCs are isolated at Layer 3.**

11.3 Architecture

A VPC is a private network logically isolated from other virtual networks.

Network architecture

Each VPC consists of a private Classless Inter-Domain Routing (CIDR) block, a VRouter, and at least a VSwitch.

- **CIDR blocks**

  A CIDR block is a private IP address range in a VPC. The IP addresses of all cloud resources deployed in the VPC are within the specified CIDR block. When

creating a VPC or a VSwitch, you must specify the private IP address range in the form of a CIDR block.

You can use any of the following standard CIDR blocks and their subnets as the IP address range of the VPC.

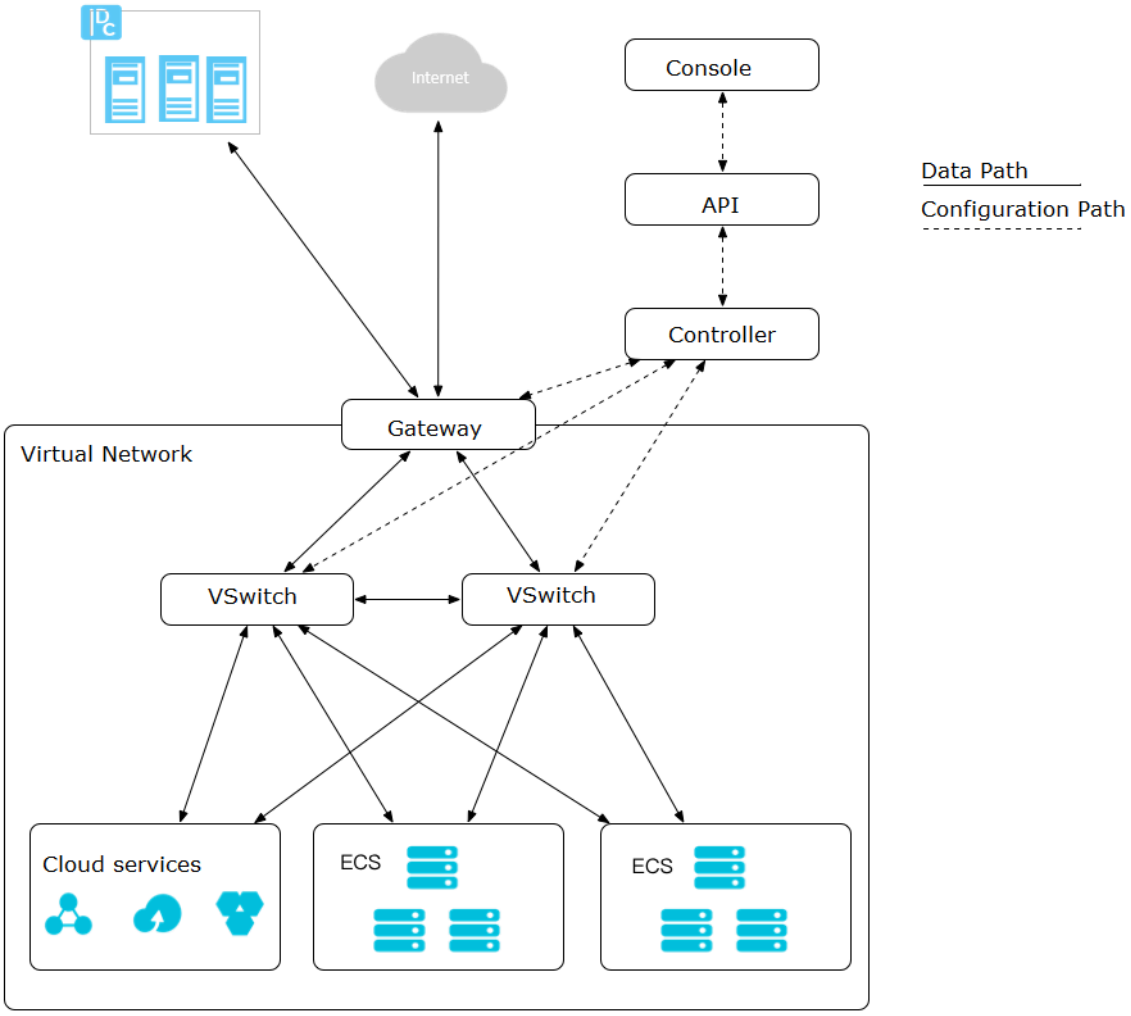
CIDR block	Number of available private IP addresses (system reserved ones excluded)
192.168.0.0/16	65,532
172.16.0.0/12	1,048,572
10.0.0.0/8	16,777,212

- VRouters

  A VRouter is the hub of a VPC. A VRouter is also an important component of a VPC. The VRouter connects the VSwitches in a VPC and serves as the gateway connecting the VPC with other networks. After you create a VPC, the system automatically creates a VRouter, which is associated with a routing table.

- Switches

  A VSwitch is a basic network device in a VPC and is used to connect different cloud product instances. After creating a VPC, you can further divide the VPC into one or more subnets by creating VSwitches. The VSwitches within a VPC are interconnected. You can deploy applications in VSwitches of different zones to improve the service availability.

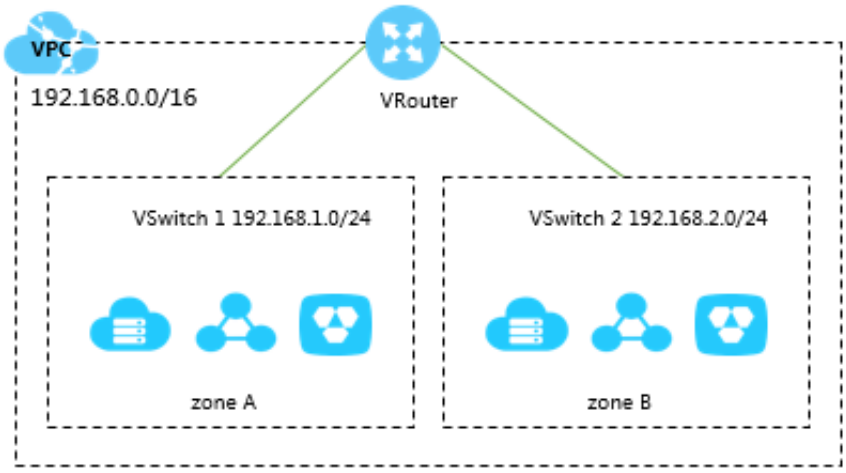


System architecture

The VPC architecture contains the VSwitches, gateway, and controller. The VSwitches and gateway form the key data path. Controllers use the protocol developed by Alibaba Cloud to forward the forwarding table to the gateway and VSwitches, completing the key configuration path. In the overall architecture, the configuration path and data path are separated from each other. VSwitches are distributed nodes. The gateway and controller are deployed in clusters. Multiple data centers are built for backup and disaster recovery. Redundant links

are provided for disaster recovery. This deployment mode improves the overall availability of the VPC.

Figure 11-1: VPC architecture



11.4 Features

A VPC is a logically isolated virtual network based on the mainstream tunneling technology.

Each VPC is identified by a unique tunnel ID. A unique tunnel ID is generated when tunnel encapsulation is performed on each data packet transmitted between the ECS instances within a VPC. Then, the data packet is transmitted over the physical network. ECS instances in different VPCs cannot communicate with each other. They have different tunnel IDs and therefore are on different routing planes.

Alibaba Cloud developed technologies such as the VSwitch, Software Defined Network (SDN), and hardware gateway based on the tunneling technology. These technologies serve as the basis for VPCs.

12 Apsara Stack Security

12.1 What is Apsara Stack Security?

Apsara Stack Security is a solution that provides Apsara Stack with a full suite of security features, such as network security, server security, application security, data security, and security management.

Background

Traditional security solutions for IT services detect attacks on network perimeters . They use hardware products such as firewalls and intrusion prevention systems (IPSs) to protect networks against attacks.

However, with the development of cloud computing, an increasing number of enterprises and organizations now use cloud computing services instead of traditional IT services. Cloud computing features low costs, on-demand flexible configuration, and high resource utilization. Cloud computing environments do not have definite network perimeters. As a result, traditional security solutions cannot effectively secure cloud assets.

With the powerful data analysis capabilities and professional security operations team of Alibaba Cloud, Apsara Stack Security provides integrated security protection services at the network layer, application layer, and server layer.

Complete security solution

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services, and provides users with a comprehensive security solution.

Security domain	Service	Description
Security management	Threat Detection Service	Monitors traffic and overall security status to audit and centrally manage security.
Server security	Server Guard	Protects Elastic Compute Service (ECS) instances against intrusions and malicious code.

Security domain	Service	Description
Application security	Web Application Firewall	Protects web applications against attacks and ensures that mobile and PC users can securely access web applications over the Internet.
Network security	Anti-DDoS	Ensures the availability of network links and improves business continuity.
Data security	Sensitive Data Discovery and Protection	Prevents data leaks and helps your business systems meet compliance requirements.
Security O&M service	On-premises security services	Help you establish and optimize your cloud security system to protect your business system against attacks by making full use of the security features of Apsara Stack Security and other Apsara Stack services.

12.2 Advantages

Since the enforcement of China Internet Security Law, Regulations on Critical Information Infrastructure Security Protection and Cloud Security Classified Protection Standard 2.0 have been published. As a result, private cloud platforms must pass the classified protection evaluation to ensure the security of cloud systems. Increasing security threats such as attacker intrusions and ransomware have led to the rising needs for security issue detection and prevention.

At the network perimeter of Apsara Stack, Apsara Stack Security uses a traffic security monitoring system to detect and block network-layer attacks in real time. It detects and removes Trojans and malicious files on servers to prevent attackers from exploiting the servers. In addition, Apsara Stack Security can block brute-force attacks and send alerts on unusual logons. This prevents attackers from stealing or destroying business data after logging on the system with weak passwords.

In-depth defense system

Apsara Stack Security comprises multiple functional modules. These modules work together to provide in-depth defense on the Apsara Stack network perimeter , within the Apsara Stack network, and on the Elastic Compute Service (ECS) instances in Apsara Stack. To help you manage security risks of Apsara Stack in

a centralized manner and in real time, Apsara Stack Security provides a unified security management system. This system allows you to manage the security policies in all security protection modules and perform association analysis on the logs.

The security protection modules provided by Apsara Stack Security cover network security, server security, application security, and threat analysis. Based on a management center that can integrate the security information from all modules, Apsara Stack Security can accurately detect and block attacks. In this way, Apsara Stack Security protects your business systems in the cloud against intrusions.

Security solutions completely integrated with the cloud platform

Apsara Stack Security is a product born from ten years of protection experience. After a decade of experience in providing security operations services for the internal businesses of Alibaba Group and six years of safeguarding the Alibaba Cloud security operations, Alibaba has obtained considerable security research achievements, security data, and security operations methods, and has built a professional cloud security team. Apsara Stack Security brings together the rich experience of these experts to develop the sophisticated systems that provide enhanced security for cloud computing platforms. This product can protect the cloud platform, cloud network environments, and cloud business systems of Apsara Stack users.

The components of Apsara Stack Security are software-defined, with a full hardware compatibility. With these components, you can implement elastic cloud computing services based on quick deployment, expansion, and implementation. The protection modules on the cloud network perimeter or in the cloud network adopt the bypass architecture, which completely fits the cloud businesses and has the minimal adverse impacts on the cloud businesses. The protection modules running on the ECS instances are all virtualized to fit the flexibility of the ECS instances.

User security situation awareness

The cloud platform provides services for users. In Apsara Stack Security console, a user can view the security protection data, generate security reports, and enable SMS and email alerts by configuring external resources.

Security capability output

Apsara Stack Security has accumulated a large number of protection policies over the last several years. The service has protected millions of users from hundreds of thousands of attacks every day. This has generated a large amount of security protection data. Apsara Stack Security analyzes over 10 TB of this data every day . The analysis results are used to enhance the fundamental security capabilities , such as the malicious IP library, malicious activity library, malicious sample library, and vulnerability library. These capabilities are applied in the protection modules of Apsara Stack Security to enhance your business security.

12.3 Architecture

Apsara Stack Security consists of Apsara Stack Security Standard Edition and optional security services.

Apsara Stack Security Standard Edition

- **Threat Detection Service**

  This module collects network traffic and server information, and detects possible vulnerability exploits, intrusions, and virus attacks through machine learning and data modeling. It also provides you with up-to-date information about ongoing attacks to help you monitor the security status of your businesses.
- **Network Traffic Monitoring System**

  This module is deployed on the network perimeter of Apsara Stack. It allows you to inspect and analyze each inbound or outbound packet of an Apsara Stack network through traffic mirroring. The analysis results are used by other Apsara Stack Security modules.
- **Server Guard**

  This module safeguards ECS instances by providing security features such as vulnerability management, baseline check, intrusion detection, and asset management. To do this, the module performs operations such as log monitoring , file analysis, and signature scanning.
- **Web Application Firewall**

  This module protects web applications against common web attacks reported by Open Web Application Security Project (OWASP), such as Structured Query

Language (SQL) injections, cross-site scripting (XSS), exploitation of vulnerabilities in web server plug-ins, Trojan uploads, and unauthorized access. It also blocks a large number of malicious visits to avoid data leaks and ensure both the security and availability of your websites.

Apsara Stack Security Standard Edition also provides on-premises security services. These services help you better use the features of Apsara Stack products such as Apsara Stack Security to secure your applications.

On-premises security services include pre-release security assessment, access control policy management, Apsara Stack Security configuration, periodic security check, routine security inspection, and urgent event handling. These services cover the entire lifecycle of your businesses in Apsara Stack and help you create a security operations system. This system enhances the security of your application systems and ensures both the security and stability of your businesses.

Optional security services

You can also choose the following service modules to enhance your system security.

- DDoS Traffic Scrubbing

  This module detects and blocks distributed denial of service (DDoS) attacks.
- Sensitive Data Discovery and Protection

  This module uses Alibaba Cloud's big data analytics capabilities and artificial intelligence (AI) technologies to detect and classify sensitive data based on your business requirements. It masks sensitive data both in transit and at rest, monitors dataflows, and detects abnormal activities. This module provides visible, controllable, and industry-compliant security protection for your sensitive data by means of precise detection and analysis.

12.4 Features

12.4.1 Apsara Stack Security Standard Edition

12.4.1.1 Threat Detection Service

Threat Detection Service (TDS) is a big data security analysis system developed by the Alibaba Cloud security team.

This module analyzes traffic data and server information to detect possible intrusions or attacks through machine learning and data modeling. It detects

vulnerability exploits and new virus attacks launched by advanced attackers, and shows you the information about ongoing attacks, enabling you to monitor the security of your business systems.

Features

The following table describes the features that TDS provides.

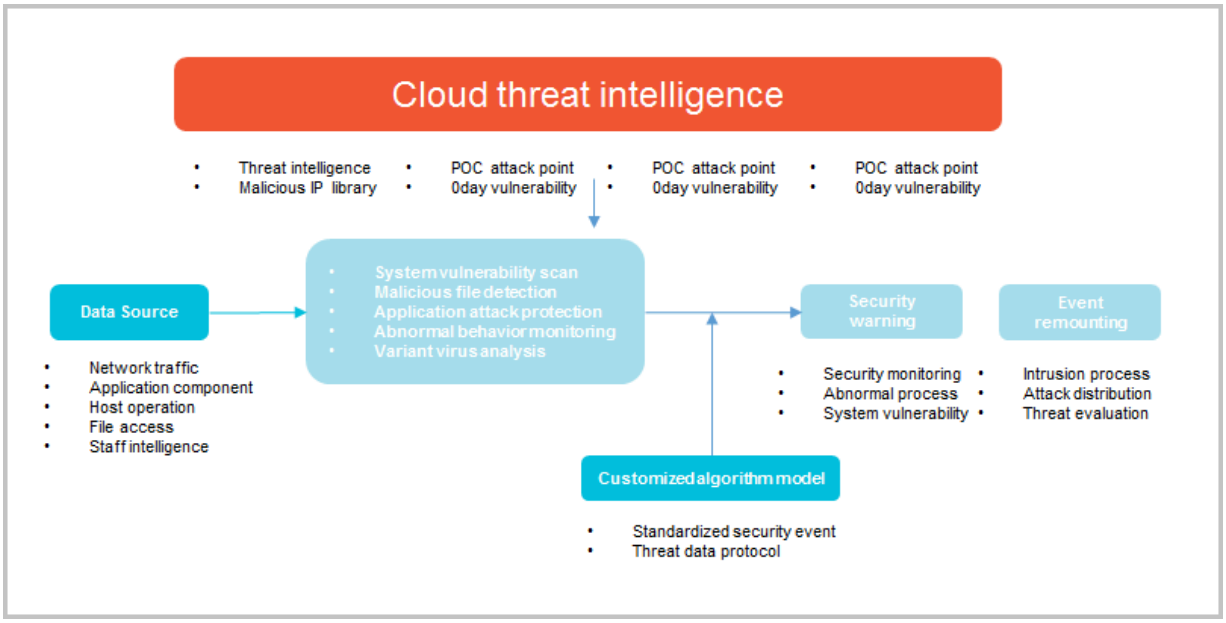
Feature		Description
Security situation overview	Security situation overview	Provides overall security information, including the number of emergencies, attacks on the current day, flaws on the current day, attack trends, latest threat analysis, latest intelligence, and protected assets information.
	Screens	Provides the following screens for displaying security information: · Map-based traffic data screen. · Server security screen.
Event analysis	Security event analysis	Uses big data algorithms and models to detect the following security events in the cloud: · Zombie activities: A server becomes a zombie when it is controlled by an attacker and launches distributed denial of service (DDoS) attacks on other servers. · Brute-force attacks: An attacker logs on to a server through brute-force attacks. · Backdoors: The WannaCry ransomware, unusual MySQL scripts, or webshells are detected. · DDoS attacks: A server encounters DDoS attacks. · Hacking tools: The logon credentials of a server are stolen, and hacking tools and attacks are detected. · Suspicious network connections: An attacker uses PowerShell to download suspicious files, runs suspicious VBScript commands, downloads malicious files or scripts to Linux servers, or runs commands in reverse shells. · Unusual traffic: A mining process is running.

Feature		Description
Traffic Security Monitoring	Traffic data collection and analysis	· Uses a bypass in traffic mirroring mode to collect inbound and outbound traffic that passes through the interconnection switch (ISW) and generates a traffic diagram. · Collects the traffic data of a Classless Inter-Domain Routing (CIDR) block or IP address, including traffic on the current day, traffic in the last 30 days, traffic in the last 90 days, and the queries per second (QPS).
	Malicious server identification	Detects attacks launched by internal servers to identify controlled internal servers.
	Web applicatio n protection	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks , and exploitation of upload vulnerabilities and common content management system (CMS) vulnerabilities.
	Unusual traffic detection	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.

How it works

The following figure shows how TDS works.

Figure 12-1: How TDS works



- **Big data security analysis platform**
  - **Network: TDS uses the HTTP requests and responses collected by the Traffic Security Monitoring module to create HTTP logs, and uses big data models to analyze the logs and detect security events and threats.**
  - **Server: TDS uses the rules engine to analyze the server process data collected by Server Guard and detect security events and threats.**
- **Security event display**
  - **Security events reported by Server Guard**
  - **Server security events discovered by server process analysis based on the rules engine**
  - **Network security events discovered by HTTP log analysis based on big data models**

Benefits

TDS has the following benefits:

- **Threat analysis based on big data**

  TDS provides analysis and computing of petabyte-level big data and collects all security data and threat intelligence of the entire network. It also uses machine learning technologies to create complete and smart security threat models that can be used in the application scenarios of millions of users.

  TDS focuses on the security trends and new threats that are faced by users of cloud computing in data centers, such as targeted Web application attacks, brute-force attacks, and system intrusions. It defends user systems against these threats from different fields.

- **Screens**

  Based on Internet visualization technologies, TDS displays the results of big data threat analysis in graphs on screens to support security decision making on Apsara Stack.

12.4.1.2 Traffic Security Monitoring

The Traffic Security Monitoring module is an Apsara Stack Security service that can detect attacks within milliseconds.

By performing in-depth analysis on the traffic packets mirrored from the Apsara Stack network ingress, this module can detect various attacks and unusual activities in real time and coordinate with other protection modules to implement defenses. The Traffic Security Monitoring module provides a wealth of information and basic data support for the entire Apsara Stack Security defense system.

Features

The following table describes the features that the Traffic Security Monitoring module provides.

Feature	Description
Traffic data collection and analysis	Uses a bypass in traffic mirroring mode to collect inbound and outbound traffic that passes through the interconnection switch (ISW) and generates a traffic diagram.

Feature	Description
Unusual traffic detection	Uses a bypass in traffic mirroring mode to detect the unusual traffic that has exceeded the scrubbing threshold and reroutes the traffic to the DDoS Traffic Scrubbing module. The traffic rate (Unit: Mbit/s), packet rate (Unit: PPS), HTTP request rate (Unit: QPS), or number of new connections can be set as the threshold.
Malicious server identification	Detects attacks launched by internal servers to identify controlled malicious servers.
Web application protection	Uses a bypass to block common attacks on Web applications at the network layer based on default Web attack detection rules. The attacks that can be blocked include Structured Query Language (SQL) injections, code and command execution, Trojan scripts, file inclusion attacks, and exploitation of upload vulnerabilities and common content management system (CMS) vulnerabilities.
Suspicious TCP connection blocking	Uses a bypass to send TCP RST packets to the server and the client to block layer-4 TCP connections.
Network log recording	Records UDP and TCP traffic logs and the Request and Response logs of HTTP queries. Threat Detection Service (TDS) uses these logs for big data analysis.

How it works

The Traffic Security Monitoring module collects data, processes the data, and then generates data processing results. It uses sockets to exchange data.

- **Collection: The module collects traffic data through multiple high-performance PCs with dual-port 10GE network interface controllers (NICs).**
- **Processing: Traffic from an IP address may pass through multiple collectors. Traffic data must be consolidated to generate usable information.**
- **Output: The module stores and provides the consolidated traffic data.**

12.4.1.3 Server Guard

Server Guard provides security protection measures such as vulnerability management, baseline check, intrusion detection, and asset management for

Elastic Compute Service (ECS) instances by means of log monitoring, file analysis, and feature scanning.

Server Guard uses the client-server model. To protect the security of ECS instances in real time, Server Guard clients work with the Server Guard server to monitor attacks, vulnerabilities, and baseline configurations at the system layer and the application layer on the ECS instances.

Features

The following table describes the features provided by Server Guard.

Category	Feature	Description
Vulnerability management	Linux software vulnerability detection and fixes	Detects vulnerabilities recorded in the official database of Common Vulnerabilities and Exposures (CVE) in software such as SSH, OpenSSL, and MySQL based on the software versions, and provides vulnerability information and solutions.
	Windows vulnerability detection and fixes	Detects critical Window vulnerabilities on your ECS instances based on the latest vulnerability information released by Microsoft, and provides Windows patches to fix the vulnerabilities, such as the Server Message Block (SMB) remote code execution vulnerability.  Note: By default, only critical vulnerabilities are reported. You can manually check for security updates and detect low-risk vulnerabilities.

Category	Feature	Description
	Web CMS vulnerability detection and fixes	Detects Web content management system (CMS) vulnerabilities based on the security intelligence provided by Alibaba Cloud by scanning directories and files. This feature also provides patches developed by Apsara Stack Security to fix vulnerabilities in software such as WordPress and Discuz!, and allows you to undo vulnerability fixes.
	Configuration and component vulnerability detection	Detects vulnerabilities that cannot be detected by software version comparison or file vulnerability scanning, and identifies critical configuration vulnerabilities in software, such as configuration and component vulnerabilities including unauthorized access to Redis and ImageMagick vulnerabilities.
Baseline check	Account security baseline check	· Detects SSH, RDP, FTP, MySQL, PostgreSQL, and SQL Server accounts with weak passwords. · Detects the at-risk accounts of your ECS instances, such as suspicious hidden accounts and cloned accounts. · Checks the compliance of the password policy of Linux servers. · Detects accounts without passwords on the ECS instances.

Category	Feature	Description
	System configuration check	Checks the system group policies, baseline logon policies, and registry configuration risks, including: · Suspicious auto-startup items in the scheduled tasks of Linux servers. · Auto-startup items on Windows servers. · Sharing configurations of the system. · SSH logon security policies of Linux servers. · Account-related security policies on Windows servers.
	Database security baseline check	Checks whether the Redis service on an ECS instance is exposed to the public network, whether unauthorized access vulnerabilities exist, and whether suspicious data is written to important system files.
	Benchmark compliance check	Checks whether the system baseline complies with the latest Center for Internet Security (CIS) CentOS Linux 7 Benchmark.
Intrusion detection - unusual logons	Disapproved logon location alerts	Automatically records all logons, and determines the approved logon cities based on the usual logon locations. It also generates alerts on logons in disapproved locations. You can customize the approved logon cities.
	Disapproved logon IP alerts	Generates alerts on logons through IP addresses that are not whitelisted after a logon IP whitelist is created.
	Disapproved logon time alerts	Generates alerts on logons within disapproved time ranges after the approved logon time range is set.
	Disapproved logon account alerts	Generates alerts on logons with disapproved accounts after the approved logon account list is created.

Category	Feature	Description
	Brute-force attack detection and blocking	Detects and blocks brute-force attacks in real time. Both SSH and RDP brute-force attacks can be detected.
Intrusion detection - webshells	Webshell detection	Uses an Alibaba-developed webshell detection engine to detect and remove webshells on your ECS instances. Both scheduled and real-time scans are supported. It detects webshells written in Active Server Pages (ASP), Hypertext Preprocessor (PHP), and Java Server Pages (JSP), and allows you to manually quarantine these webshell files.
Intrusion detection - suspicious processes	Suspicious process activity detection	Detects suspicious process activities such as reverse shells, Java processes running CMD commands, and unusual file downloads using bash.
Asset management	Asset grouping	Allows you to group your ECS instances into a maximum of four layers, filter assets by region, online status, or other features, and manage the asset tags.
	Asset fingerprints	· Listening port: collects and displays the listening port information and records changes. This allows you to easily check the listening port status. · Account: collects the account and permission information to discover privileged accounts and detect privilege escalations. · Process: collects and displays the information about the processes through snapshots to list normal processes and detect suspicious processes. · Software: lists the software installation information so that the affected assets can be quickly located when a critical vulnerability is exploited.

Category	Feature	Description
Server logs	Log retrieval	· Previous logon logs: records successful system logons. · Brute-force attack logs: records failed system logons. · Process snapshot logs: records the information about the running processes on the server at a specific time. · Listening port snapshot logs: records the listening port information on the server at a specific time. · Account snapshot logs: records the information about logon accounts on the server at a specific time. · Process startup logs: records the process startup information on the server. · Network connection logs: records the network connections started by the server.

How it works

Server Guard uses the client-server model. The client is installed on ECS instances. The client communicates with the server through a TCP persistent connection and uses HTTP to obtain scripts, rules, and installer packages from the server.

The client can be used in Windows or Linux. It can automatically connect to the server for online updates.

The key features of Server Guard work as follows:

- Vulnerability management: The client collects the ECS instance information, including component information, software versions, file information, and registry information. Then, the client checks whether the information matches the vulnerability detection rules provided by the server. The information that matches the rules will be sent to the server for further analysis. The detected vulnerabilities will be displayed in the Server Guard console. You can fix vulnerabilities in the console or by calling API operations. After receiving the vulnerability patches from the server, the client on the vulnerable ECS instance

automatically fixes the vulnerabilities and synchronizes the vulnerability status to the server.

- Baseline check: When you manually start a check or a periodic check is triggered, the Server Guard server sends a baseline check request to the client. The client then collects the server information according to the check policy and compares the information with the security baseline. Check items that do not comply with the baseline are labeled as at-risk items and reported to the server.
- Unusual logon detection: The client monitors the logon logs of the server system in real time. In a Linux system, the `/var/log/secure` and `/var/log/auth.log` files are also monitored. All failed and successful logons are recorded. Unusual logons or brute-force attacks will be reported to the server.
- Webshell detection: The client uses an Alibaba-developed dynamic webshell detection engine to detect complex webshells. It then restores these webshells to an identifiable status to analyze the hidden webshell activities. This prevents webshells from bypassing the detection due to the use of static detection rules.
- Suspicious process detection: The Server Guard server uses a data analysis rules engine to analyze the server process data collected by the client. By doing so, the server can detect suspicious processes such as reverse shells, mining processes, DDoS Trojans, worms, viruses, and hacking tools.
- Log collection: The client collects logs such as processes logs and network logs.

Scenarios

Server Guard is applicable to server security protection in the following scenarios:

- Use common software for website building

  In this scenario, attackers may intrude servers by exploiting vulnerabilities in common software. You can use Server Guard to detect and fix vulnerabilities.
- Use Web application services

  Attackers may steal website data through both internal and external Web services. You can use Server Guard to prevent attackers from launching attacks or controlling your servers.

12.4.1.4 Web Application Firewall

Web Application Firewall (WAF) protects the Web applications of cloud users against common Web attacks.

Different from traditional web application firewalls, Apsara Stack WAF uses intelligent semantic analysis algorithms to identify Web attacks. WAF also integrates a learning model to enhance its analysis capability so that it can meet your daily security protection requirements without relying on traditional rule libraries.

WAF protects the traffic of businesses on HTTP and HTTPS websites. In the WAF console, you can import certificates and private keys to enable end-to-end encryption. This prevents the interception of business data on the links.

WAF not only prevents common Web application attacks defined by Open Web Application Security Project (OWASP) but also mitigates HTTP flood attacks. In addition, WAF allows you to customize protection policies based on the businesses of your website to block malicious Web requests.

Features

The following table describes the features provided by WAF.

Feature	Description
Protection against common Web attacks	**Detects Structured Query Language (SQL) injections, cross-site scripting (XSS), intelligence, cross-site request forgery (CSRF), server-side request forgery (SSRF), Hypertext Preprocessor (PHP) deserialization, Java deserialization, Active Server Pages (ASP) code injections, file inclusion attacks, file upload attacks, PHP code injections, command injections, crawlers, and server responses.** **WAF provides five built-in protection templates, including the template with default protection policies, monitoring mode template, anti-DDoS template, template for financial customers, and template for Internet customers. WAF allows you to customize the decoding algorithms in the templates, enable or disable each attack detection module separately, and set the detection granularity.**

Feature	Description
HTTP flood mitigation	Allows you to set access frequency control rules for domain names and URLs to restrict the access frequency of IP addresses or sessions that meet the criteria or block these IP addresses or sessions. Restricts the access frequency of known IP addresses or sessions or block these IP addresses or sessions. The HTTP flood mitigation rules do not apply to IP addresses or sessions that have been added to the whitelist.
Custom and precise access control	Supports precise access control based on the following HTTP contents or their combinations: URI, GET parameters , decoded path, HOST header, complete cookie, POST parameters, complete body, HTTP status code, and response content.

How it works

WAF performs protocol parsing and in-depth decoding on the Web access traffic . It then calls the access control, rule detection, and semantic analysis engines to analyze the traffic and determines whether to allow or block the traffic based on the preset policies. Besides, WAF provides a good human-machine interaction interface for administrators to adjust protected websites and security policies.

Scenarios

WAF can be used for Web application protection in fields such as government, finance, insurance, e-commerce, online to offline (O2O), Internet Plus, and games. It provides the following features:

- Prevents website data leaks caused by SQL injections.
- Mitigates HTTP flood attacks by blocking a large number of malicious requests. This ensures the availability of your website.
- Prevents website defacement arising from Trojans to ensure the credibility of your website.
- Provides virtual patches that enable quick fix for newly discovered vulnerabil ities.

12.4.2 Optional security services

In addition to the security services provided by Apsara Stack Security Standard Edition, multiple optional security services are also provided to meet various security needs. We recommend that you choose optional security services based on your business needs.

12.4.2.1 DDoS Traffic Scrubbing

Backed by its large-scale and distributed operating system and more than a decade of experience in defending against security attacks, Alibaba Cloud has designed and developed the DDoS Traffic Scrubbing module based on the cloud computing architecture to protect the Apsara Stack platform against large amounts of distributed denial of service (DDoS) attacks.

Features

The following table describes the features provided by the DDoS Traffic Scrubbing module.

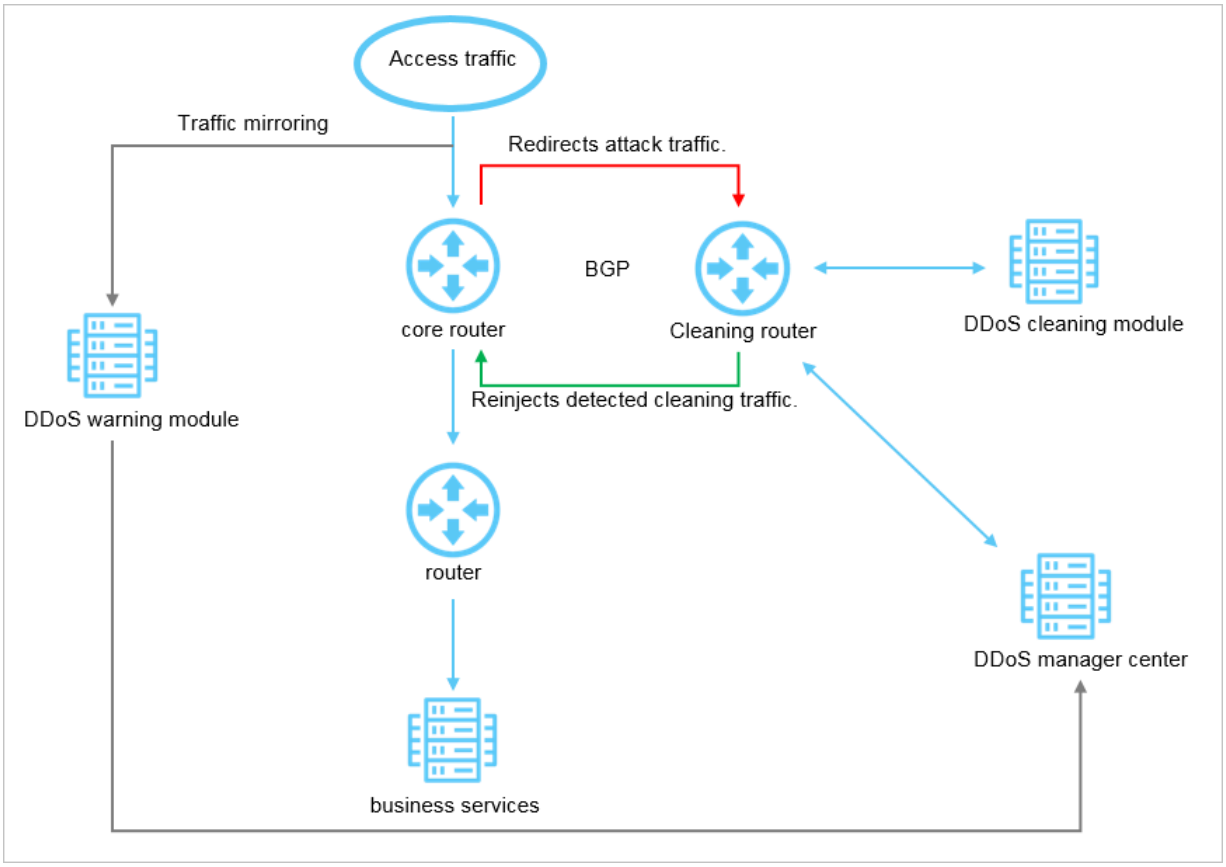
Feature	Description
Traffic scrubbing against DDoS attacks	Detects and prevents attacks such as SYN flood, ACK flood, ICMP flood, UDP flood, NTP flood, DNS flood, and HTTP flood.
DDoS attack display	Allows you to view DDoS attacks in the console and search for DDoS attacks by IP address, status, and event information.
DDoS traffic analysis	Allows you to monitor and analyze the traffic of a DDoS attack, and view the attack traffic protocol and the top 10 IP addresses that have launched most attacks.

How it works

After the Traffic Security Monitoring module detects unusual traffic, the DDoS Traffic Scrubbing module reroutes, scrubs, and reinjects the traffic, as shown in

Figure 12-2: Traffic scrubbing. **This mitigates DDoS attacks and ensures normal running of businesses.**

Figure 12-2: Traffic scrubbing



The Traffic Security Monitoring module sends information about the detected DDoS attacks to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module is connected to the border gateway device. When a DDoS attack is detected, this module configures a Border Gateway Protocol (BGP) path for the border gateway to reroute the attack traffic to the DDoS Traffic Scrubbing module. The DDoS Traffic Scrubbing module then scrubs the traffic based on the configured scrubbing policies, filters out unusual traffic, and reinjects the normal traffic to the border gateway.

Note:
Apsara Stack Security cannot scrub the traffic between internal networks.

Advantages

The DDoS Traffic Scrubbing module has the following feature advantages:

- **Detection of all common DDoS attacks**

  This module protects you from various DDoS attacks, such as HTTP flood, SYN flood, UDP flood, UDP DNS query flood, stream flood, ICMP flood, and HTTP GET flood, at the network layer, transport layer, and application layer. This module also informs you of the website defense status through real-time SMS messages.
- **Automatic response to attacks within one second**

  This module uses the world leading attack detection and prevention technologies. It can complete the protection process within one second, covering attack discovery, traffic rerouting, and traffic scrubbing. This module triggers traffic scrubbing when the traffic scrubbing thresholds are violated or when DDoS attacks are detected during network behavior analysis. This reduces network jitter and ensures the availability of your businesses in the case of DDoS attacks.
- **High scalability and high redundancy of anti-DDoS capabilities**

  With high scalability and high redundancy of the cloud computing architecture, this module can be easily scaled up to realize high scalability of anti-DDoS capabilities.
- **Bidirectional protection to avoid the abuse of cloud resources**

  This module not only protects your system against external DDoS attacks but also detects resource abuse in your cloud environment. If any of your cloud resources in Apsara Stack is used to launch DDoS attacks, the Traffic Security Monitoring module will cooperate with Server Guard to restrict the network access of the hijacked resource and generate an alert.

12.4.2.2 Sensitive Data Discovery and Protection

Sensitive Data Discovery and Protection (SDDP) is a data security service used to detect and protect sensitive data in Apsara Stack big data services.

SDDP uses Alibaba's big data analytics capabilities and artificial intelligence (AI) technologies to detect and classify sensitive data based on your business requirements. It can also both dynamically and statically mask sensitive data, monitor dataflows, and detect abnormal activities. It provides visible, controllable, and industry-compliant security protection for your sensitive data by using precise detection and analysis. SDDP can detect and protect sensitive data in a variety of Apsara Stack big data services, such as MaxCompute, Object Storage Service (OSS), Table Store, and ApsaraDB for RDS.

Features

The following table describes features of SDDP.

Feature		Description
Classification and detection of sensitive data	Detection of new data	A department administrator can authorize SDDP to scan and protect data assets based on business requirements. SDDP only scans and monitors authorized data assets.
	Sensitive data classification	SDDP can classify sensitive data detected in big data services, such as MaxCompute, OSS, Table Store, and ApsaraDB for RDS. You can define classification rules for sensitive data by using keywords, regular expressions, or other methods.
	Sensitive data detection	SDDP has built-in algorithms for detecting sensitive data, and uses file clustering, deep neural networks, and machine learning to detect sensitive images, text, and fields.
Management of sensitive data permissions	Asset permissions detection	SDDP can redirect you to pages that display the permissions of data assets and allows you to view the accounts that have permissions to access those assets. The data assets include MaxCompute projects, MaxCompute tables, MaxCompute columns, MaxCompute packages , OSS buckets, Table Store instances, and Table Store tables.
	Account permissions detection	SDDP allows you to view all accounts in a department and search for departments or accounts in fuzzy search mode. SDDP displays relationships between departments and accounts in a hierarchical and visible layout.
	Abnormal permission s usage detection	SDDP automatically detects abnormal permissions usage in big data services, such as MaxCompute, OSS, and Table Store.

Feature		Description
Monitoring of dataflows and operations	Dataflow monitoring	SDDP monitors dataflows among entities , including data storage services (such as MaxCompute, OSS, and Table Store), data transmission services (such as Datahub and CDP), the data stream processing service Blink, external databases, and external files. It displays dataflows and abnormal activities on dynamic graphs. This way, you can click an abnormal activity on a graph to redirect to the page for handling the abnormal activity.
	Abnormal data operation detection	SDDP detects abnormal operations in big data services, such as MaxCompute, OSS, and Table Store.
	Abnormal dataflow detection	SDDP detects abnormal dataflows (including abnormal downloads) in big data services, such as MaxCompute, OSS, and Table Store.
	Detection rule customization	SDDP allows you to customize rules for detecting abnormal dataflows and operations based on algorithms.
Abnormal activity processing	Configuration for abnormal activity detection	SDDP allows you to configure thresholds and rules for detecting abnormal activities, such as abnormal dataflows, permissions usage, and data operations.
	Abnormal activity processing	SDDP processes abnormal activities with a built-in console. You can search for abnormal activities by department, event type, account, processing status, or time of occurrence.
	Abnormal activity statistics	SDDP collects statistics on the processing status of abnormal activities, including abnormal dataflows, permissions usage, and data operations, and then dynamically displays these statistics.

Feature		Description
Static data masking	Static data masking	SDDP statically masks sensitive data in big data services, such as MaxCompute, OSS, Table Store, and ApsaraDB for RDS. It supports the following masking algorithms : hash masking, shield masking, substitution masking, conversion masking, encryption masking, and shuffle masking.
Intelligent audit	Intelligent audit	SDDP collects and audits the operation logs of big data services, such as MaxCompute, OSS, and ApsaraDB for RDS.

Scenarios

- Complies with laws and regulations on personal information protection.

  SDDP detects personal information in large amounts of data, automatically marks risk levels for personal information, and effectively detects data leaks. By using SDDP, enterprises can ensure that their systems comply with laws and regulations on personal information protection.
- Classifies and protects enterprise sensitive data.

  SDDP classifies and detects sensitive data, manages data permissions, and identifies abnormal activities (such as abnormal dataflows, permissions usage, and data operations) based on specified rules. This way, enterprises can properly protect their sensitive data of diverse classifications.
- Handles data leaks.

  SDDP detects abnormal activities based on specific rules and allows you to centrally summarize and handle these activities. This helps enterprises process data leaks online and provides effective support for security O&M.

Benefits

As a data security module of Alibaba Cloud Security, SDDP can detect and protect sensitive data in real-time computing services (such as Blink, Datahub, and Table Store) and offline computing services (such as MaxCompute and OSS). It can detect structured, semi-structured, and unstructured sensitive data based on the same standards. SDDP provides the following benefits:

- **Precise detection**

  SDDP uses a built-in rule engine, a natural language processing model, and a neural network model to precisely detect sensitive personal information, sensitive system configurations, and confidential documents in a large amount of data.

- **Closed-loop management**

  SDDP implements closed-loop management that covers detection, protection, and handling to help enterprises effectively avoid risks.

- **Intelligent detection**

  SDDP provides an intelligent and multi-level filtering model to effectively detect abnormal activities and meet operational requirements.

- **Flexible definition**

  SDDP allows you to customize a variety of data based on your business requirements, such as rules for detecting sensitive data, definitions of sensitive data, and thresholds and rules for detecting abnormal activities.

13 Apsara Stack DNS

13.1 What is Apsara Stack DNS?

Apsara Stack DNS is an Apsara Stack service that resolves domain names. Apsara Stack DNS translates the requested domain names based on the rules and policies you set for domain names and IP addresses, and redirects the requests from the client to the corresponding cloud services, enterprise business systems, or services provided by Internet service providers.

Apsara Stack DNS provides basic domain name resolution and scheduling services for VPC environments.

- **Weight-based scheduling to meet your business needs for zone active-active recovery, active geo-redundancy, zone-disaster recovery, and geo-disaster recovery**
- **Domain name isolation by tenant to meet your requirements for department resource isolation**

You can perform the following operations on your VPC by using Apsara Stack DNS:

- **Access other ECS instances deployed in your VPC.**
- **Access cloud service instances provided by Apsara Stack.**
- **Access custom enterprise business systems.**
- **Access Internet services and businesses.**
- **Establish network connections between Apsara Stack DNS and a user-created DNS over a leased line.**

13.2 Benefits

Apsara Stack DNS is a key network service. It controls the data flow of Apsara Stack, resolves domain names, balances and schedules server loads, and connects Apsara Stack to on-premises data centers. Apsara Stack DNS offers multiple solutions

for cloud environment deployment, high availability of data centers, server load balancing, and disaster recovery.

Management of enterprise domain names

Apsara Stack DNS manages and resolves enterprise domain names. Apsara Stack DNS provides the following functions:

- **DNS resolution and reverse DNS resolution for domain names of cloud service instances, such as ECS instances.**
- **DNS resolution and reverse DNS resolution for your internal domain names.**
- **Addition, modification, and deletion of DNS records of the following types: A, AAAA, CNAME, MX, PTR, TXT, SRV, NAPTR, CAA, and NS.**
- **Addition of multiple DNS records of the A, AAAA, and PTR types on one host. By default, the resolution results include all matching records. The records can be randomly rotated for load balancing.**
- **Addition of multiple DNS records of the A, AAAA, and CNAME types on one host. To implement global traffic scheduling, resolution results are returned based on the weight of each record.**

Flexible network creation and merging

Apsara Stack DNS forwards queries for enterprise domain names to allow you to flexibly create or merge networks. Apsara Stack DNS offers the following forwarding modes:

- **Forward queries for all domain names.**
- **Forward queries for specific domain names.**

Internet access from enterprise servers

When the Internet is accessible, Apsara Stack DNS supports recursive DNS for Internet domain names. You can access Internet services by using enterprise servers.

Tenant isolation

Apsara Stack DNS provides VPC-based private zone management and resolution features, which allow you to isolate DNS data and resolution by tenant.

You only need to purchase one set of Apsara Stack DNS services to implement VPC isolation.

Unified management platform

The Apsara Stack DNS console is integrated into the Apsara Stack console. You can use the same account to manage both Apsara Stack DNS and other Apsara Stack services. In terms of control and management, Apsara Stack DNS has the following advantages:

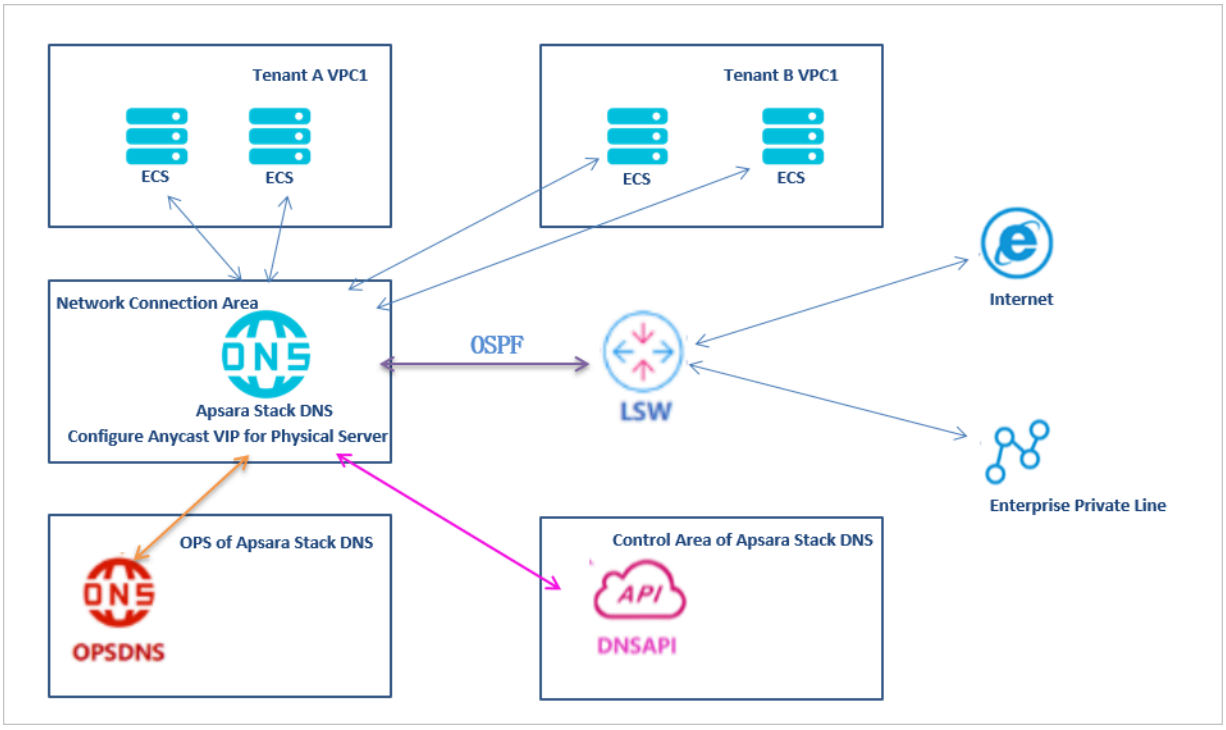
- **Easy-to-use web-based data and service management.**
- **Cluster-based deployment and high scalability. You can add DNSs based on your needs.**
- **Multi-zone deployment, zone active-active recovery, and zone-disaster recovery.**
- **Anycast-based deployment, which delivers high availability and disaster recovery.**

API openness

You can call the Apsara Stack DNS API to integrate Apsara Stack DNS with other systems.

13.3 Architecture

Figure 13-1: Apsara Stack DNS architecture



Apsara Stack DNS architecture description

- **Two or more physical servers are deployed in the network access area.**
- **Two control interfaces are bound and uplinked to an access switch (ASW). The gateway is the default gateway of the internal network.**
- **Two service interfaces are uplinked to a LAN switch (LSW) by using Equal-Cost Multi-Path (ECMP) routing. These interfaces advertise anycast VIP routes in compliance with Open Shortest Path First (OSPF), and are connected to the Internet.**
- **The control system is deployed in a container in the control area.**

13.4 Features

Internal domain name management

Apsara Stack DNS provides management for internal domain names. You can register, search, and delete internal domain names and add descriptions. You can add, delete, and modify the following types of DNS records: A, AAAA, CNAME, MX , PTR, TXT, SRV, NAPTR, CAA, and NS. With the internal domain name resolution feature of Apsara Stack DNS, you can resolve domain names for servers in a VPC . The DNS endpoint is deployed in anycast mode. For disaster recovery, DNS can switch services between servers that are located in different data centers.

Domain name forwarding management

Apsara Stack DNS can forward DNS queries for some or all domain names to other DNS servers.

Two forwarding modes are available: forward all requests with recursion and forward all requests without recursion.

- **Forward all requests without recursion: Forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names or the request is timed out, a message is returned to the DNS client indicating that the query failed.**
- **Forward all requests with recursion: Forwards DNS requests to the target DNS server. If the target DNS server cannot resolve the domain names, the local DNS server is used to resolve them.**

Recursive resolution

With recursive resolution, you can resolve Internet domain names to access Internet services.

Tenant isolation (standard edition only)

With VPC-based private zone management and resolution features, enterprises can isolate DNS data and resolution by tenant.

14 MaxCompute

14.1 What is MaxCompute?

14.1.1 Overview

MaxCompute is an offline data processing service developed by Alibaba Cloud based on the Apsara system. It is capable of processing large volumes of data. MaxCompute can process terabytes or petabytes of data in scenarios that do not have high real-time processing requirements. MaxCompute is used in fields such as log analysis, machine learning, data warehousing, data mining, and business intelligence.

MaxCompute provides an easy-to-use approach to analyze and process large amounts of data without deep knowledge of distributed computing. MaxCompute is widely implemented by Alibaba across its businesses for tasks such as data warehousing and BI analysis for large Internet enterprises, website log analysis, e-commerce transaction analysis, and exploration of user characteristics and interests.

MaxCompute provides the following features:

- **Data channel**
  - **Tunnel: provides highly-concurrent offline upload and download services. The tunnel service enables you to upload or download large volumes of data to or from MaxCompute. You must use a Java API to access the tunnel service.**
  - **DataHub: provides real-time upload and download services. Unlike data uploaded through the tunnel service, data uploaded through DataHub is available immediately.**
- **Computing and analysis**
  - **SQL: MaxCompute stores data in tables, and provides SQL query capabilities to manipulate the data. MaxCompute can be used as database software capable of processing terabytes or petabytes of data. MaxCompute SQL does not support transactions, indexes, or operations such as UPDATE and DELETE. The SQL syntax used in MaxCompute is different from that in Oracle and MySQL. SQL statements from other database engines cannot be migrated**

seamlessly to MaxCompute. MaxCompute SQL responds to queries within a few minutes or seconds, instead of milliseconds. MaxCompute SQL is easy to learn. You can get started with MaxCompute SQL based on your prior experience of database operations, without having a deep understanding of distributed computing.

- MapReduce: Initially proposed by Google, MapReduce is a distributed data processing model that has become popular and widely implemented for a variety of business scenarios. This topic briefly describes the MapReduce model. You must have a basic knowledge of distributed computing and relevant programming experience before using MapReduce. MapReduce provides a Java API.
- Graph: a processing framework designed for iterative graph computing. Graph computing jobs use graphs to build models. A graph is a collection of vertices and edges that have values. MaxCompute Graph iteratively edits and evolves graphs to obtain analysis results.
- Unstructured data access and processing (integrated computing scenarios): Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

  MaxCompute allows you to process the following data sources by creating external tables:

  - Internal data sources: OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.
  - External data sources: HDFS (Open Source), ApsaraDB for MongoDB, and Hbase.
- Unstructured data access and processing in MaxCompute: By reading data from and writing data to volumes, MaxCompute can store unstructured data, which otherwise must be stored in an external storage system.

· Spark on MaxCompute: a big data analytics engine designed by Alibaba Cloud to provide big data processing capabilities for Alibaba, government agencies, and enterprises.

- **Elasticsearch on MaxCompute: an enterprise-class system to retrieve information from large volumes of data and provide near-real-time search performance for government agencies and enterprises.**

14.1.2 Features and benefits

Features

- **MaxCompute is a distributed system designed for big data processing. MaxCompute is a core service in the Alibaba Cloud computing solution and is used to store and compute structured data. It is also a basic computing component of the Alibaba Cloud big data platform. MaxCompute is designed to support multiple tenants and provide data security and horizontal scaling. Based on an abstract job processing framework, the service provides centralized programming interfaces for various data processing tasks of different users.**
- **MaxCompute uses a distributed architecture that can be scaled as needed.**
- **MaxCompute provides an automatic storage and fault tolerance mechanism to ensure high data reliability.**
- **MaxCompute allows all computing tasks to run in sandboxes to ensure high data security.**
- **MaxCompute uses RESTful APIs to provide services.**
- **MaxCompute can upload or download high-concurrency, high-throughput data.**
- **MaxCompute supports two service models: the offline computing model and the machine learning model.**
- **MaxCompute supports data processing methods based on programming models such as SQL, MapReduce, Graph, and MPI.**
- **MaxCompute supports multiple tenants, allowing multiple users to collaborate on data analysis.**
- **MaxCompute provides user permission management based on ACLs and policies, allowing you to configure flexible data access control policies to prevent unauthorized access to data.**
- **MaxCompute provides Elasticsearch on MaxCompute for enhanced applications.**
- **MaxCompute provides Spark on MaxCompute for enhanced applications.**
- **MaxCompute supports access and processing of unstructured data.**
- **MaxCompute supports the deployment of multiple clusters in a single region.**
- **MaxCompute can be deployed across different regions.**

Benefits

- **The only big data cloud service in China and a real data sharing platform: Warehousing, mining, analysis, and sharing of data can all be performed on the same platform. Alibaba Group implements this unified data processing platform in several of its own products such as Aliloan, Data Cube, DMP (Alimama), and Yu'e Bao.**
- **Support for large numbers of clusters, users, and concurrent jobs: A single cluster can contain more than 10,000 servers and maintain 80% linear scalability. A single MaxCompute instance can support more than 1 million servers in multiple clusters without restrictions (linear scalability is slightly affected). It can be deployed across multiple data centers in a zone. It can support over 10,000 users, over 1,000 projects, and over 100 departments (of multi-tenants). It can support more than 1 million jobs (daily submitted jobs on average) and more than 20,000 concurrent jobs.**
- **Big data computing at your fingertips: You do not have to worry about the storage difficulties and the prolonged computing time caused by the increasing data volume. MaxCompute automatically expands the storage and computing capabilities of clusters based on the volume of data to process, allowing you to focus on maximizing the efficiency of data analysis and mining.**
- **Out-of-the-box service: You do not have to worry about cluster construction, configuration, and O&M. Only a few simple steps are required to upload and analyze data in MaxCompute.**
- **Secure and reliable data storage: User data is protected by the multi-level data storage and access security mechanisms against loss, theft, and exposure. These mechanisms include multi-replica technology, read/write request authentication, and application and system sandboxes.**
- **Multi-tenancy for multi-user collaboration: You can have multiple data analysts in your organization to work together by configuring different data access policies, while ensuring that each analyst can only access data within their own permissions. This maximizes work efficiency while ensuring data security.**
- **Multi-region deployment: You can specify compute clusters to efficiently utilize resources. Data exchanges between clusters are completed within MaxCompute, and data replication and synchronization between clusters are managed based on configured policies. Because cross-region data processing is no longer involved, the waiting time for data processing is significantly reduced.**

14.1.3 Benefits

Compared with traditional databases, MaxCompute has the following benefits.

Table 14-1: Comparison of benefits

Benefit	Traditional databases	MaxCompute
System scalability	Disks cannot be shared across more than 100 nodes. Table and database sharding causes application data collision, resulting in massive computing overhead. This significantly compromises application analysis capabilities.	MaxCompute supports more than 10,000 nodes that can store more than 1.5 EBs of data. For example, during Alibaba's Double 11 event, MaxCompute processed more than 300 PBs of data in six hours.
Data type support	Cannot process unstructured data.	Can process both structured and unstructured data.
High availability	Redundant storage solutions are not available. Traditional backup and recovery approaches are inapplicable to large volumes of data (measured in PBs), and a single point of failure can cause the entire database to become unavailable.	Provides the shared-nothing architecture and multi-replica data model. This eliminates single points of failure.
Complex computing capability	Iterative computing and graph computing capabilities are not available. The disk sharing technology and complex computing operations result in massive data exchanges between nodes, imposing tremendous bandwidth pressure.	Provides distributed storage and multiple computing frameworks such as MR, SQL, iterative computing, MPI, and graph computing.

Benefit	Traditional databases	MaxCompute
Concurrency	A single large-scale computing task (such as index computing) can consume all system resources, and incur network and disk (data dictionary) bottlenecks. This makes highly concurrent access impossible.	Provides comprehensive multi-tenant isolation and resource management tools, so that you can easily view cluster resources and manage the resources used by each service. It can support up to 10,000 concurrent access requests.
Performance support	The indexing mechanism makes it difficult to support analytical applications of real-time data. Large amounts of data collision cause analytical predictions to take more than 24 hours, resulting in a performance bottleneck.	Focuses on the concurrent computing of large amounts of data. It provides available real-time data, and multiple high-performance computing capabilities, such as high-performance large-scale offline computing, real-time multi-dimensional analysis of large amounts of data, and stream computing.

14.1.4 Scenarios

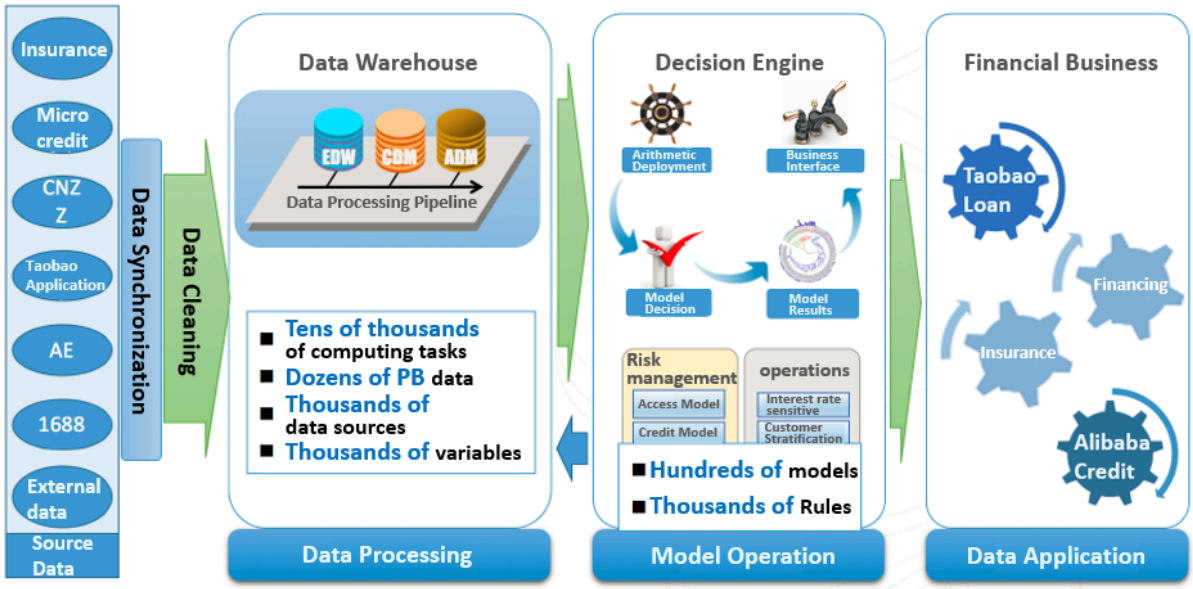
MaxCompute is designed for use in three big data processing scenarios:

- Establishment of SQL-based large data warehouses and BI systems
- Development of big data applications based on MapReduce and MPI distributed programming models
- Development of big data statistics models and data mining models based on statistics and machine learning algorithms

The following describe some real-world scenarios.

Data warehouse construction

Figure 14-1: Data warehouse construction



MaxCompute enables you to easily build a cloud-based data warehouse. With MaxCompute capabilities such as partitioning, data table statistics, and table life cycle management, you can easily enhance the storage of historical data warehouse information, divide hot and cold tables, and control data quality.

Alibaba's financial data warehousing team has built a sophisticated and powerful data warehousing system based on MaxCompute. This system provides six layers: the source data layer, ODS layer, enterprise data warehousing layer, common dimensional modeling layer, application marketplace layer, and presentation layer.

- **The source data layer processes data from all sources, including Taobao, Alipay, B2B, and external data sources.**
- **ODS provides a temporary storage layer for data import.**
- **The enterprise data warehousing layer uses the 3NF modeling technique to divide data, including all historical data, by topic (such as item or shop).**
- **The common dimensional modeling layer uses the dimensional modeling approach to create modeling layers for general business applications. This layer shields the upper layers from changes in business requirements, and provides consistent and actionable data to the upper layers.**
- **The application marketplace layer is a demand-oriented layer that provides a data marketplace for specific applications.**

- **The presentation layer provides several data portals and services that can be accessed by applications.**

This system architecture inevitably involves tasks such as metadata management.

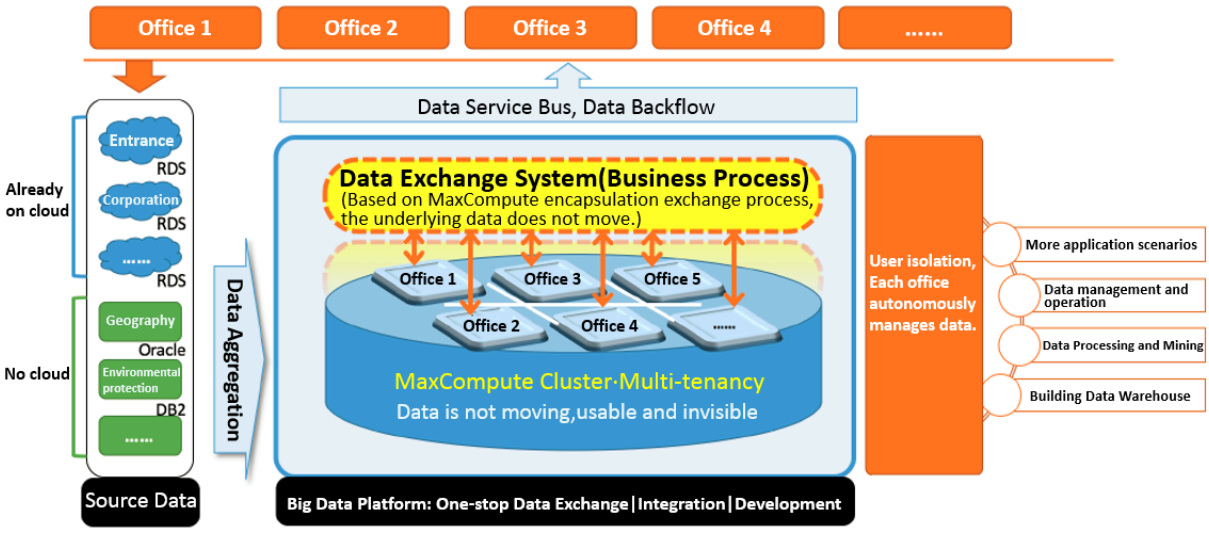
The financial data warehouse is used to perform offline computing tasks based on MaxCompute SQL. It also uses a series of metric rules and algorithms to make decisions offline for online decision-making.

MaxCompute-based data warehouses differ from traditional databases in the following ways:

- **Historical data storage: MaxCompute is able to store large amounts of data. You do not have to dump historical data to cheaper storage media as you would do in traditional databases.**
- **Partitioning: Traditional databases provide a wide range of partitioning methods such as range partitioning. MaxCompute provides fewer partitioning methods, but are sufficient for use in data warehousing scenarios. Whatever the method, you can build a data warehouse based on the same concept and principle as a table partition.**
- **Wide tables: MaxCompute stores data in fields, making it ideal for creating wide tables.**
- **Data integration: Traditional databases use stored procedures for data processing and integration. MaxCompute splits the logic of these operations into discrete SQL statements. Though the implementation is different, the algorithms are the same. In many years of experience, we found that splitting the operation logic into discrete SQL statements is clearer and more efficient, while stored procedures are more flexible and capable of processing complex logic.**

Big data sharing and exchange

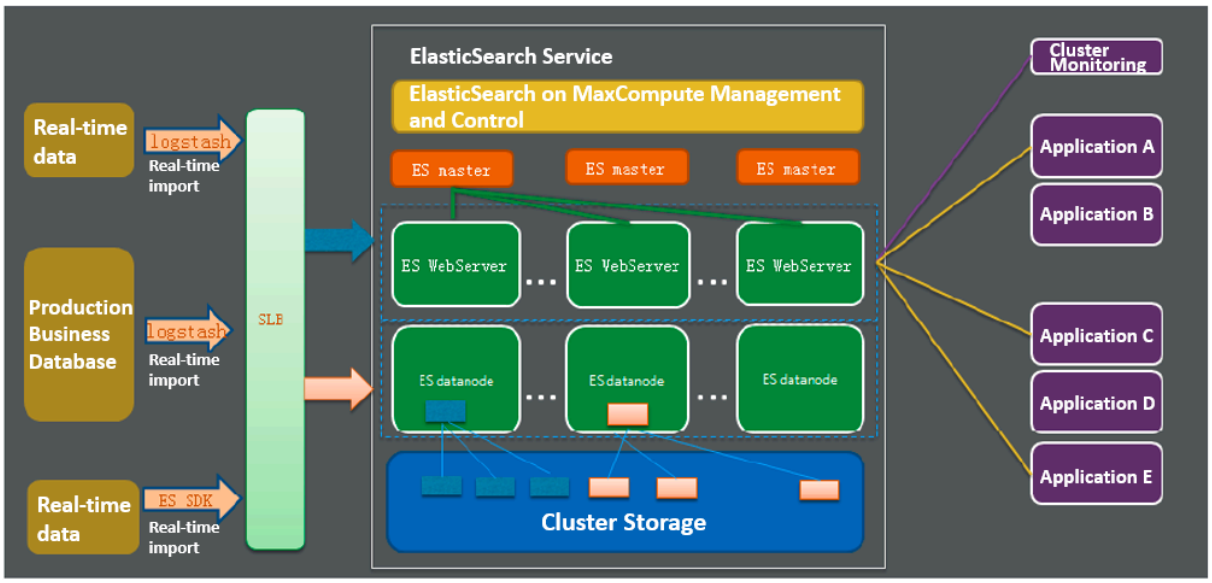
Figure 14-2: Big data sharing and exchange



MaxCompute provides a wide range of permission management methods and flexible data access control policies. MaxCompute provides a wide range of access control mechanisms, including the ACL authorization, role-based authorization, policy authorization, cross-project authorization, and label security mechanism. MaxCompute provides column-level security solutions. This can meet the security requirements within an organization or across multiple organizations. For projects that demand high security, MaxCompute provides the project protection mechanism to prevent data leakage, and provides logs of all user operations to facilitate retrospective audits.

Typical applications of Elasticsearch on MaxCompute

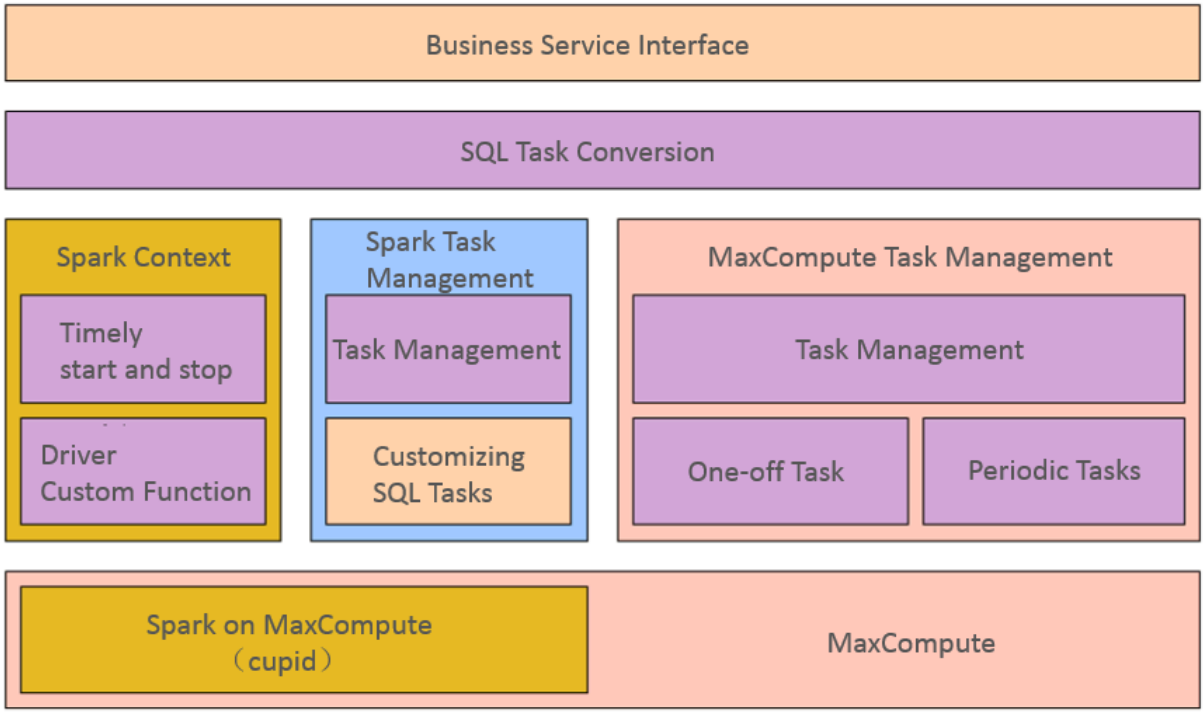
Figure 14-3: Typical applications



Elasticsearch on MaxCompute allows you to launch a set of Elasticsearch services by submitting jobs in a MaxCompute cluster. Native Elasticsearch code is not modified when applied in a project. Elasticsearch on MaxCompute runs in the same way as native Elasticsearch clusters.

Typical applications of Spark on MaxCompute

Figure 14-4: Typical applications



Spark on MaxCompute provides business computing platform and applications in Client mode. The preceding figure shows the application framework.

14.1.5 Service specifications

14.1.5.1 Software specifications

14.1.5.1.1 Overview

This section describes the software specifications of MaxCompute.

14.1.5.1.2 Control and service

Table 14-2: Specifications

Item	Description
Number of control nodes	Greater than or equal to 3.
Number of MaxCompute front-end servers	Greater than or equal to 2. MaxCompute front-end servers can be deployed together with control nodes.
Number of tunnels	Greater than or equal to 2. Tunnels can be deployed together with compute nodes.

Item	Description
Number of DataHubs	Greater than or equal to 2. DataHubs can be deployed together with compute nodes.

14.1.5.1.3 Data storage

Table 14-3: Specifications

Item	Description
Logical storage capacity per node	12 TB
Total storage capacity	The storage capacity can be scaled out by adding more nodes.

Note:
The size of logically stored data to a large extent determines the size of the cluster to be evaluated.

14.1.5.1.4 Size of a single cluster

Table 14-4: Specifications

Item	Description
Offline computing cluster	An offline computing cluster can contain 3 to 10,000 machines.

14.1.5.1.5 Projects

Table 14-5: Specifications

Item	Description
Creation of projects	Supported.
Acquisition of project metadata	Supported.
Deletion of projects	Supported.
Setting of the default lifecycle of tables	Supported.
Number of supported projects	Over 1,000

14.1.5.1.6 User management and security and access control

Table 14-6: Specifications

Item	Description
Cross-project access	**Supported. You can authorize cross-project access to organize tables and resources as packages and install them in other projects.**
Service (odps_server and tunnel) authentication and access control	**Supported. AccessKey ID and AccessKey Secret can be used to authenticate users and control their permissions.**
Prevention of data outflow from a project	**You can prevent data outflow and specify exceptions when necessary.**
Label-based security	**Label-based security (LabelSecurity) can be set to enable column-level access control.** **Note:** **LabelSecurity is a mandatory access control policy that provides a wide range of security level settings.**
Authorization to users	**Supported.**
Authorization to roles	**Supported. You can customize roles and assign roles to users. Different roles are granted different permissions.**

Item	Description
Project-specific authorization	The following permissions can be granted on a project: · View project information (excluding any project objects), such as the creation time. · Update project information (excluding any project objects), such as comments. · View the list of all object types in the project. · Create tables in the project. · Create instances in the project. · Create functions in the project. · Create resources in the project. · Create volumes in the project. · Grant all preceding permissions.
Table-specific authorization	The following permissions can be granted on a table: · Read table metadata. · Read table data. · Modify table metadata. · Overwrite or add table data. · Delete the table. · Grant all preceding permissions.
Function-specific authorization	The following permissions can be granted on a function: · Read. · Update. · Delete. · Grant all preceding permissions.

Item	Description
Authorization for resources, instances, jobs, and volumes	The following permissions can be granted on a resource, instance, job, or volume: · Read. · Update. · Delete. · Grant all preceding permissions.

Item	Description
Sandbox protection	The sandbox mechanism can restrict access to system resources in MapReduce and UDF programs. Specific restrictions are as follows: · Direct access to local files is not allowed. You can only read resource information and generate log information through System.out and System.err. Note: You can view log information by running the Log command on the MaxCompute client. · Direct access to Apsara Distributed File System is not allowed. · JNI calls are not allowed. · Java threads cannot be created, and Linux commands cannot be executed by sub-threads. · Network access operations such as acquiring local IP addresses are not allowed. · Java reflection is not allowed. You cannot force access to protected or private members to be valid.
Control over the quotas of storage and computing resources	Supported. You can limit the number of files and used disk capacity in a project. You can also use quotas to limit the available CPU and memory capacity of the project.

14.1.5.1.7 Resource management and task scheduling

Table 14-7: Specifications

Item	Description
File count quota and storage capacity quota	The quotas vary with projects.
Configuration of CPU quota for a resource group	You can configure the minimum or maximum number of virtual CPUs that can be used by a resource group.
Configuration of memory quota for a resource group	You can configure the minimum or maximum amount of virtual memory that can be used by a resource group.
Resource preemption	Preemption of resources within a quota group is supported.
Task scheduling methods	Fair scheduling and first-in-first-out (FIFO).
Configuration of task priorities	By default, task priorities are assigned in a project. You can configure the priorities as needed.
Restart of a failed task	Supported.
Speculative execution of a task	Supported.

14.1.5.1.8 Data tables

Table 14-8: Specifications

Item	Description
Data storage methods	CFile data exclusive to MaxCompute is stored in columns in Apsara Distributed File System.
Data compression	Supported. The efficiency of compression is dependent on the data format. The compression ratio between the original and compressed data is 3:1. Infrequently accessed data can be archived in RAID to reduce the storage space it occupies by 50%.
Lifecycle	Supported.
Basic data types	BigInt, String, Boolean, Double, DateTime, and Decimal.

Item	Description
Partitions	Supported. Only String type partitions are supported.
Maximum number of columns	1,024
Maximum number of partitions	60,000
Partition levels	A table can contain up to five partition levels.
Views	Supported. A view can only contain one valid SELECT statement. Materialized views are not supported.
Statistics	Supported. You can define statistical metrics for data tables and view, analyze, and delete statistics.
Comments	Supported. You can make comments for both tables and columns. Comments can be up to 1024 characters in length.

14.1.5.1.9 SQL

14.1.5.1.9.1 DDL

Table 14-9: Specifications

Item	Description
Creation of tables	Supported.
Deletion of tables	Supported.
Renaming of tables	Supported.
Creation of views	Supported.
Deletion of views	Supported.
Renaming of views	Supported.
Adding of partitions	Supported.
Deletion of partitions	Supported.
Adding of columns	Supported.
Modification of column names	Supported.
Modification of comments	Supported. You can modify comments for tables and columns.

Item	Description
Modification of the lifecycle of tables	Supported.
Disabling of the lifecycle for specific table partitions	Supported. The command syntax is as follows: `ALTER TABLE table_name [partition_spec] ENABLE|DISABLE LIFECYCLE`
Emptying of data from non-partitioned tables	Supported. The command syntax is as follows: `TRUNCATE TABLE table_name`
Modification of table owners	Supported.
Modification of the time when a table or partition was last modified	Supported.

14.1.5.1.9.2 DML

Table 14-10: Specifications

Item	Description
Dynamic partition filtering	Supported. This technique can reduce the amount of data to be read. The command syntax is as follows: `select_statment FROM from_statement WHERE PT1 IN (SUBQUERY) AND PT2 IN ( SUBQUERY)... ;`
Multiple outputs	Supported. A single SQL statement can contain up to 128 outputs. **Note:** In each output, you can only specify once whether to target a partition in a partitioned table or target a non-partitioned table.
Data update and overwriting	Supported. Batch update is supported.
Aggregation	Supported.

Item	Description
Sorting	Supported. Sorting must be performed with the limit syntax.
Nested subqueries	Supported.
Joins	Supported. SQL joins include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.
UNION ALL	Supported.
CASE WHEN	Supported.
Relational operations	Supported.
Mathematical operations	Supported.
Logical operations	Supported.
Implicit conversions	Supported.
MAPJOIN	Supported. To speed the JOIN operation when volume of data is small, SQL loads all specified small tables into the memory of a program executing the JOIN operation. The default maximum data size is 512 MB. The maximum size cannot exceed 2 GB. Up to six small tables can be specified. **Note:** Take note of the following limits: · The left table of a LEFT OUTER JOIN clause must be a large table. · The right table of a RIGHT OUTER JOIN clause must be a large table. · Both the left and right tables of an INNER JOIN clause can be large tables. · MAPJOIN cannot be used in a FULL OUTER JOIN clause. · MAPJOIN supports small tables as subqueries. · When MAPJOIN is used and a small table or subquery is referenced, you must reference the alias of the small table or subquery. · MAPJOIN supports both non-equivalent JOIN conditions and multiple conditions connected by using OR statements.

Item	Description
Query of the execution plans of DML statements	Supported. The description of the final execution plan corresponding to a DML statement can be displayed. The command syntax is as follows: `EXPLAIN <DML query>;`

14.1.5.1.9.3 Built-in functions

Table 14-11: Specifications

Item	Description
Built-in functions	Supported. Built-in functions include string functions, date functions, mathematical functions, regular functions, and window functions.

14.1.5.1.9.4 User-defined functions

Table 14-12: Specifications

Item	Description
Scalar functions	Supported. You can use the Java SDK and Python SDK to write scalar functions.
Aggregate functions	Supported. You can use the Java SDK and Python SDK to write aggregate functions.
Table functions	Supported. You can use the Java SDK and Python SDK to write table functions .
Implicit conversions	Supported.

14.1.5.1.10 MapReduce

14.1.5.1.10.1 Programming support

Table 14-13: Specifications

Item	Description
Java language	Supported.

Item	Description
Standalone debugging mode	Supported.
Extended MapReduce model	Supported. A Map operation can be followed by any number of Reduce operations. Example: Map-Reduce-Reduce.

14.1.5.1.10.2 Job size

Table 14-14: Specifications

Item	Description
Maximum number of mappers	100,000
Maximum number of reducers	2,000
Setting of the number of mappers and reducers	Supported. You can change the number of mappers by changing the input volume of each Map worker. By default, the number of reducers is set at 25% of the number of mappers. You can change this proportion to suit your business needs.
Setting of the memory of mappers and reducers	Supported. The default memory of a mapper or reducer is 2 GB.

Note:
The maximum numbers of mappers and reducers are related to the cluster size.

14.1.5.1.10.3 Input and output

Table 14-15: Specifications

Item	Description
Input and output of a table	Supported.
Processing of unstructured data	Supported. Volumes are suited to store unstructured data. MaxCompute MapReduce can be used to process unstructured data.
Input and output of multiple tables	Supported. The numbers of inputs and outputs cannot exceed 128.

Item	Description
Reading of resources	Supported. A single task can reference up to 256 resources. The total size of all referenced resources cannot exceed 2 GB. Note: The maximum number of read attempts for a resource is 64.

14.1.5.1.10.4 MapReduce computing

Table 14-16: Specifications

Item	Description
Custom setup, map, and cleanup methods of mappers	Supported.
Custom setup, reduce, and cleanup methods of reducers	Supported. Transmitted messages are processed in the next iteration.
Custom partition columns or partitioners	Supported.
Configuration of mapper output columns to be sorted and grouped by keys	Supported. Note that custom key comparators are not supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Map-only jobs	Supported. To implement Map-only jobs, set the number of Reduce jobs to 0.
Configuration of job priorities	Supported.

14.1.5.1.11 Graph

14.1.5.1.11.1 Programming support

Table 14-17: Specifications

Item	Description
Java language	Supported.

Item	Description
Standalone debugging mode	Supported.

14.1.5.1.11.2 Job size

Table 14-18: Specifications

Item	Description
Maximum number of concurrent workers	1,000
Custom worker CPU and memory	Supported. By default, a worker has two CPU cores and 4 GB of memory. A worker can have up to eight CPU cores and 12 GB of memory.

14.1.5.1.11.3 Graph loading

Table 14-19: Specifications

Item	Description
Loading of graph data from MaxCompute tables	Supported.
Division of graphs by vertex	Supported.
Custom partitioners	Supported.
Custom split size	Supported. The default split size is 64 MB.
Custom conflict logic upon data loading	Supported. For example, creating duplicate vertices and edges is considered a conflict logic.

14.1.5.1.11.4 Iterative computing

Table 14-20: Specifications

Item	Description
Bulk Synchronous Parallel (BSP) computing model	Supported.
Transmission of messages between vertices	Supported. Transmitted messages are processed in the next iteration.

Item	Description
Multiple iteration termination conditions	1. The maximum number of iterations is reached. 2. All vertices enter the halted state. 3. An aggregator determines to terminate the iteration.
Automatic checkpoint mechanism	Supported.
Custom aggregators	Supported.
Custom combiners	Supported.
Custom counters	Supported. A single job cannot have more than 64 custom counters.
Custom conflict logic	Supported. For example, sending messages to a non-existent vertex is considered a conflict logic.
Writing of computing results to MaxCompute tables	Supported.
Configuration of job priorities	Supported.

14.1.5.1.12 Processing of unstructured data

14.1.5.1.12.1 Processing of Table Store data

Table 14-21: Specifications

Item	Description
Table Store data types	A variety of data types are supported.

14.1.5.1.12.2 Processing of OSS data

Table 14-22: Specifications

Item	Description
User-defined split and range functions	Supported.
User-defined maximum number of concurrent mappers	Supported.
User-defined file list	Supported.

14.1.5.1.12.3 Multiple data sources

Table 14-23: Specifications

Item	Description
Support for various open-source data formats through the STORED AS syntax	**Supported data formats include PARQUET, ORC, SEQUENCEFILE, TEXTFILE, and AVRO.**

14.1.5.1.13 Spark on MaxCompute

14.1.5.1.13.1 Programming support

Table 14-24: Specifications

Item	Description
Native Apache Spark APIs	**Supported. You can use native Spark APIs to write code and process data stored in MaxCompute.**
Native methods to submit Spark jobs	**Supported.**
Multiple native Spark components	**Spark SQL, Spark MLlib, GraphX, and Spark Streaming are currently supported.**
Multiple programming languages	**MaxCompute data can be processed using Scala, Python , Java, and R languages.**

14.1.5.1.13.2 Data sources

Table 14-25: Specifications

Item	Description
Processing of unstructured data	**Supported. You can use Spark APIs to write code and process data stored in OSS and Table Store.**
Processing of data from MaxCompute tables and resources	**Supported.**

14.1.5.1.13.3 Scalability

Table 14-26: Specifications

Item	Description
Deep integration of Spark and MaxCompute	**Supported. Spark and MaxCompute share cluster resources. Spark resources can be scaled from large-scale MaxCompute clusters.**

14.1.5.1.14 Elasticsearch on MaxCompute

14.1.5.1.14.1 Programming support

Table 14-27: Specifications

Item	Description
Native Elasticsearch APIs	**Supported.**

14.1.5.1.14.2 System capabilities

Table 14-28: Specifications

Item	Description
Real-time analysis and retrieval of data at the petabyte level	**Supported.**
Web-based display for basic server metrics	**Supported. A user-friendly O&M platform for index databases and full-text retrieval clusters can be used to monitor the status of index databases and machines in real time.**
Data snapshot technology based on Apsara Distributed File System	**Supported. Rapid data backup and recovery can be performed to ensure data reliability.**
Millisecond-level response to keyword-based and comprehensive searches and second-level response to fuzzy searches	**Supported.**
Real-time analysis and retrieval of imported data and query response times within 500 milliseconds	**Supported. The storage architecture is powered by the distributed cache-accelerated block device technology.**

Item	Description
In-memory off-heap storage and processing of index data and fine-grained memory management	Supported.

14.1.5.1.15 Other extensions

The following extended plug-ins and tools are both client-specific and open-source. You can download the plug-ins and tools at *https://github.com/aliyun/*.

Table 14-29: Specifications

Item	Description
R language support	RODPS is a plug-in for the MaxCompute client to support the R language.
Plug-ins and tools	Eclipse plug-ins and command line tools are available .
OGG	OGG plug-ins synchronize data from OGG to DataHub.
Flume	Flume plug-ins synchronize data from Flume to DataHub.
FluentD	FluentD plug-ins synchronize data from FluentD to DataHub.
JDBC	JDBC interfaces are partially supported.
Sqoop	Sqoop can be used to exchange data with MaxCompute.

14.1.5.2 Hardware specifications

The following table lists the hardware specifications of MaxCompute.

Table 14-30: Hardware specifications

Node type	Server configuration	Number of nodes	Description
Management node	· CPU: dual-socket 8-core or higher · Memory: 256 GB or higher · Disk: two 4 TB NVMe U.2 SSDs · NIC: two 10 GE NICs for network bonding	N/A	We recommend that you use Intel Platinum 81xx series processors or higher configurations.

Node type	Server configuration	Number of nodes	Description
Control node	· CPU: dual-socket 8-core or higher · Memory: 128 GB or higher · Disk: one 4 TB SATA HDD with 7200 RPM performance · NIC: two 10 GE NICs for network bonding	8/13	· We recommend that you use Intel Platinum 81xx series processors or higher configurations. · When the number of data nodes is less than 500, the number of control nodes is 8. When the number of data nodes is more than 500, the number of control nodes is 13. · We recommend that you deploy data nodes in containers when the number of data nodes is less than 500. · When all control nodes are physical servers and the number of data nodes is less than 1,000, you can implement a hybrid deployment of control nodes and data nodes based on your actual needs. · The system disk capacity is greater than or equal to 240 GB.

Node type	Server configuration	Number of nodes	Description
Hybrid deployment of management nodes and control nodes	· CPU: dual-socket 8-core or higher · Memory: 256 GB or higher · Disk: one 4 TB NVMe U.2 SSD · NIC: two 10 GE NICs for network bonding	N/A	· Hybrid deployment is recommended when the number of data nodes is less than 500 and is not expected to be increased. · Assume that the number of data nodes is approximately 500 and is expected to increase to more than 500. When you deploy the nodes for the first time, we recommend that you deploy them separately on physical servers.

Node type	Server configuration	Number of nodes	Description
Data node	· CPU: dual-socket 8-core or higher · Memory: 128 GB or higher · Disk: twelve 2 TB, 4 TB, 6 TB, or 8 TB SATA HDDs with 7200 RPM performance · NIC: two 10 GE NICs for network bonding	Depends on the amount of data.	· We recommend that you use Intel Golden 61xx series processors or higher configurations. · The recommended ratio of core quantity to memory capacity is 1:4. · We recommend that you add a 4 TB NVMe U.2 SSD when the number of cores is greater than or equal to 48. · Number of nodes = [(Total planned data volume × Data expanding rate (1.3) × Data compression rate (1) × Number of replicas (3))/ Disk utilization rate (0.85)/Disk formatting loss (0.9)/((Number of disks (12) - Number of system reserved blocks (1)) × Disk capacity (8 TB))] rounded up.

Note:

· We recommend that you use the preceding configurations in offline scenarios as needed.

- **We recommend that you do not use two or more machine types for compute nodes of MaxCompute.**
- **We recommend that you do not use both 1 GE and 10 GE NICs for MaxCompute.**
- **The configuration of machines to be added cannot be lower than that of the existing machines.**
- **The reuse of compute nodes needs to be evaluated together with the business side.**

14.1.5.3 Specifications of DNS resources

Table 14-31: Specifications

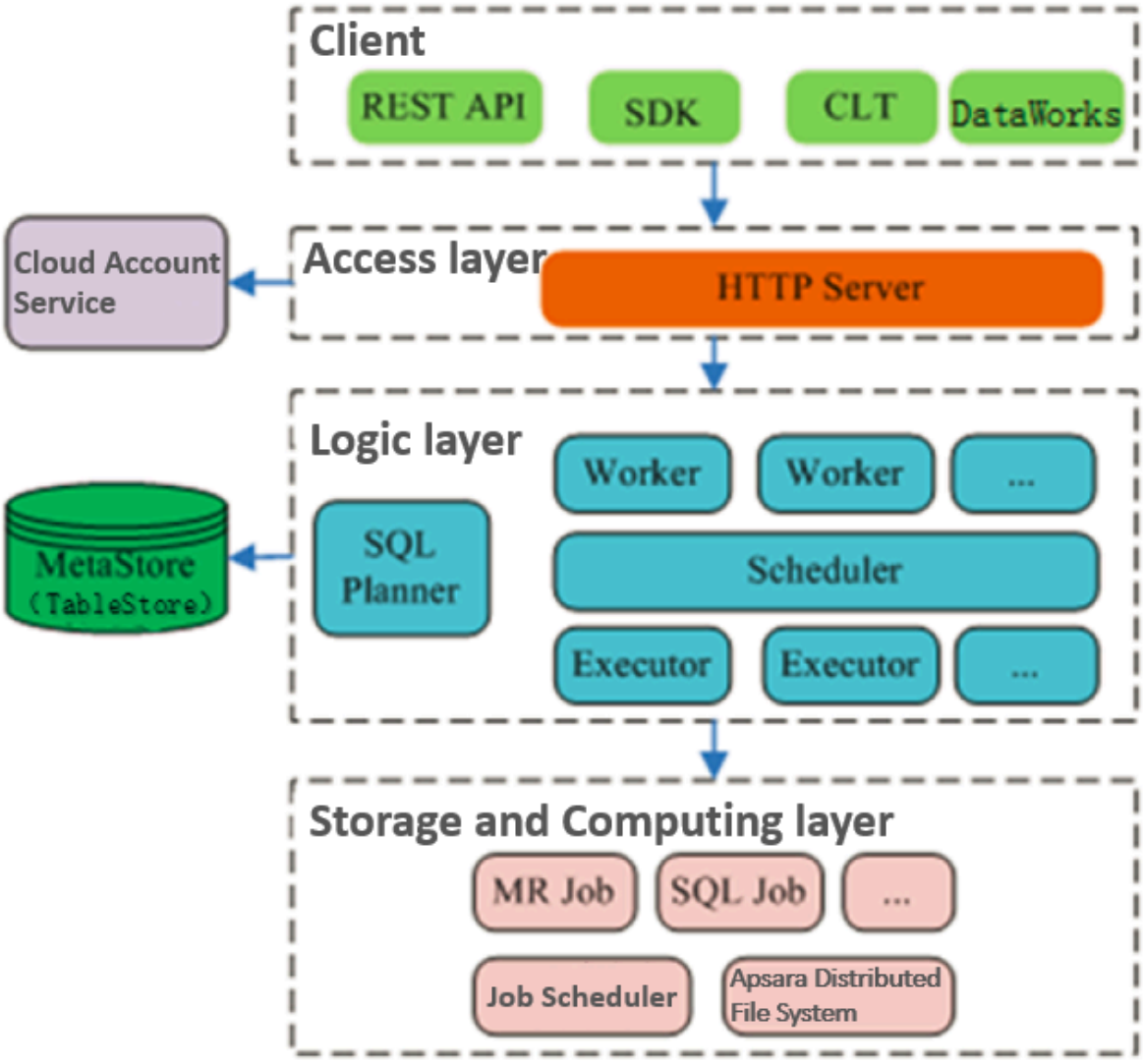
Resource name	Domain name	Description
odps_frontend	odps_frontend_server_inner_dns	The internal domain name of the MaxCompute front-end server. This domain name is not subject to VPC.
	odps_frontend_server_public_dns	The private domain name of the MaxCompute front-end server.
	odps_frontend_server_internet_dns	The public domain name of the MaxCompute front-end server.
tunnel_frontend	odps_tunnel_frontend_server_inner_vip	The internal domain name of the front-end server for MaxCompute Tunnel. This domain name is not subject to VPC.
	odps_tunnel_frontend_server_public_vip	The private domain name of the front-end server for MaxCompute Tunnel.

Resource name	Domain name	Description
	odps_tunnel_frontend_server_internet_vip	The public domain name of the front-end server for MaxCompute Tunnel.
cupid_web_proxy	odps_jobview_dns	The internal domain name of the MaxCompute Jobview. This domain name is not subject to VPC.
logview	odps_logview_inner_dns	The internal domain name of the MaxCompute Logview. This domain name is not subject to VPC.
	odps_logview_public_dns	The private domain name of the MaxCompute Logview.
web_console	odps_webconsole_inner_dns	The internal domain name of the MaxCompute Web console. This domain name is not subject to VPC.
	odps_webconsole_public_dns	The private domain name of the MaxCompute Web console.

14.2 Architecture

Figure 14-5: MaxCompute architecture **shows the MaxCompute architecture.**

Figure 14-5: MaxCompute architecture



The MaxCompute service is divided into four parts: client, access layer, logic layer, and storage and computing layer. Each layer can be scaled out.

The following methods can be used to implement the functions of a MaxCompute client:

- **API: RESTful APIs are used to provide offline data processing services.**
- **SDK: RESTful APIs are encapsulated in SDKs. SDKs are currently available in programming languages such as Java.**

- **Command line tool (CLT): This client-side tool runs on Windows and Linux. CLT allows you to submit commands to manage projects and use DDL and DML.**
- **DataWorks: DataWorks provides upper-layer visual ETL and BI tools that allow you to synchronize data, schedule tasks, and create reports.**

The access layer of MaxCompute supports HTTP, HTTPS, load balancing, user authentication, and service-level access control.

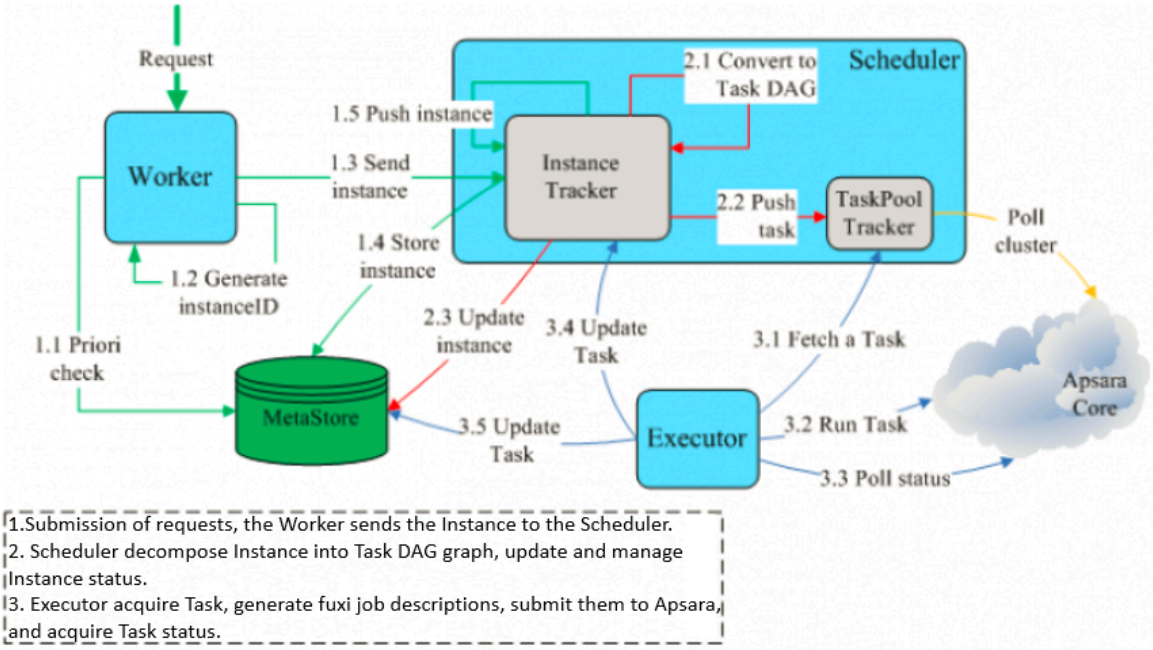
The logic layer is at the core of MaxCompute. It supports project and object management, command parsing and execution logic, and data object access control and authorization. The logic layer is divided into control and compute clusters. The control cluster manages projects and objects, parses queries and commands, and authorizes access to data objects. The compute cluster executes tasks. Both control and compute clusters can be scaled out as required. The control cluster is comprised of three different roles: Worker, Scheduler, and Executor. These roles are described as follows:

- **The Worker role processes all RESTful requests and manages projects, resources, and jobs. Workers forward jobs that need to launch Fuxi tasks (such as SQL, MapReduce, and Graph jobs) to the Scheduler for further processing.**
- **The Scheduler role schedules instances, splits instances into multiple tasks, sorts tasks that are pending for submission, and queries resource usage from FuxiMaster in the compute cluster for throttling. If there are no idle slots in Job Scheduler, the Scheduler stops processing task requests from Executors.**
- **The Executor role is responsible for launching SQL and MapReduce tasks. Executors submit Fuxi tasks to FuxiMaster in the compute cluster and monitor the operating status of these tasks.**

In summary, when you submit a job request, the Web server at the access layer queries the IP addresses of registered Workers and sends API requests to randomly selected Workers. The Workers then send these requests to the Scheduler for scheduling and throttling. Executors actively poll the Scheduler queue. If the necessary resources are available, the Executors start executing tasks and

return the task execution status to the Scheduler. The following figure shows the corresponding business execution logic.

Figure 14-6: Business execution logic



The storage and computing layer of MaxCompute is a core component of the proprietary cloud computing platform developed by Alibaba Cloud. As the kernel of the Apsara system, this layer runs in the compute cluster independent of the control cluster. The preceding MaxCompute architecture diagram illustrates only major modules of the Apsara kernel, such as Apsara Distributed File System and Job Scheduler.

Among the modules, Apsara Distributed File System is designed to aggregate the storage resources of a large number of machines and provide users with reliable large-scale distributed storage services. Apsara Distributed File System is an important part of the Apsara kernel.

Apsara Distributed File System includes three masters and multiple chunkservers. A master is responsible for the storage and management of file metadata, while a chunkserver is responsible for data storage. Identical blocks of data are stored on multiple chunkservers to ensure its reliability. In normal cases, data is stored in Apsara Distributed File System in three copies. All MaxCompute data files are stored in Apsara Distributed File System. The data files can be found in the /

product/aliyun/odps directory. It is important to note that masters operate in hot standby mode. Only one master operates at a time.

- master:
  1. PanguMaster maintains metadata for the entire file system, including namespaces, file-to-block mappings, and data block storage addresses.
  2. PanguMaster is the heart of the distributed file storage system and controls system-level activities such as garbage collection for isolated data blocks, data merging among chunkservers, chunkserver health check, and recovery of data blocks lost due to a down chunkserver.
  3. PanguMaster also manages data access requests originating from multiple clients at the same time to ensure the integrity of data in the cluster.
  4. PanguMaster only allows the client to perform operations on metadata. Data transmissions are conducted directly among chunkservers.
- chunkserver: The files in Apsara Distributed File System are divided into fixed-size storage units called chunks. A machine that stores chunks is called a chunkserver. PanguMaster assigns a 128-bit ID to a chunk when the chunk is created. The Apsara Distributed File System client reads the chunks stored in disks based on chunk IDs.

To provide better support for the processing of structured data in MaxCompute tables, the MaxCompute team has implemented a special Apsara Distributed File System file format called CFile.

- CFile is a file format based on column storage. It is designed to reduce invalid disk read operations during offline data processing. Data in the file is clustered by column into blocks and compressed to reduce storage space. This means that in offline data processing scenarios, you only need to read the required data. This avoids unnecessary disk operations, improves disk read efficiency, and reduces network bandwidth consumption.
- The CFile storage structure can be logically divided into three areas: data area, index area, and header area. The data area stores the user data that is clustered by column and uses blocks as organizational units. The index area stores the indexes corresponding to the data blocks of each column, which includes the starting position of each block in the file, the length of a compressed block, and the data amount in a block (for variable-length data types such as string). The header area stores the metadata of each column in the file, such as the

starting position of the column index, index length, column type information, compression method, and row count and version of user data.

- MaxCompute supports the following data types:
  - Bigint: represents an 8-byte signed integer.
  - Boolean: represents a logical true or false value.
  - Double: represents an 8-byte double-precision floating-point number.
  - String: represents a string in UTF-8 format. MaxCompute functions automatically assume that string objects contain UTF-8 encoded strings. If the string is encoded in other formats, an error occurs.
  - Datetime: represents a date and time in the `YYYY-MM-DD HH:mm:SS` format. Example: 2012-01-02 10:09:25

Job Scheduler is a module for resource management and task scheduling in the Apsara kernel. It also provides a basic programming framework for application development. It is designed to make full use of the hardware resources of the entire cluster to meet the computing requirements of users and systems. Job Scheduler supports the processing of two application types: a low-latency online service called FuxiService and a high-throughput offline processing application called FuxiJob. Job Scheduler is similar to YARN in Hadoop.

- FuxiService: a resident process in Job Scheduler. You can send requests to create and destroy a service. Job Scheduler does not proactively destroy a service process.
- FuxiJob: a temporary task in Job Scheduler. When a task ends, the resources are released and reclaimed by Job Scheduler.

Job Scheduler schedules and allocates cluster storage and computing resources to upper-layer applications. Job Scheduler is able to manage computing resource quotas, access control policies, and job priorities to ensure that resources are shared effectively. Job Scheduler provides a data-driven, multi-level parallel computing framework, which is similar to the MapReduce programming model . The framework is ideal for complex applications such as large-scale data processing and large-scale computing.

Job Scheduler has two masters and multiple Tubos. Masters operate in cold standby mode. Only one master operates at a time. A Tubo process is started on each compute node to manage available resources of each machine, such as the CPU,

memory, hard disk, and network, and record the resources used on each machine. The Tubo process on each machine reports the resource usage to FuxiMaster, which centrally manages and schedules the resources.

14.3 Features

14.3.1 Tunnel

14.3.1.1 Overview

Data upload and download tools provided by MaxCompute are compiled based on the Tunnel SDK. This topic describes the major APIs of the Tunnel SDK.

The usage of the SDK varies according to the version. For specific information, see *SDK Java Doc*.

Table 14-32: Major APIs

API	Description
TableTunnel	An entry class of the MaxCompute Tunnel service.
TableTunnel.UploadSession	A session that uploads data to a MaxCompute table.
TableTunnel.DownloadSession	A session that downloads data from a MaxCompute table.
InstanceTunnel	An entry class of the MaxCompute Tunnel service.
InstanceTunnel.DownloadSession	A session that downloads data from a MaxCompute instance. This session applies only to SQL instances that start with the SELECT keyword and are used to query data.

Note:
The tunnel endpoint supports automatic routing based on the MaxCompute endpoint settings.

14.3.1.2 TableTunnel

This topic describes the TableTunnel API.

Definition

Definition:

```
public class TableTunnel {
```

```
public DownloadSession createDownloadSession(String projectName,
String tableName);
public DownloadSession createDownloadSession(String projectName,
String tableName, PartitionSpec partitionSpe c);
public UploadSession createUploadSession(String projectName, String
tableName);
public UploadSession createUploadSession(String projectName, String
tableName, PartitionSpec partitionSpec);
public DownloadSession getDownloadSession(String projectName, String
tableName, PartitionSpec partitionSpec, String id);
public DownloadSession getDownloadSession(String projectName, String
tableName, String id);
public UploadSession getUploadSession(String projectName, String
tableName, PartitionSpec partitionSpec, String id);
public UploadSession getUploadSession(String projectName, String
tableName, String id); public void setEndpoint(String endpoint);
}
```

Description:

- **Lifecycle: the duration from the creation of the TableTunnel instance to the end of the program.**
- **TableTunnel provides a method to create UploadSession and DownloadSession objects.** `TableTunnel.UploadSession` **is used to upload data, and** `TableTunnel.DownloadSession` **is used to download data.**
- **A session refers to the process of uploading or downloading a table or partition. A session consists of one or more HTTP requests to Tunnel RESTful APIs.**
- **Upload sessions of TableTunnel use the** `INSERT INTO` **semantics. Multiple upload sessions of the same table or partition does not affect each other, and the data uploaded in each session is stored in an independent directory.**
- **In an upload session, each RecordWriter is matched with an HTTP request and is identified by a unique block ID. The block ID is the name of the file correspond ing to the RecordWriter.**
- **If you use the same block ID to enable a RecordWriter multiple times in the same session, the data uploaded by the RecordWriter that calls the** `close()` **function last will overwrite all previous data. This feature can be used to retransmit data of a block when data upload fails.**

API implementation process

1. **The** `RecordWriter.write()` **function uploads your data as files to a temporary directory.**
2. **The** `RecordWriter.close()` **function moves the files from the temporary directory to the Data directory.**

3. **The `session.commit()` function moves each file in the Data directory to the directory where the corresponding table is located and updates the table metadata. This way, data moved into a table by the current task will be visible to the other MaxCompute tasks such as SQL and MapReduce.**

API limits

- **The value of a block ID must be greater than or equal to 0 and less than 20000. The size of data to be uploaded in a block cannot exceed 100 GB.**
- **A session is uniquely identified by its session ID. The lifecycle of a session is 24 hours. If your session times out due to the transfer of large volumes of data, you must transfer your data in multiple sessions.**
- **The lifecycle of an HTTP request corresponding to a RecordWriter is 120 seconds. If no data flows over an HTTP connection within 120 seconds, the server closes the connection.**

   **Note:**
  HTTP has an 8 KB buffer. When you call the `RecordWriter.write()` function, your data may be saved to the buffer and no inbound traffic flows over the corresponding HTTP connection. In this case, you can call the `TunnelRecordWriter.flush()` function to forcibly flush data from the buffer.

- **When you use a RecordWriter to write logs to MaxCompute, the RecordWriter may time out due to unexpected traffic fluctuations. Therefore, we recommend that you:**
  - **Do not use a RecordWriter for each data record. Otherwise, a large number of small files are generated, because each RecordWriter corresponds to a file. This affects the performance of MaxCompute.**
- **Do not use a RecordWriter to write data until the size of cached code reaches 64 MB.**
- **The lifecycle of a RecordReader is 300 seconds.**

14.3.1.3 InstanceTunnel

This topic describes the InstanceTunnel API.

Definition:

```
public class InstanceTunnel{
 public DownloadSession createDownloadSession(String projectName,
String instanceID);
```

```
 public DownloadSession createDownloadSession(String projectName,
String instanceID, boolean limitEnabled);
 public DownloadSession getDownloadSession(String projectName, String
id);
 }
```

Parameter description:

- **projectName: the name of a project.**
- **instanceID: the ID of an instance.**

Limits: Although InstanceTunnel provides an easy way to obtain instance execution results, it is subject to the following permission limits to ensure data security:

- **If the number of records does not exceed 10,000, all users who have the read permission on the specified instance can use InstanceTunnel to download the data. This is also applicable to the scenario of calling a Restful API to query data.**
- **If the number of records exceeds 10,000, only users who have the permission to read all the source tables from which the specified instance queries data can use InstanceTunnel to download the data.**

14.3.1.4 UploadSession

This topic describes the UploadSession API.

API definition:

```
public class UploadSession {
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec) throws TunnelException;
UploadSession(Configuration conf, String projectName, String tableName
, String partitionSpec, String uploadId) throws TunnelException;
public void commit(Long[] blocks); public Long[] getBlockList();
public String getId();
public TableSchema getSchema();
public UploadSession.Status getStatus(); public Record newRecord();
public RecordWriter openRecordWriter(long blockId);
public RecordWriter openRecordWriter(long blockId, boolean compress);
}
```

UploadSession API description.

Table 14-33: UploadSession API

Item	Description
Lifecycle	**From the upload instance creation to the end of the uploading.**

Item	Description
Purpose	Creates an upload instance by calling a constructor method or by using the TableTunnel class. · Request mode: synchronous. · The server creates an upload session and generates a unique upload ID. You can get the upload ID by running getId on the console.
Upload data	· Request mode: asynchronous. · Call openRecordWriter to generate a RecordWriter instance. The blockId parameter identifies the data to upload this time and the position of the data in the table. The value range is [0, 20000]. In case the uploading fails, the data is re-uploaded based on the block ID.
Check uploading	· Request mode: synchronous. · Call getStatus to get the uploading status. · Call getBlockList to get a list of the block IDs of successful uploading instances, check the block ID list, and re-upload data for failed uploading instances.
Stop uploading	· Request mode: synchronous. · Call commit(Long[] blocks). The blocks parameter indicates the list of block IDs of successful uploading instances. The server will verify the block ID list. · The verification improves data correctness. If the provided block list is different from the block list on the server, an error is reported.
Status	· UNKNOWN: Initial value set while server just creates a session. · NORMAL: An UPLOAD object is created successfully. · CLOSING: The server sets the upload session to CLOSING status before calling the COMPLETE method (to complete uploading). · CLOSED: The uploading is completed (data has been moved to the directory where the result table is). · EXPIRED: The upload session is timed out. · CRITICAL: An error occurs.

 **Notice:**

- **blockId in the same UploadSession API must be unique. That is, after a block ID is used to start RecordWriter in an upload session, data is written, and the session is closed and committed, this block ID cannot be used to start another RecordWriter.**
- **The maximum size of a block is 100 GB. We strongly recommend that 64 MB or more data is written into each block. Otherwise, the computing performance will seriously degrade.**
- **Each session has a 24-hour life cycle on the server.**
- **You are advised to have data prepared before calling openRecordWriter. A network action is triggered every time the Writer writes 8 KB data. If no network action is triggered in the last 120 seconds, the server closes the connection and the Writer becomes unavailable. You have to start a new Writer.**

14.3.1.5 DownloadSession

This topic describes the DownloadSession class.

API definition:

```
public class DownloadSession {
DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec) throws TunnelException
DownloadSession(Configuration conf, String projectName, String
tableName, String partitionSpec, String downloadId) throws TunnelExce
ption
public String getId()
public long getRecordCount() public TableSchema getSchema()
public DownloadSession.Status getStatus()
public RecordReader openRecordReader(long start, long count)
public RecordReader openRecordReader(long start, long count, boolean
compress)
}
```

DownloadSession API description.

Table 14-34: DownloadSession API

Parameter	Description
Lifecycle	**From the creation of the Download instance to the end of the download process.**

Parameter	Description
Purpose	Creates a Download instance by calling a constructor method or using TableTunnel. · Request mode: Synchronous. · The server creates a session for this Download and generates a unique download ID to mark the Download. The console can get data with a get ID. The operation has a high overhead. The server creates indexes for the data files. If many data files exist, the operation takes a long time. Then the server returns the total number of records, and starts concurrent downloads according to the number of records.
Download data	· Request mode: Asynchronous. · Call openRecordReader to generate a RecordReader instance. The Start parameter marks the start position of record for this download. The value of Start is equivalent to or greater than 0. The Count parameter marks the number of records for this download. The value of Count is greater than 0.
View the download process	· Request mode: Synchronous. · Call getStatus to get the download status.
Status	· UNKNOWN: the initial value that is set when the server creates a session. · NORMAL: The download object is successfully created. · CLOSED: The download session is completed. · EXPIRED: The download session times out.

14.3.1.6 TunnelBufferedWriter

This topic describes the TunnelBufferedWriter interface.

The upload process is complex due to limits on block management and connection timeout on the server. The Tunnel SDK provides an enhanced RecordWriter, TunnelBufferWriter, to simplify the upload process.

The TunnelBufferedWriter interface is defined as follows:

```
public class TunnelBufferedWriter implements RecordWriter {
    public TunnelBufferedWriter(TableTunnel.UploadSession session,
CompressOption option) throws IOException;
    public long getTotalBytes();
    public void setBufferSize(long bufferSize);
    public void setRetryStrategy(RetryStrategy strategy);
    public void write(Record r) throws IOException;
```

```
    public void close() throws IOException;
}
```

A TunnelBufferedWriter object is described as follows:

- **Lifecycle: the duration from the time RecordWriter is created to the time the data upload ends.**
- **TunnelBufferedWriter instance: You can call the openBufferedWriter interface of UploadSession to create a TunnelBufferedWriter instance**
- **Data upload: When you call the Write interface, data is first written to the local cache. After the cache is full, the data is submitted to the server in batches to avoid connection timeout. In addition, if the upload fails, the system automatically retries the upload operation.**
- **End upload: Call the Close interface and then call the Commit interface of UploadSession to end the upload process.**
- **Buffer control: You can use the setBufferSize interface to modify the memory occupied by the buffer (in bytes), preferably 64 MB or more to prevent the server from generating too many small files, which may affect performance. The valid range is 1 MB to 1000 MB. The default value is 64 MB, which is recommended in most cases.**
- **Retry policy settings: You have three retry avoidance policies to choose from: EXPONENTIAL_BACKOFF, LINEAR_BACKOFF, and CONSTANT_BACKOFF. For example, the following code segment sets the Write retry count to 6. To avoid unnecessary retries, each retry is performed only after exponentially ascending intervals of 4s, 8s, 16s, 32s, 64s, and 128s by default.**

```
RetryStrategy retry
  = new RetryStrategy(6, 4, RetryStrategy.BackoffStrategy.EXPONENTIA
L_BACKOFF)
writer = (TunnelBufferedWriter) uploadSession.openBufferedWriter();
writer.setRetryStrategy(retry);
```

Note:

We recommend that you do not adjust the preceding settings.

14.3.2 SQL

MaxCompute SQL is a structured query language whose syntax is similar to Oracle , MySQL, and Hive SQL. MaxCompute SQL can be regarded as a subset of standard SQL. However, MaxCompute SQL is not equivalent to a database, because it does

not possess many characteristics that a database has, such as transactions, primary key constraints, and indexes.

MaxCompute SQL is applicable to scenarios that have large amounts of data (measured in TBs) and that do not have high real-time processing requiremen ts. It takes a relatively long time to prepare and submit each job. Therefore, MaxCompute SQL is not optimal for services that need to process thousands of transactions per second.

14.3.3 MapReduce

MapReduce is a programming model equivalent to Hadoop MapReduce. This model is used for parallel MaxCompute operations on TB-level large-scale datasets.

You can use the MapReduce Java API to write MapReduce programs to process MaxCompute data. The Map and Reduce concepts are borrowed from functional and vector programming languages. This helps programmers run their programs on distributed systems without having to perform distributed parallel programmin g.

MapReduce works only after you specify a Map function and a concurrent Reduce function. The Map function maps a group of key-value pairs to another group of key-value pairs. The Reduce function ensures that all elements in the mapped key-value pairs share the same key group.

MaxCompute MapReduce has the following characteristics:

- Provides Hadoop-style MapReduce functions designed for MaxCompute (used to process tables and volumes).
- Supports the input and output of only built-in data types of MaxCompute.
- Supports the input and output of multiple tables to different partitions.
- Capable of reading resources.
- Does not allow you to use views as data inputs.
- Provides a limited sandbox security environment.

The following procedure shows how MapReduce processes data:

1. Before you perform Map operations, ensure that `partition` is set for the input data. The input data is divided into equally sized blocks called partitions. Each partition is processed as the input of a single Map worker so that multiple Map workers can work in parallel.

2. After partitioning, multiple Map workers start processing the data in parallel. Each Map worker reads its respective partition data, computes the data, and exports the result to Reduce.

   Note:
   When a Map worker generates data, it must specify a key for each output record. The key determines the Reduce worker for which the data entry is targeted. Multiple keys may correspond to a single Reduce worker. Data entries with the same key are sent to the same Reduce worker. A single Reduce worker may receive data entries for multiple keys.

3. Before entering the Reduce stage, the MapReduce framework sorts data based on Key values to make data entries with the same Key value adjacent. If you specify `Combiner`, the framework will call Combiner to combine data entries that share the same Key value.

   Note:
   You can customize the Combiner logic. Unlike the typical MapReduce framework protocol, MaxCompute requires the input and output parameters of Combiner to be consistent with those of Reduce. This process is generally called `Shuffle`.

4. When entering the Reduce stage, data entries with the same Key value will be in the same Reduce worker. A single Reduce worker may receive data from multiple Map workers. Each Reduce worker performs the Reduce operation on multiple data entries with the same Key value. After the Reduce operation, all data of the same key is converted into a single value.

Note:
This topic only provides a brief introduction to MapReduce. For more information, see related documentation.

14.3.4 Graph

Graph is the computing framework of MaxCompute designed for iterative graph processing. It provides programming interfaces similar to Pregel, allowing you to develop efficient machine learning and data mining algorithms.

Large amounts of data on the Internet is structured as graphs, such as social networking and logistics information. Graph computing models are iterative computing models. Throughout the entire computing process, multiple iterations are performed to achieve convergence. For example, for machine learning algorithms that require iterative learning model parameters, Graph is more suited than MapReduce. In common usage scenarios, you can abstract a question as a graph. Then, you can set the vertex as the center of the graph, and use supersteps for iterative updating.

MaxCompute Graph currently works in two modes:

- Offline mode: suitable for large-scale computing. Similar to MapReduce jobs, this mode involves loading and computing.
- Interactive mode: suitable for small-scale computing. You can implement a UDF and then use the command line for interaction.

In offline mode, loading and computing are independent processes. Loaded data resides in the memory. You can apply different computing logics to the loaded data. For example, the risk control department may load a set of data once a day. The operations personnel will apply different query logics to the data to view the relationships between the data.

MaxCompute Graph has been applied to many businesses in Alibaba. For example, weighted PageRank algorithms are used to compute influence metrics for Alipay users, and variational Bayesian EM models are used to predict users' car brands based on the properties of the items purchased by users.

14.3.5 Unstructured data processing (integrated computing scenarios)

Alibaba Cloud introduced the MaxCompute-based unstructured data processing framework so that MaxCompute SQL commands can directly process external user data, such as unstructured data from OSS. You are no longer required to first import data into MaxCompute tables.

You can run a simple DDL statement to create an external table in MaxCompute, and associate MaxCompute tables with external data sources. This table can then act as an interface between MaxCompute and external data sources. The external table can be accessed in the same way as a MaxCompute table, and computed by MaxCompute SQL.

MaxCompute allows you to process the following data sources by creating external tables:

- **Internal data sources: OSS, Table Store, AnalyticDB, ApsaraDB for RDS, HDFS (Alibaba Cloud), and TDDL.**
- **External data sources: HDFS (open source), ApsaraDB for MongoDB, and Hbase.**

14.3.6 Unstructured data processing in MaxCompute

MaxCompute has the following problems when processing unstructured data: MaxCompute stores data as volumes and must export generated unstructured data to an external system for processing.

To alleviate these problems, MaxCompute uses external tables to enable connection s between MaxCompute and various data types. MaxCompute uses external tables to read and write data volumes as well as process unstructured data from external sources such as OSS.

14.3.7 Enhanced features

14.3.7.1 Spark on MaxCompute

14.3.7.1.1 Open-source platform - Cupid

14.3.7.1.1.1 Overview

MaxCompute is a big data solution independently developed by Alibaba Cloud that leads the industry in scale and stability. The big data open-source communities are actively developing big data solutions. All kinds of systems are rapidly emerging and growing to meet various requirements. To better serve users and to diversify the MaxCompute ecosystem, the MaxCompute team has developed the Cupid platform to connect MaxCompute with open-source communities. The Cupid platform integrates the diversity of open-source communities with the scale and stability of the Apsara system.

The software stacks of open-source communities and the Apsara system are similar with slight differences.

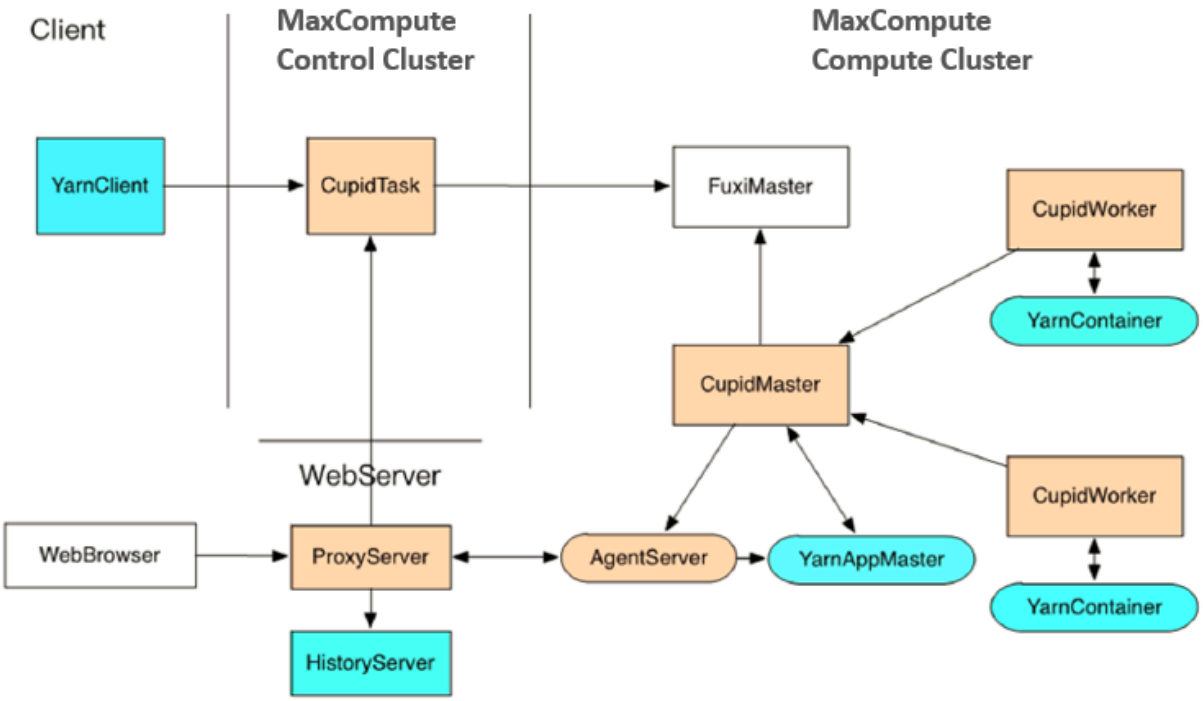
Most open-source communities use HDFS as a distributed file system, while the Apsara system uses Apsara Distributed File System. Most open-source communitie s use YARN as a distributed scheduling system, while the Apsara system uses Job Scheduler. On top of Job Scheduler are the computing engines designed for all

kinds of scenarios. Cupid provides compatibility with open-source communities for open-source applications (such as Spark) to run on MaxCompute.

14.3.7.1.1.2 Compatibility with YARN

YARN has three application-oriented APIs: YarnClient, AMRMClient, and NMClient. YarnClient is used to submit applications to a cluster. AMRMClient is used by AppMaster to send messages to Resource Manager to request and release resources. NMClient is used to start and stop application containers.

Figure 14-7: YARN on MaxCompute



The preceding figure shows the process of submitting a YARN application to be run on MaxCompute. The yellow boxes indicate Cupid components, while the light blue boxes indicate open-source components. The procedure is as follows:

1. Use a Spark client that encapsulates the YarnClient class to access the MaxCompute control cluster and submit a job to FuxiMaster.
2. FuxiMaster starts a CupidMaster. Then, the CupidMaster starts YarnAppMaster based on the YARN protocol.
3. YarnAppMaster interacts with FuxiMaster through CupidMaster to request and release resources.

4. To start a new container, you must first use Tubo in Job Scheduler to start a CupidWorker. The CupidWorker will then start the container based on the YARN protocol.

Note:

Typically, YarnAppMaster provides a UI. The UI is implemented through Cupid based on a proxy mechanism.

14.3.7.1.1.3 Compatibility with FileSystem

Most open-source communities use HDFS as a distributed storage solution. The FileSystem API provided by Hadoop is compatible with Alibaba Cloud OSS and Amazon S3. Apsara Distributed File System is compatible with FileSystem API. Open-source jobs submitted to MaxCompute can be run natively on Apsara Distributed File System.

Note:

Apsara Distributed File System does not directly provide external services. The data in Apsara Distributed File System can only be used as intermediate job data, such as checkpoints. You can use OSS to make the data stored in Apsara Distributed File System accessible to other environments.

14.3.7.1.1.4 DiskDrive

Most open-source applications use local file systems for data processing, such as the shuffle and storage modules in Spark. In environments with large clusters, disks are important system resources. Disks must be centrally managed to ensure high availability, performance, and security. In the Apsara system, disks are centrally managed by Apsara Distributed File System. To provide local file system APIs based on Apsara Distributed File System, the Cupid team has designed and implemented the DiskDriverService system by integrating Web-based storage into MaxCompute.

14.3.7.1.2 Feature extensions

14.3.7.1.2.1 Overview

MaxCompute provides the Cupid framework to support open-source applications. This allows Spark to be run on MaxCompute. For ease of use and better integration with MaxCompute, there are several extensions available for Spark on MaxCompute to add features such as the secure isolation of open-source Spark applicatio

ns, mutual access between MaxCompute data and Spark data, and support for interactions in multi-tenant clusters.

The following sections describe these extensions.

14.3.7.1.2.2 Security isolation

Spark jobs submitted to the MaxCompute computing cluster are run in sandboxes , preventing attacks on the system. A parent-child process architecture is used for the entire system. The Cupid framework runs in the parent process, and Spark runs in the child processes. When Spark requires access to system services, the parent process accesses the services on behalf of Spark by communicating with the child processes.

14.3.7.1.2.3 Data interconnection

An advantage of running Spark on MaxCompute is that resources used by Spark and MaxCompute jobs are shared across all clusters. This allows jobs to directly access their data without having to pull data across different clusters.

This data includes metadata and storage data. For security reasons, Spark must be authenticated through the MaxCompute account system before it can store MaxCompute data. Spark on MaxCompute provides OdpsRDD and OdpsDataFrame so that users can use Spark APIs on MaxCompute. Spark SQL has direct access to MaxCompute metadata for SQL optimization and can directly store and retrieve MaxCompute data at the physical layer.

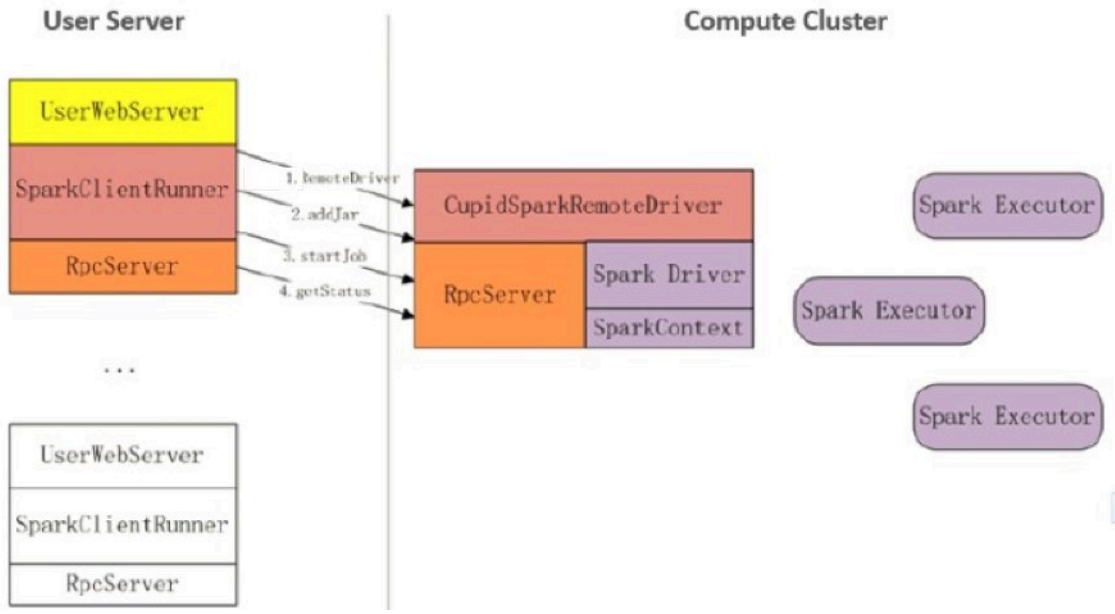
14.3.7.1.2.4 Client mode

The `yarn-cluster` and `yarn-client` modes are commonly used in open-source communities for Spark-related development efforts. In `yarn-cluster` mode, you can submit a Spark job to a YARN cluster. After the job is run, the client generates a log that indicates the job status. In this mode, you cannot submit a job to a Spark session multiple times in real time, and the client cannot obtain the running status and result of each job. The `yarn-client` mode takes effect for interactive scenarios. However, to use the yarn-client mode, you need to launch the Spark driver process from the client side. You cannot use a Spark session as a service in this mode. The MaxCompute team has developed the `Client` mode based on Spark on MaxCompute to solve the preceding problems. The Client mode has the following features:

1. **The client is a lightweight process that does not require you to launch the Spark driver process.**
2. **The client provides a set of APIs that can be used to submit jobs in real time to the same Spark session in MaxCompute clusters. It can also monitor the statuses of all jobs in the Spark session.**
3. **The client can build dependencies between jobs by monitoring job statuses and results.**
4. **You can compile an application JAR package in real time and submit it to the original Spark session through the client.**
5. **The client can be integrated into the Web servers of a service, and can also be scaled horizontally.**

In `Client` mode, you need to use CupidSparkClientRunner to start a Spark session in a MaxCompute cluster. Then, you can use CupidSparkClientRunner to perform operations on the client side, such as submitting jobs and viewing the running statuses and results of the jobs. Cached data can be shared between jobs. You can also construct multiple CupidSparkClientRunner objects to interact with the same Spark session. The following figure shows the block diagram of the Spark Client mode.

Figure 14-8: Spark Client mode



The procedure for using the Spark Client mode is as follows:

1. **You submit a job to a MaxCompute cluster to launch CupidSparkRemoteDriver and obtain the SparkClientRunner object.**
2. **You use SparkClientRunner to add the JAR package that will execute the job to RemoteDrive.**
3. **SparkClientRunner uses the job classname to submit the job to RemoteDriver. RemoteDriver then runs the job.**
4. **SparkClientRunner monitors the job status based on the job ID returned after the job is submitted.**

14.3.7.1.2.5 Spark ecosystem support

The Spark ecosystem covers diverse components, including MLlib, Streaming, PySpark, SparkR, GraphX, and SQL. Spark on MaxCompute provides a complete Spark ecosystem that supports the scaling of original resources in open-source communities. The ecosystem provides consistent user experience with that of open-source communities. Spark on MaxCompute also supports access to the Spark UI and historical log files.

14.3.7.2 Elasticsearch on MaxCompute

14.3.7.2.1 Terms

term

An exact value that can be indexed. You can use a term query to search for an exact match.

text

A piece of unstructured data. Typically, a text is parsed into individual terms that are stored in an Elasticsearch index library.

cluster

A collection of one or more nodes that provide external indexing and search services. Elasticsearch is deployed in the Apsara cluster of MaxCompute. Elasticsea rch clusters are a part of the Apsara cluster.

node

A logical service in an Elasticsearch cluster. A node can store data and participate in the cluster's indexing and search capabilities.

shard

A single Lucene instance which is a relatively low-level feature managed by Elasticsearch. An Elasticsearch cluster automatically manages all the shards in a cluster. When a node fails, Elasticsearch moves the shards to a different node or adds a new node.

replica

A distinct copy in Elasticsearch. Elasticsearch on MaxCompute allows you to have multiple replicas across different nodes to improve system-level availability. We recommend that you set the default number of replicas for this service to 1.

index

A collection of documents that have similar characteristics. For example, you can have an index for customer data, an index for a product catalog, and another index for order data. An index is identified by a name (that must be all lowercase) that is used to refer to the index when you perform indexing, search, update, and delete operations on the documents in the index. You can define as many indexes as you want in a single Elasticsearch cluster.

type

A logical partition of an index. You can define one or more types in an index. Typically, a type is defined as a document that has a common set of fields.

mapping

A process that defines document fields and their types as well as other index-wide settings. A mapping is similar to a schema definition in a relational database. Each index has a mapping. A mapping can either be defined in advance or automatically generated when you store a document for the first time.

document

A JSON-formatted string which is stored in Elasticsearch, similar to a row in a relational database. Each document has a type and an ID. A document is a JSON object which contains zero or more fields, or key-value pairs.

field

A simple value or a nested structure. Fields are similar to columns in relational database tables. Each field has a field type.

14.3.7.2.2 How Elasticsearch on MaxCompute works

14.3.7.2.2.1 Overview

Elasticsearch on MaxCompute is based on the open source Elasticsearch. It can run the Elasticsearch service on MaxCompute clusters.

On the MaxCompute client, you can start and manage your Elasticsearch service as needed and configure the number of nodes, disk space, memory size, and custom settings. The resources consumed by the Elasticsearch service are counted against your MaxCompute quota.

The following sections describe how Elasticsearch on MaxCompute functions work.

14.3.7.2.2.2 How distributed architecture works

Basic principles

An Elasticsearch cluster consists of multiple nodes. MaxCompute ensures high availability by controlling the start and stop of Elasticsearch services and nodes, allocating computing resources, and implementing failover based on a centralized scheduling mechanism.

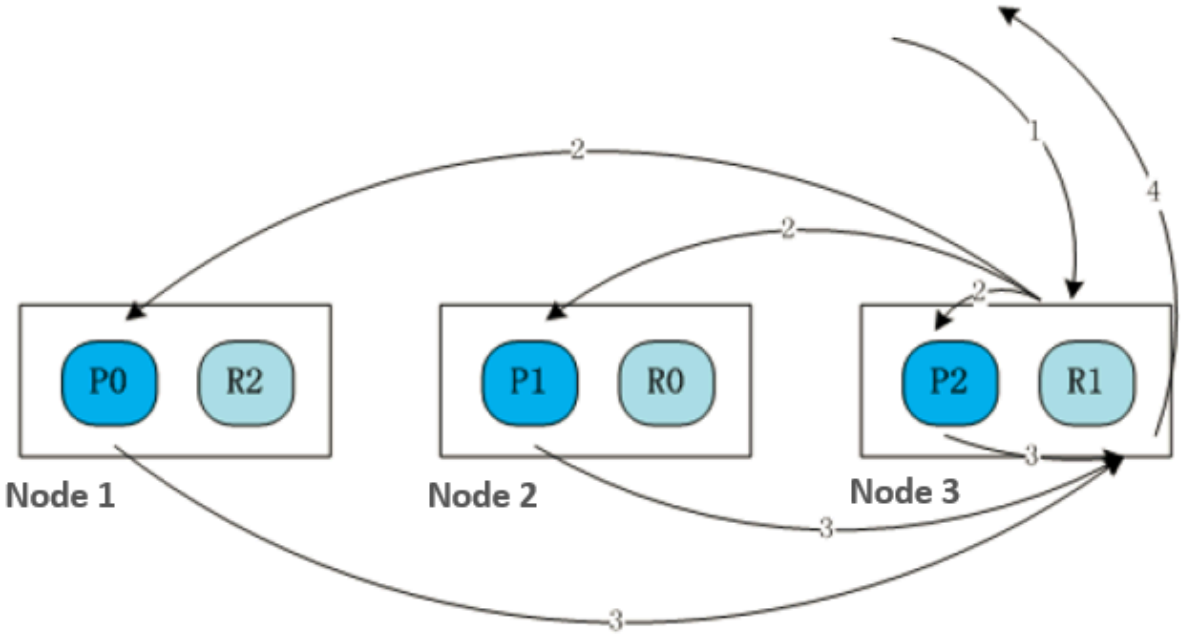
Data is replicated into multiple copies and stored in Apsara Distributed File System . This guarantees that no data is lost due to the failure of a few nodes.

An index is split into multiple shards, which are evenly distributed across multiple nodes in a cluster. The system simultaneously retrieves data shards in multiple nodes, improving retrieval performance.

Implementation process

The following figure shows the distributed retrieval workflow.

Figure 14-9: Distributed retrieval workflow



As shown in the preceding figure, each cluster consists of three nodes. The index has three shards: P0, P1, and P2. These shards are distributed across the three nodes. Each shard is replicated in 1:1 mode, generating three replicas: R0, R1, and R2. The retrieval process is as follows:

1. **A user sends a retrieval request to Node 3.**
2. **After receiving the request, Node 3 sends a retrieval request (2) to P0, P1, and P2 based on the recorded index shard information.**
3. **The nodes where P0, P1, and P2 are located search for the requested information in the specified shards. A retrieval result message (3) is sent to Node 3.**
4. **Node 3 collects the retrieval results from other nodes and returns the retrieval results to the user in an acknowledgment message (4).**

When multiple nodes are performing data retrieval at the same time, the retrieval speed is improved. The performance of distributed retrieval increases with the number of nodes.

14.3.7.2.2.3 How full-text retrieval works

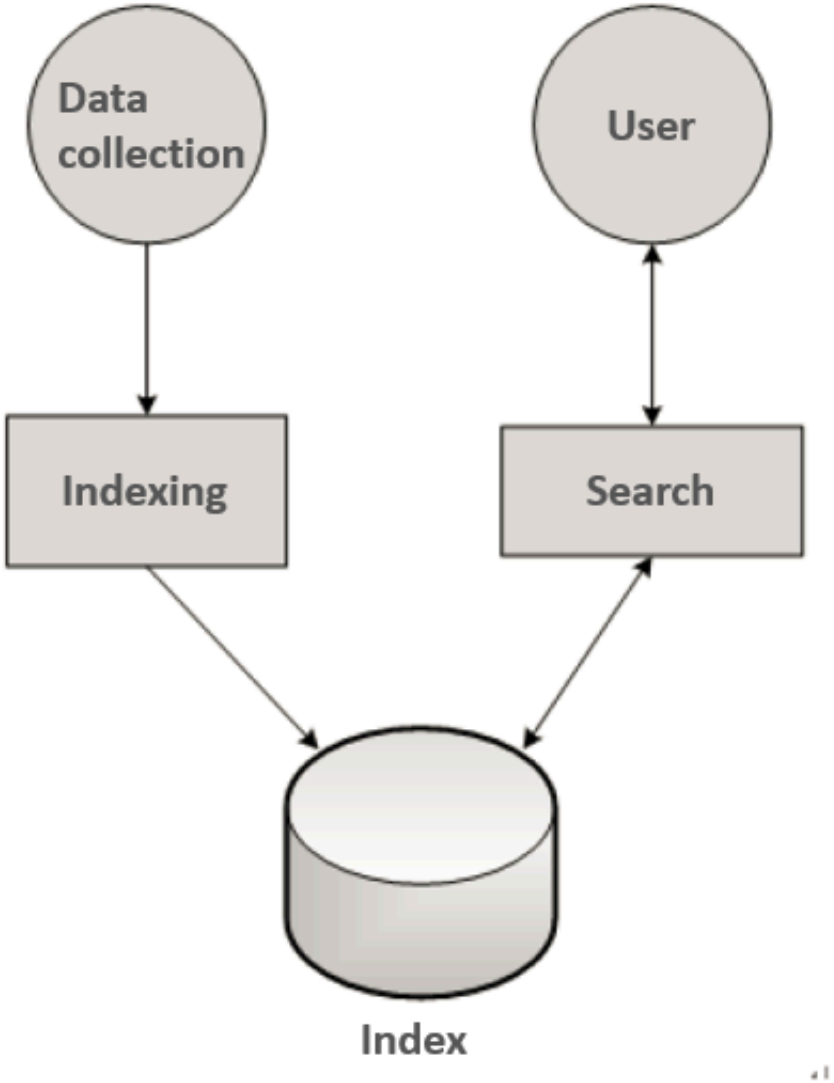
Basic principles

Full-text retrieval refers to techniques used to search for data records containing specified contents from large volumes of texts. In the retrieval process, data in texts is segmented by words, and an inverted index is created based on mappings from words to documents to allow fast document retrieval.

Implementation process

The following figure shows the full-text retrieval process.

Figure 14-10: Full-text retrieval process



The retrieval process is as follows:

1. **The data collection module collects structured and unstructured data, converts the data into the field + value format, and submits the data to the indexing module.**
2. **The indexing module segments the data, creates inverted indexes based on a predefined indexing method, and saves the indexes. The field type, indexing method, and segmentation rules are configured on the retrieval management page.**
3. **The search module receives and processes user requests. Requests are parsed to obtain indexes, fields, and query statements, and then matched to records in the inverted indexes.**
4. **The indexing module returns data that meets user-defined requirements such as sorting rules and request quantity.**

14.3.7.2.2.4 How authentication control works

Basic principles

Authentication control is implemented at the entrance used for external services to check whether users have been authorized to access the index libraries.

Implementation process

The authentication control process is as follows:

1. **Elasticsearch on MaxCompute provides retrieval management and O&M platforms that are only accessible after logon. User account information is verified and authenticated by a centralized authentication module before logon. Any user who fails the authentication is denied access to the platforms.**
2. **The administrator can use the MaxCompute client to add Elasticsearch users and configure permissions for the users.**
3. **The system authenticates all users who attempt to access index libraries. After passing authentication, you will be able to retrieve or perform operations on data in the libraries.**

15 DataWorks

15.1 What is DataWorks?

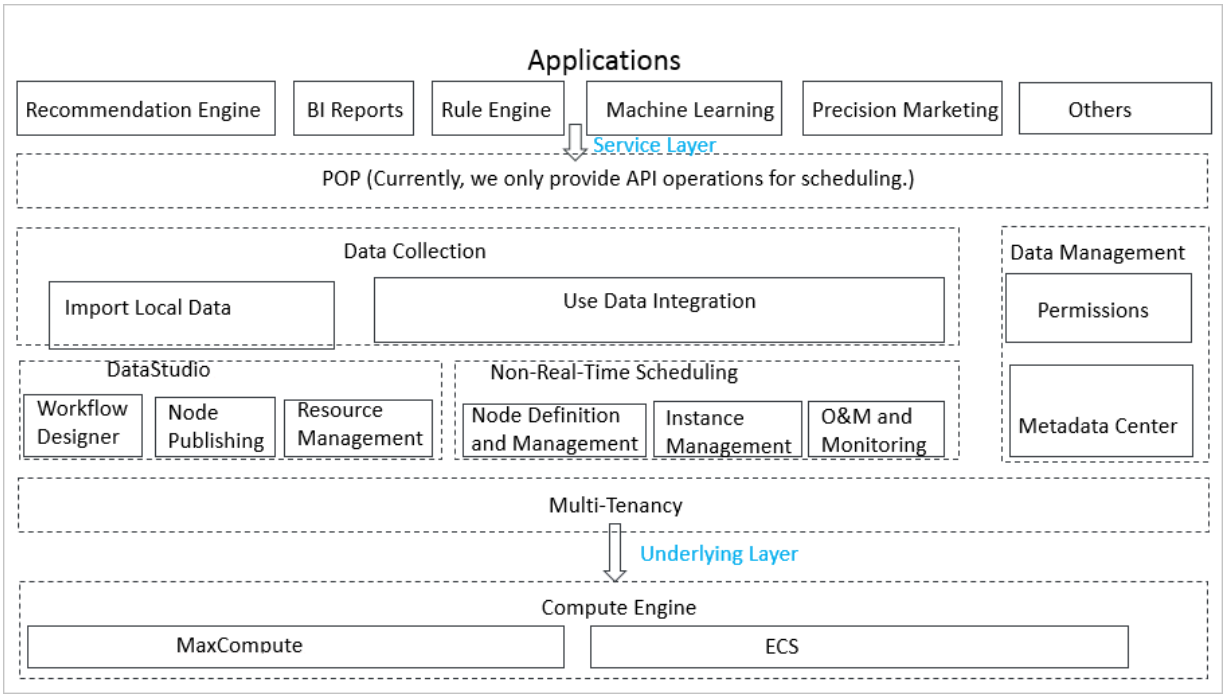
15.1.1 Product overview

DataWorks is a big data platform provided by Alibaba Cloud. It provides end-to-end solutions for enterprises and individual users, and is integrated with the development platform, the management system, and the offline scheduling system.

DataWorks is aimed at mining the full value of the data.

- **It enables large enterprises to build petabyte-level and even exabyte-level data warehouses. The enterprises can improve their business operations through data integration, data asset management, and data analysis.**
- **Small and medium-sized enterprises and individual users can quickly build data-based applications, which drive data based innovations.**

Figure 15-1: Product components



DataWorks consists of an integrated development environment (IDE), a scheduling system, a data integration tool, and a data management system.

- **Integrated development environment (IDE): A development tool that can be used to write SQL, MapReduce (MR), or shell code. The IDE supports collaborative development and version control. By using the visual process design tool, you can quickly define the dependencies among different tasks.**
- **Scheduling system: A system that can schedule millions of tasks in a day. You can manage your tasks online, and view the logs, scheduling status, and monitoring alerts.**
- **Data integration tool: An integration tool that can be used to configure synchronization tasks between heterogeneous data sources. More than 80% of databases and storage systems provided by Alibaba Cloud, include relational databases, FTP, HDFS, which can be configured as a source or a destination in the synchronization task. You can also create a task that runs periodically to synchronize data on a periodic basis.**
- **Data management system: A system that can be used to manage data in MaxCompute. You can manage permissions, view the data lineage, and view the metadata.**

15.1.2 Scenarios

Large data warehouses

Enterprises can use DataWorks in Apsara Stack to build large data warehouses.

DataWorks provides superior data processing capabilities:

- **Massive storage: Supports PB- and EB-level data warehouses and linear expansion of storage size.**
- **Data integration: Supports data synchronization and integration across heterogeneous data sources to eliminate data islands.**
- **Data analytics: Supports MaxCompute-based big data processing capabiliti es, programming frameworks such as SQL and MapReduce, and a visualized workflow designer.**
- **Data management: Supports unified metadata management and permission-based data access control.**
- **Batch scheduling: Supports periodic task execution and processing for millions of tasks in a day, real-time task monitoring, and timely error alerting.**

Data-driven management

- **Innovative businesses: Data mining, data modeling, and real-time decision making can be implemented based on big data analytics results provided by DataWorks.**
- **Small and medium-sized enterprises: With DataWorks, data can be quickly analyzed and put to commercial use, which help enterprises generate marketing strategies.**

15.2 Technical advantages

Capability of processing big data

DataWorks uses MaxCompute as its computing engine, which supports a maximum of 5,000 servers in a single cluster. DataWorks can access data from different clusters, which allows you to easily process your big data. The offline scheduling system can run millions of concurrent jobs. You can also configure rules and alerts to monitor the running of nodes in real time.

Key features

- **DataWorks supports join operations for trillions of data records, millions of concurrent jobs, and petabytes (PB) of I/O throughput each day.**
- **DataWorks allows you to share data across clusters and data centers, and scale out clusters to a maximum of tens of thousands.**
- **DataWorks provides efficient and easy-to-use SQL and MapReduce engines, and supports most standard SQL syntax.**
- **MaxCompute protects user data from loss, breach, or theft by using multi-layer data storage and access security mechanisms, including triplicate backups, read/write request authentication, application sandboxes, and system sandboxes.**

Integrated data processing environment

DataWorks integrates development, scheduling, monitoring, and alerting for nodes , and management of data.

Key features

- **DataWorks provides you with all the required features for data processing.**
- **You can design and edit workflows in a visual designer that is similar to Kettle.**

- **DataWorks provides a collaborative development environment. You can create and assign roles for varying nodes, such as development, online scheduling, maintenance, and data permission management, without locally processing data and nodes.**

Integration from disparate data sources

DataWorks supports reading data from 11 disparate data sources and writing data to 12 disparate data sources. You can also configure dirty data filtering and bandwidth throttling.

Key features

- **DataWorks can read data from data sources of the following types: MySQL, Oracle , SQL Server, PostgreSQL, ApsaraDB for RDS, ApsaraDB for DRDS, MaxCompute , FTP, Object Storage Service (OSS), Hadoop Distributed File System (HDFS), Dameng, and Sybase.**
- **DataWorks can write data to data sources of the following types: MySQL, Oracle , SQL Server, PostgreSQL, ApsaraDB for RDS, ApsaraDB for DRDS, MaxCompute, AnalyticDB, Memcache, OSS, HDFS, Dameng, and Sybase.**
- **DataWorks supports dirty data filtering and bandwidth throttling.**
- **DataWorks supports recurring nodes, including recurring sync nodes.**

Web-based software

You can use DataWorks whenever an internal or a public network is available.

Multitenancy

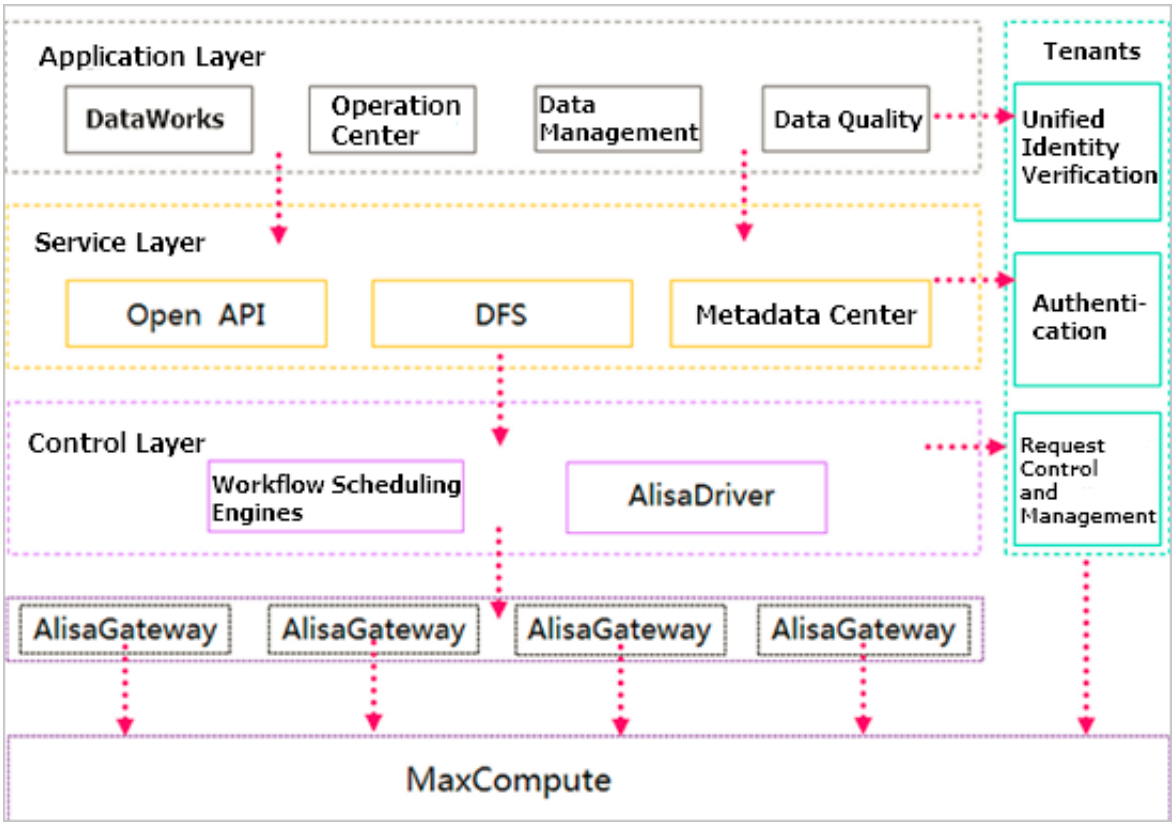
DataWorks uses multitenancy to isolate data among tenants. Each tenant separately manages their own permissions, data, resources, and members.

Open platform

DataWorks provides all modules as components and services. You can use APIs to develop extra functions of DataWorks.

15.3 Architecture

System architecture



DataWorks adopts the design of components and services, and consists of the following three layers.

- **Control layer: the core of DataWorks batch data processing. The workflow scheduling engine generates and runs node instances. AlisaDriver coordinates and controls the running of all nodes.**
- **Service layer: provides services for the application layer and other external applications.**
- **Application layer: runs on top of the service layer, and provides the graphical interface for user interactions.**

Security architecture

The security architecture of DataWorks features error proofing, basic security, and optional security tools.

- **Error proofing ensures proper running of DataWorks during coding, deployment , and configuration.**

- **Basic security ensures the security of data for DataWorks by using features such as resource isolation among tenants, user identity verification, authentication, and log auditing.**
- **Optional security tools in DataWorks allow you to customize security policies for the protection and management of your system and data.**

Multi-tenancy

DataWorks adopts multi-tenancy.

- **Storage and compute resources are scalable. You can manage your own resources, and request resource quotas as needed.**
- **Tenants are isolated. Each tenant separately manages its own data, permissions, accounts, and roles.**

15.4 Services

15.4.1 DataStudio

DataStudio is an integrated development environment (IDE) in DataWorks, which supports database warehousing, data query, ETL, and algorithm development for big data. It also supports online collaborative development and version control.

Features

- **DataStudio provides a visualized workflow designer, which is similar to the Kettle tool. It allows you to design workflows, and manage nodes of each workflow.**
- **DataStudio supports the upload of local files.**
- **DataStudio supports data integration from heterogeneous data stores.**

Note:

Data Integration supports the following data store types:

- **Types of source data stores: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, FTP, OSS, HDFS, Dameng, and Sybase.**
- **Types of target data stores: MySQL, Oracle, PostgreSQL, RDS, MaxCompute, Memcache, OSS, HDFS, Dameng, and Sybase.**

- **DataStudio provides a web-based programming and debugging environment that allows you to create SQL, MR, shell (limited support), and data synchronization nodes.**
- **DataStudio supports node deployment across MaxCompute projects. You can deploy nodes and code to the scheduling system across different workspaces.**
- **DataStudio adopts version control, node locking, and conflict detection mechanisms to facilitate collaborative development.**
- **DataStudio enables you to search and use MaxCompute tables, resources, and user-defined functions.**

15.4.2 Data Management

Data Management enables you to perform queries on the tables, view details of tables, and manage permissions on tables. You can also add tables to your favorites. For more information, see the Data Management topic in *DataWorks User Guide*.

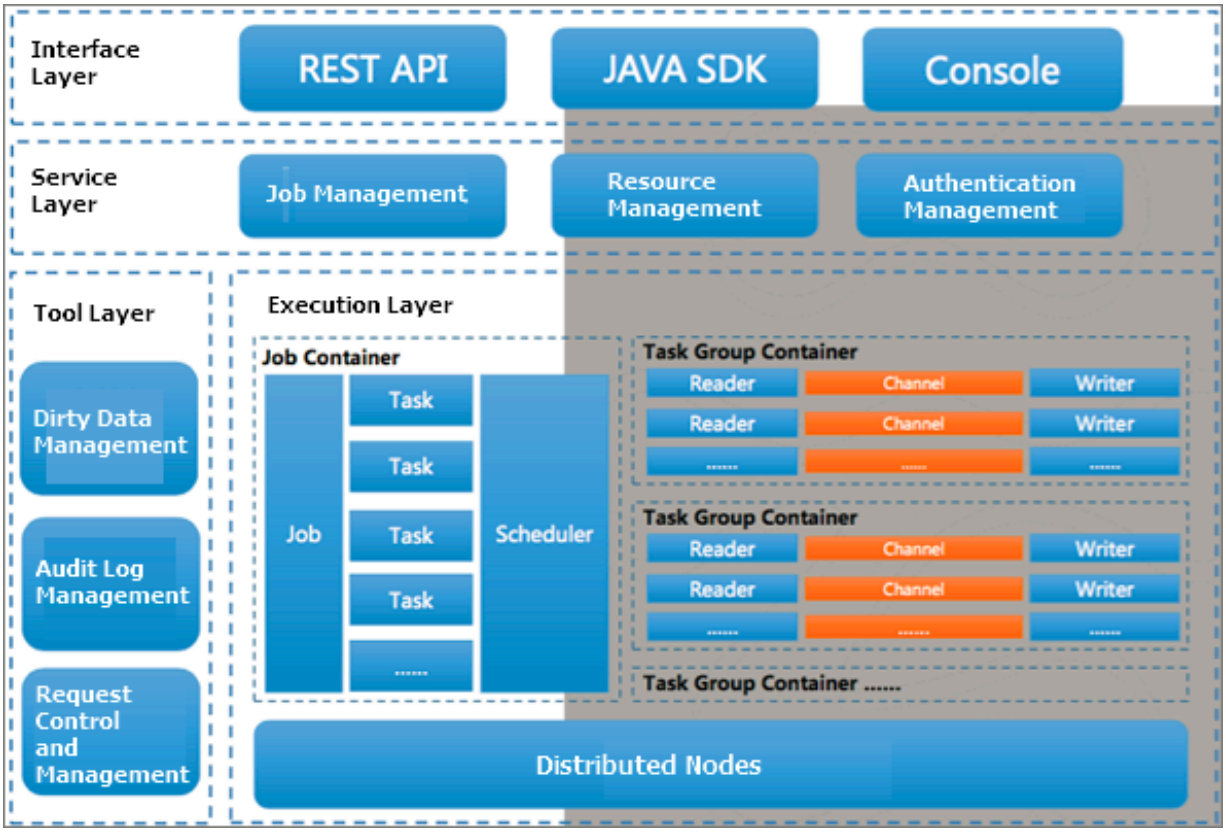
15.4.3 Data Integration

Data Integration is a data synchronization platform that provides stable, efficient, and scalable services. Data Integration provides transmission channels for batch data stored in MaxCompute, and Realtime Compute. Data Integration implements fast integration on data from heterogeneous data stores.

Data Integration provides connectors and a framework. The connectors are used for reading and writing data, and the framework is used for common operations in data synchronization and transmission. Data Integration provides two types of connectors:

- **Reader: reads data from the data store**
- **Writer: writes data to the data store**

You can develop readers and writers for Data Integration to support more data store types.



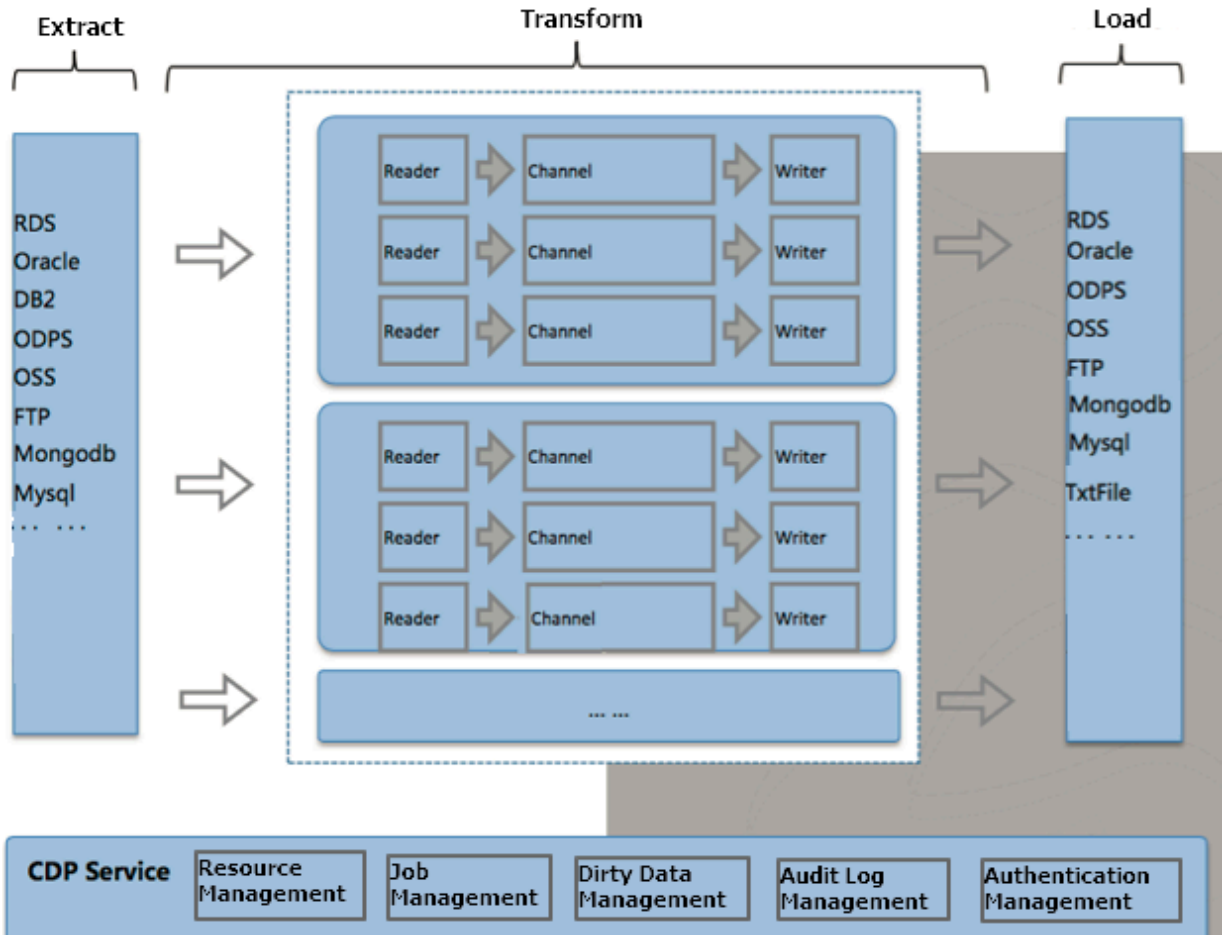
The interface layer provides three methods of using the Data Integration service: RESTful API, Java SDK, and console. RESTful API supports multiple programming languages. We recommend that you use Java SDK to avoid manual operations such as signature, authentication, and HTTP request making. The console is developed based on the command line tool, which allows you to use the majority of Data Integration functionalities. Data Integration also provides developers with a web UI based on RESTful API.

The service layer includes resource management, job management, and authentica tion management. For more information, see the product overview.

The tool and execution layers form the core of Data Integration. The two layers run extract, transform, load (ETL) jobs. All synchronization jobs that are submitted to

Data Integration run on the execution layer. The execution layer uses DataX as the synchronization engine.

Figure 15-2: Process of an ETL job



Features

- **Data Integration supports the following types of data stores:**
  - **Relational databases: MySQL, PostgreSQL, Oracle, Db2, and general relational databases**
  - **NoSQL databases: Table Store and Memcache**
  - **Big data platforms: MaxCompute**
  - **Unstructured data stores: OSS, HDFS, and FTP**

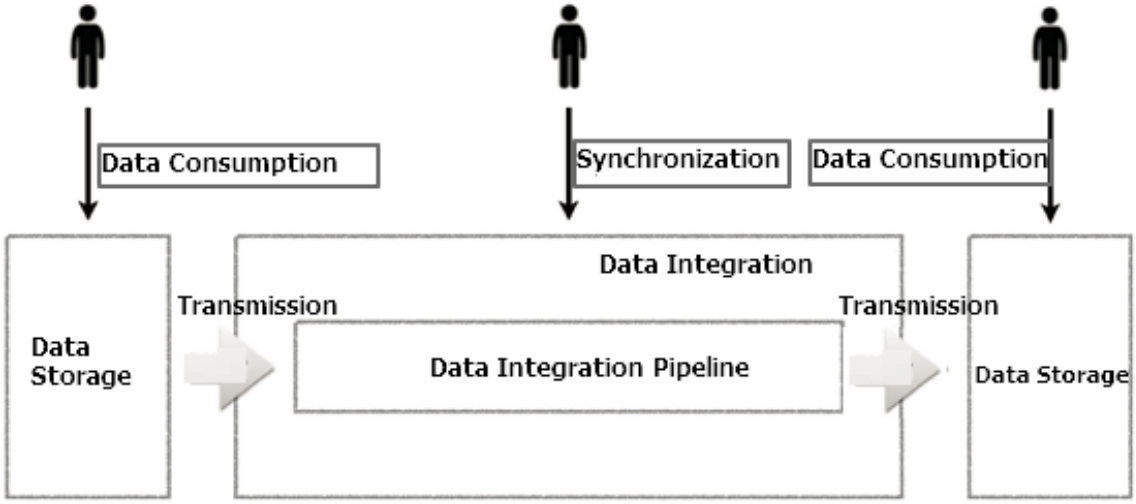
  You can use the following JDBC URLs when you configure connections to general relational databases such as Dameng, Db2, and PPAS:

  - **Dameng: jdbc:dm://ip:port/database**
  - **Db2: jdbc:db2://ip:port/database**
  - **PPAS: jdbc:edb://ip:port/database**

  Data Integration supports recurring batch synchronization from the source to the target. For example, you can configure a data synchronization node that runs on a daily, weekly, or monthly basis. When the batch data synchronization node starts, a snapshot of source data is taken. The system then reads data from the snapshot and writes the data to the target. Each batch data synchronization node has a life cycle.

  Data Integration only defines the synchronization process. Data transmission during the synchronization process is under the control of the Data Integration cluster. Data channels and streams are invisible to users. Data Integration does

not provide any API for data consumption. You need to consume data on both the source and target data stores.



- Consistent data quality
  - Data Integration supports conversions between different data types.
  - It accurately identifies, filters, collects, and displays dirty data to ensure the quality of data.
  - Data Integration supports job performance reporting, which helps you track node status, such as data volume and dirty data.
- Efficient data transmission
  - Data Integration supports one-way data channels, and allows a single process to reach the maximum data transfer rate (up to 1,600 Mbit/s) on each server.
  - It adopts a distributed architecture, and supports transmission for gigabytes (GB) to terabytes (TB) of data.
- User-friendly control experience
  - Data Integration implements accurate control of channels, record streams, and byte streams.
  - You can rerun any threads, processes, and jobs that fail.

- **Clear core design**
  - **Data Integration provides a professional framework and an efficient execution engine. The engine supports common connectors, standardizes the process of developing connectors, and automatically detects new connectors.**
  - **Data Integration provides clearly defined and easy-to-use connector APIs that allow developers to focus on the implementation of the connector instead of the framework.**

15.4.4 Tenant management

- **Workspace management**

  The Workspace Management page includes basic workspace settings.

  - **Sandbox whitelist: IP addresses and domains that can access the workspace**
  - **Compute engine: information of the MaxCompute engine**
- **User management**

  On the Members page, you can assign and unassign a role from specified members.
- **Permission list**

  On the Permissions page, you can view the permissions of a role for tables and workspaces.

15.4.5 Data Quality

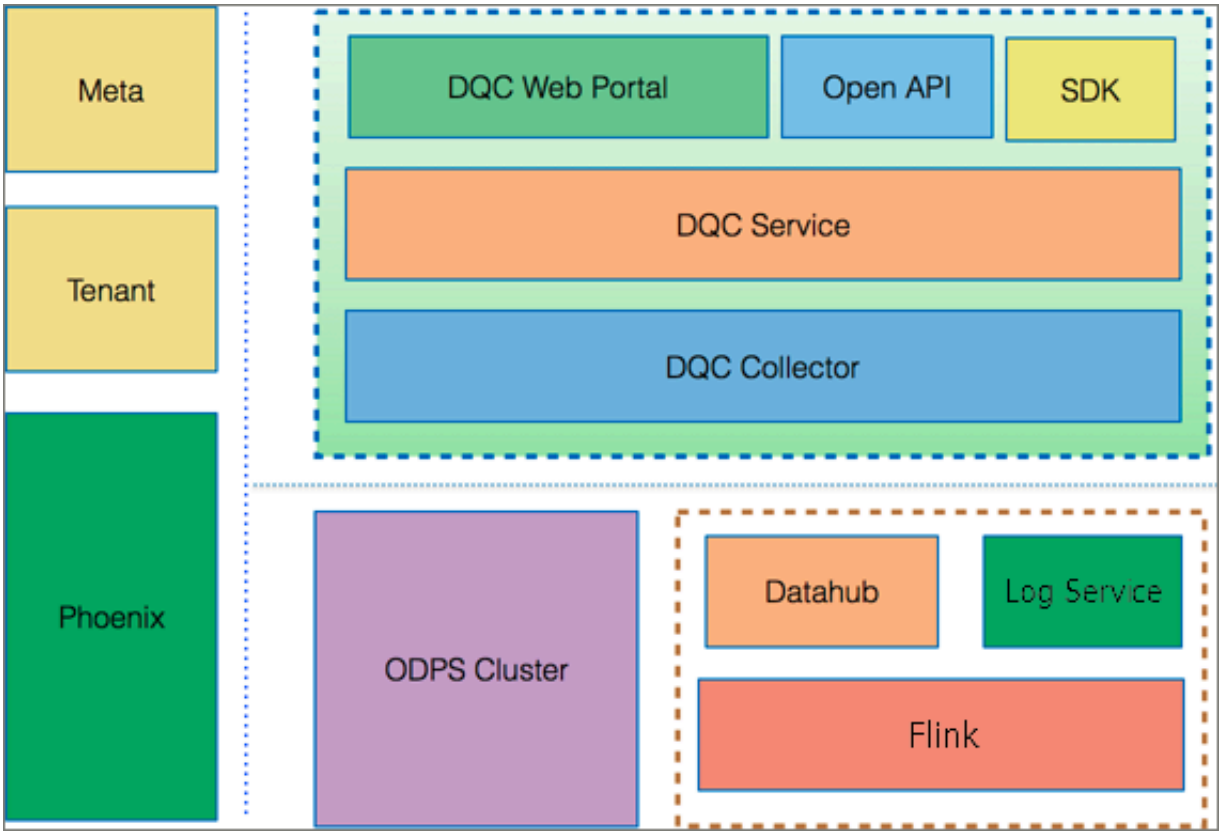
15.4.5.1 Overview of Data Quality

Data Quality is a platform that provides data quality check and management services. You can use it to monitor both real-time and batch data during the entire data processing cycle. When you use Data Quality to monitor real-time data, it can detect discontinuity, delay, and other user-defined data issues in DataHub data streams. When you use Data Quality to monitor batch data, it can detect abnormal data in the production process, protect downstream data from being affected by abnormal data, and promptly notify you about the abnormal data. This helps to ensure the correctness of your data.

Data Quality requires the access to the metadata, fields, and tables, and requires user and tenant management. In the scenario of monitoring batch data, Data Quality uses MaxCompute as the compute engine. In the scenario of monitoring

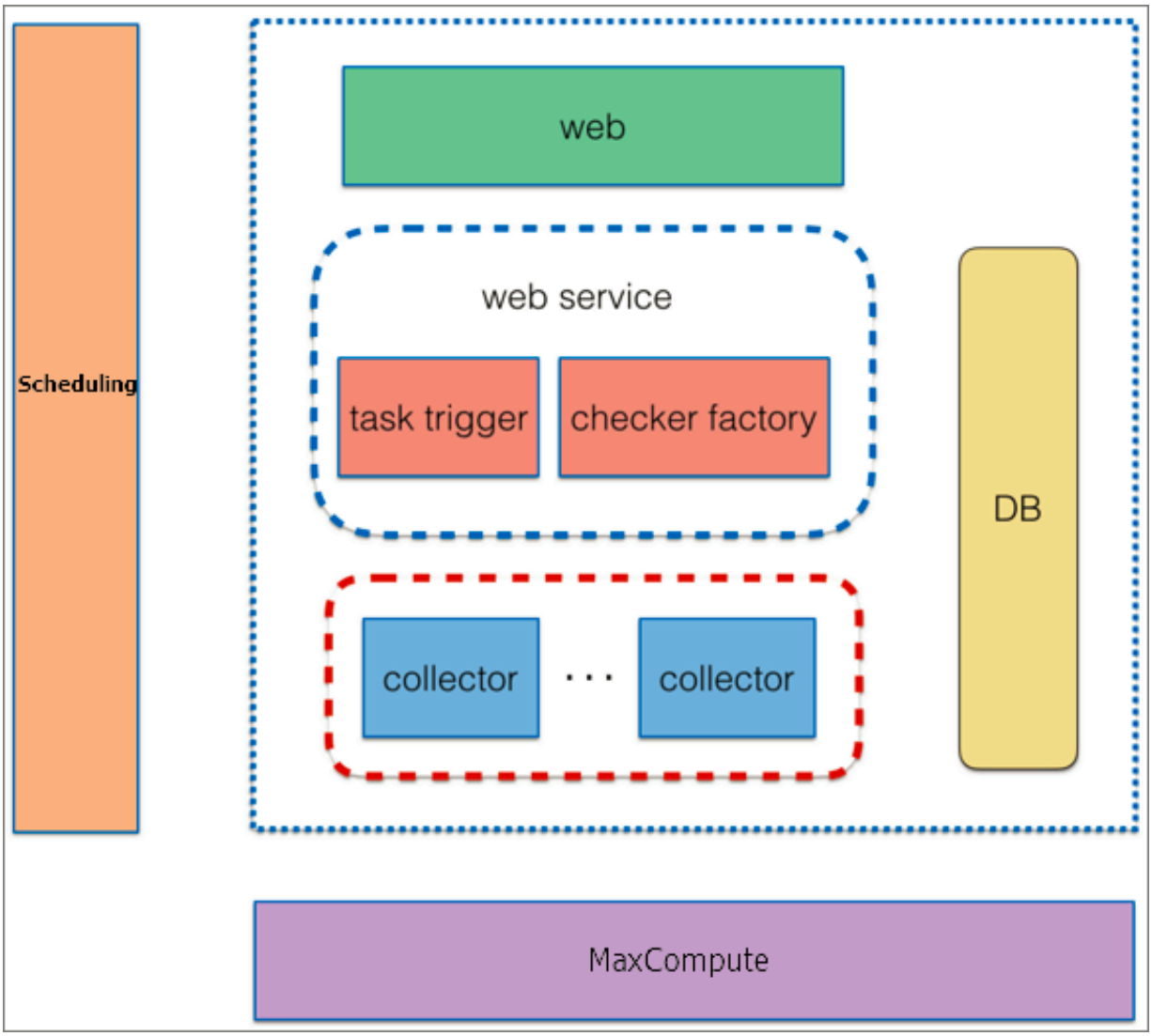
real-time data, Data Quality uses the Flink framework as the streaming data processing tool. Data Quality consists of three components: the web portal, the check service, and the data collection service.

Figure 15-3: Data Quality architecture



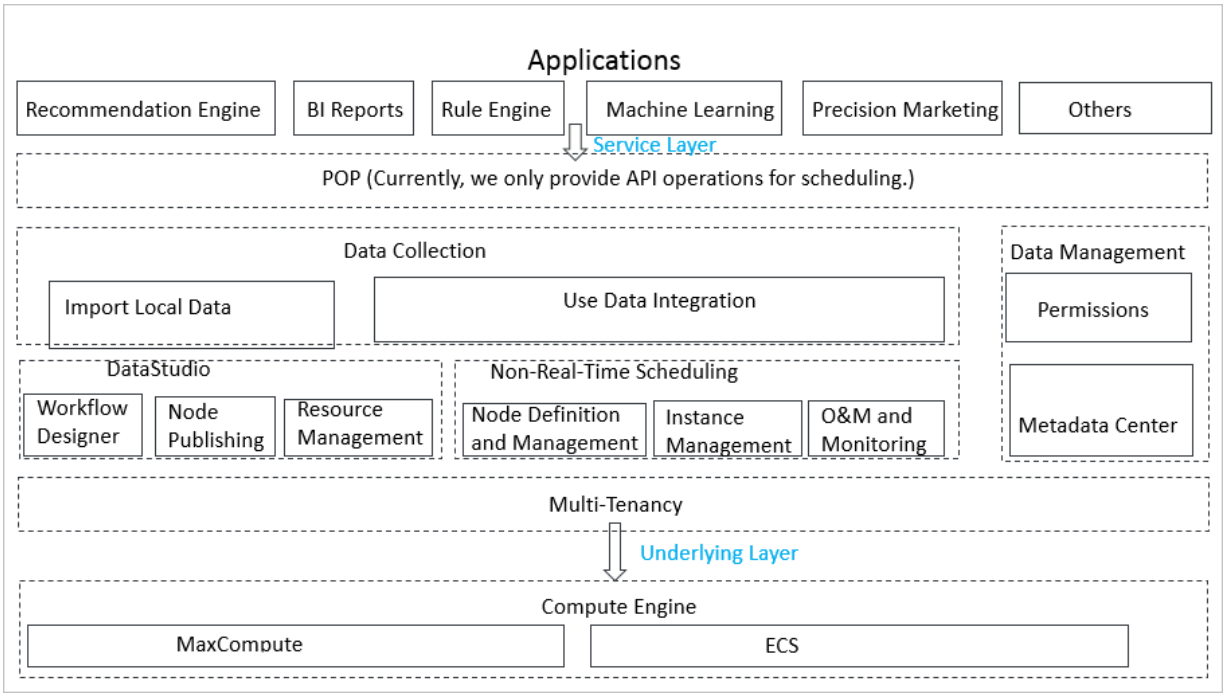
15.4.5.2 Use Data Quality to monitor batch data

Architecture



- **Web: Web UI provides a graphical interface for users. It consists of rule management, search by node, subscription management, dashboard, permission control, and cache management.**
- **Web service: The web service layer provides access to databases, checks data quality, parses jobs, and triggers jobs. The checker factory module checks samples by using quality check logics such as comparison of fixed value, fluctuation, and variance detection.**
- **Collector: The collector module consists of multiple data collection engines that obtain data samples based on user specified rules. Data collection engines classify the rules based on potency, rule types, and sampling methods. Before sending the rules to MaxCompute to obtain data samples, data collection engines apply logical splitting and combination to the rules.**

Principle



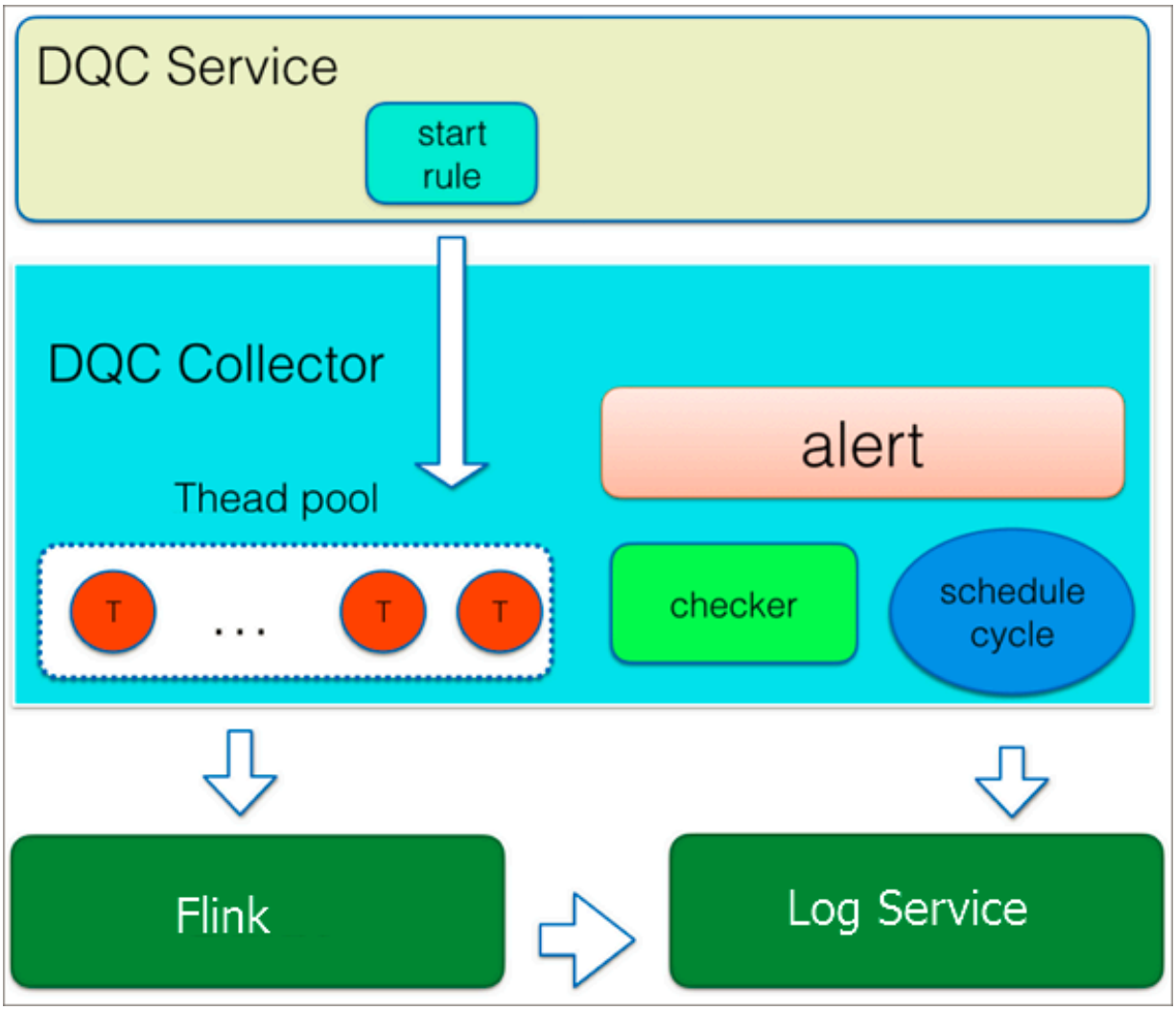
1. **The scheduling system sends a request that triggers the service module to check the quality of data in the specified partitions of a table. The request contains the partition expression, table information, and node schedule.**
2. **Based on the partition expression, a server in the service module obtains the set of rules that are applied to the current node. The server submits a request for obtaining data samples to data collection engines and returns the request result to the scheduling system. The scheduling system first allocates resources to run nodes that are associated with strong rules.**
3. **The data collection engines further classify the set of rules based on potency , rule types, and sampling methods. The MaxCompute cluster collects data samples based on the sampling methods.**
4. **After finishing data sampling based on strong rules, the data collection engines instruct the service module to check data quality. After the quality check, the service module sends the check results to the scheduling system, and the scheduling system determines whether to block the node.**
5. **After the quality check by using strong rules, the service module returns the results to the data collection engines. The data collection engines continue the sampling process, and send the processed data for check based on weak rules. After the weak rule check is complete, the quality check ends.**

Benefits

- **Data Quality provides built-in rule templates and comprehensive data quality metrics. The templates support filed and table level rules with a fluctuation threshold or fixed value comparison. You can create rules from the templates to check whether data entries are null or unique or use discrete values, the maximum, minimum, average, or sum to evaluate the data quality. You can also create custom rules for special requirements.**
- **Data Quality clusters are horizontally scalable. You can add servers if Data Quality reaches the maximum concurrency. Data Quality also includes a reliable fault-tolerance system that ensures that data collection jobs are accurate and consistent.**
- **Data Quality supports rule classification based on potency and severity levels . When you use Data Quality to monitor batch data, you can classify rules into weak and strong rules based on potency. You can also set thresholds to reflect the warning and error severity levels of check results based on the deviation from the expected value. When strong rule check results show a significant deviation from expected values, the node is blocked to protect downstream data against dirty data. This ensures the correctness of data during the data processing cycle.**
- **Data Quality provides a potency based execution mechanism that first runs the nodes that are associated with strong rules. The collector module supports running nodes based on the potency.**
  - **If available resources are limited, this mechanism ensures that you first run nodes that are associated with strong rules.**
  - **If available resources are sufficient, this mechanism allows nodes that are associated with weak rules to run.**

15.4.5.3 Use Data Quality to monitor real-time data

Architecture



Rules for monitoring real-time data are converted into Flink SQL statements. Data Quality uses Flink to read data from DataHub and write check results to Log Service. The collector module of Data Quality regularly obtains abnormal data from Log Service, writes the data to Redis, and then triggers alerts. The service module of Data Quality synchronizes the alerts from Redis to other databases for users to query.

Principle

1. **After you enable a rule, the service module creates a Logstore. The service module uses an SQL parser to declare a dimension table used for referencing a DataHub topic. The service module uses a rule converter to generate a CREATE TABLE statement and combine table operations. Then, the service module submits a Flink job and updates the next quality check time.**

2. One of the servers in the service module first establishes a lock to serve as the master. The master collects data from DataHub topics on a regular basis and sends the data to the collector module for quality check.
3. The collector module uses a LogHub consumer to subscribe to the Logstore. Then, the collector module writes abnormal data to Redis, and determines whether to send alerts.
4. The service module starts the Quartz scheduler worker service, and writes the data from Redis to another database for users to query.

Benefits

- You can use Data Quality to monitor real-time data in various scenarios. Data Quality detects discontinuity and delay of real-time data streams, and allows you to create Flink SQL queries as custom rules. Data Quality also supports join operations for multiple streams and dimension tables.
- Data Quality supports monitoring on data delay at the level of seconds.
- Data Quality allows you to specify an alert interval and the number of alerts for each rule to reduce redundant alerts.
- Data Quality allows you to set thresholds at the warning and error severity levels. This helps you identify the deviation of check results from expected values.
- Data Quality uses hashing algorithms to remove duplicate alerts. This ensures data idempotence in the real-time computing process.

15.4.6 Data Asset Management

Data Asset Management provides portal management, data asset category management, data source management, and business unit management. Using the Data Asset Management service helps you understand your core data assets and standardize your management process.

15.4.7 Real-Time Analysis

The Real-Time Analysis service, which is based on MaxCompute, provides you with quick data query and data preview. The service is suitable for data analysis and data exploration.

- Supports creating, renaming, and deleting folders and files.

1. Click Run to run the SQL statements.
2. View the results.

15.4.8 Data Service

Data Service provides features such as API hosting, authentication, authorization, and management. You can create APIs for tables, and publish the APIs by using the API Gateway service.

Features:

- **Supports various data sources, including relational databases, and NoSQL databases.**

  Supported data sources: MySQL, Oracle, PostgreSQL, ApsaraDB for RDS, Table Store, MongoDB, and Lightning.
- **Provides the wizard mode, which can be used to create APIs without writing code .**
- **Provides the script mode. You can create APIs by writing SQL statements.**
- **Provides accurate access control. You can customize permissions on APIs, table rows, and table columns.**
- **Provides API Gateway and HTTP request methods.**
- **Supports a variety of network environments, including local private networks, VPCs, and Alibaba Classic networks.**
- **Provides API creation, grouping, and publishing.**
- **Provides organization-based API isolation.**
- **Provides Open API, which supports registering, managing, and testing APIs.**
- **Supports a variety of API execution environments, including stand-alone environments and the EAS container service.**
- **Supports debugging APIs online. You can view the API call information and the performance in real time.**

15.4.9 Intelligent Monitor

Intelligent Monitor is a system that monitors and analyzes tasks in DataWorks. Intelligent Monitor sends alerts based on specified rules, times, methods, and recipients. It automatically uses the most appropriate alert time, method, and recipients. Intelligent Monitor has the following benefits:

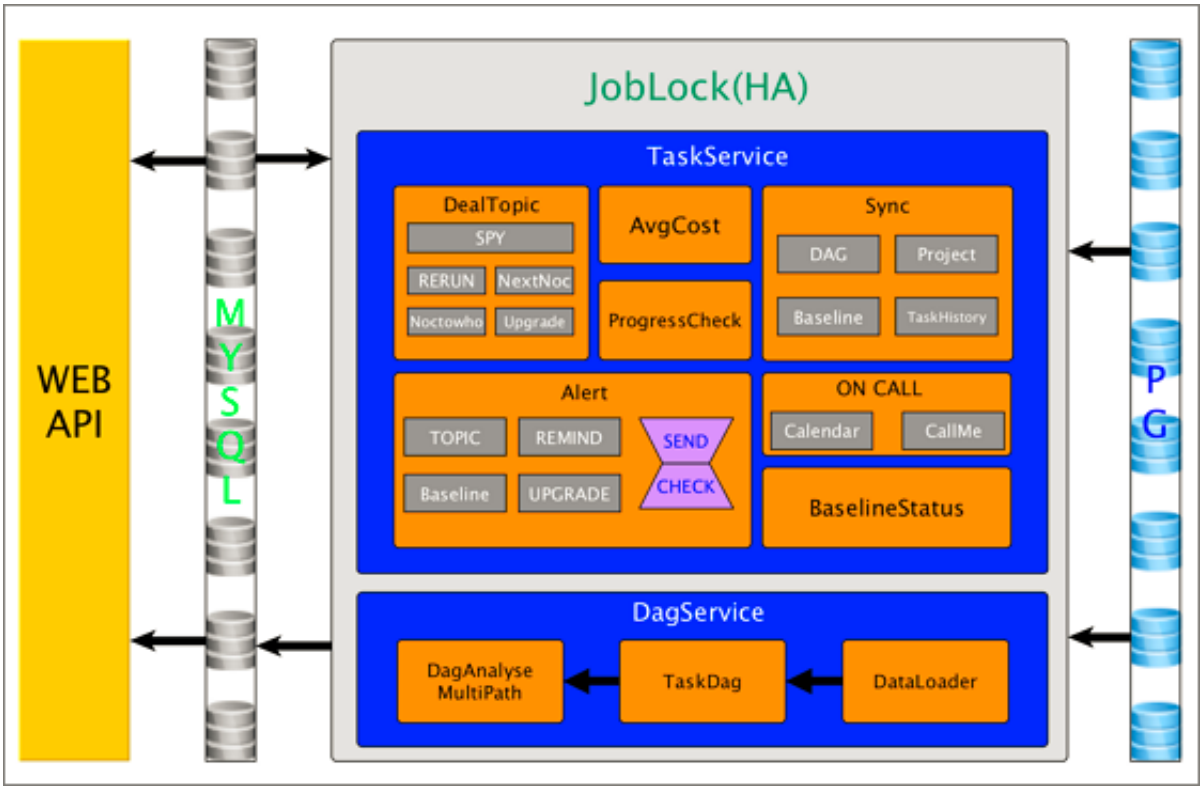
- **Alerting rules are easy to configure.**
- **You are free of invalid alerts.**
- **All important tasks are monitored.**

General monitoring systems cannot meet the requirement of DataWorks. The reasons are:

- **DataWorks has considerable tasks, and you cannot accurately locate all tasks that need to be monitored. Dependencies between tasks are complex. Even if you know what are the most important tasks, you have difficulties in figuring and monitoring all related tasks. If you monitor all tasks, invalid alerts may be generated. You have to determine which alerts are useful.**
- **Different tasks require different alert configurations. Some alerts are sent when the task runs for more than one hour, while others are sent when the task runs for more than two hours. It is difficult to specify different configuration for each single task.**
- **Different types of alerts are sent at different time. For example, an alert for an unimportant task can be send after you begin to work, and an alert for an important task should be sent immediately when the error occurs. Moreover, the importance of each task is hard to determine.**
- **Different alerts require different operations to turn off.**

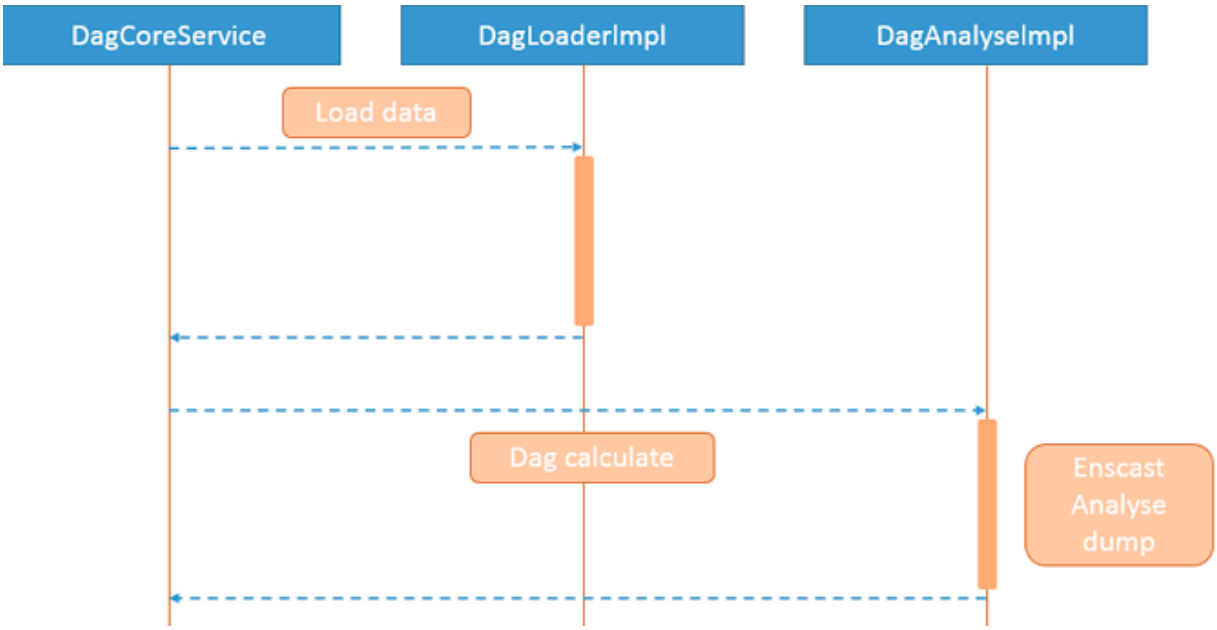
The Monitor feature automatically creates alert rules for tasks that are configured with baselines. You can also customize alert rules by completing basic settings.

Technical architecture



- **Dagservice: Analyzes all tasks on each DAG based on the baseline settings. Dagservice determines the estimated completion time, the key path, the required completion time, and whether the task is suspended. The information collected by Dagservice provides the basis for TasksService.**
- **TaskService: Performs different tasks based on the information provided by DagService, including estimating the completion time, acquiring and fixing events, and customizing baseline alerts.**
- **WebService: Provides HTTP APIs that can be called to send requests. You can call APIs to view the Intelligent Monitor information, such as baseline instances, alert information, events, and gantt charts.**

How it works



DagService collects the information of all nodes on each DAG based on the baselines and the average running time of each task. The information contains the estimated completion time, the required completion time, the key path, whether the node is blocked, and whether the node is a child of a suspended node.

TaskService runs tasks based on the task configuration and the information provided by DagService. The database lock ensures that one task is executed by only one server. When a server is down, another server takes over the task, which ensures the high availability of the monitoring service.

15.4.10 Scheduling system

15.4.10.1 Overview

The scheduling system is one of the core systems in DataWorks, which is responsible for scheduling all tasks. The scheduling system runs tasks based on the specified time and the dependencies.

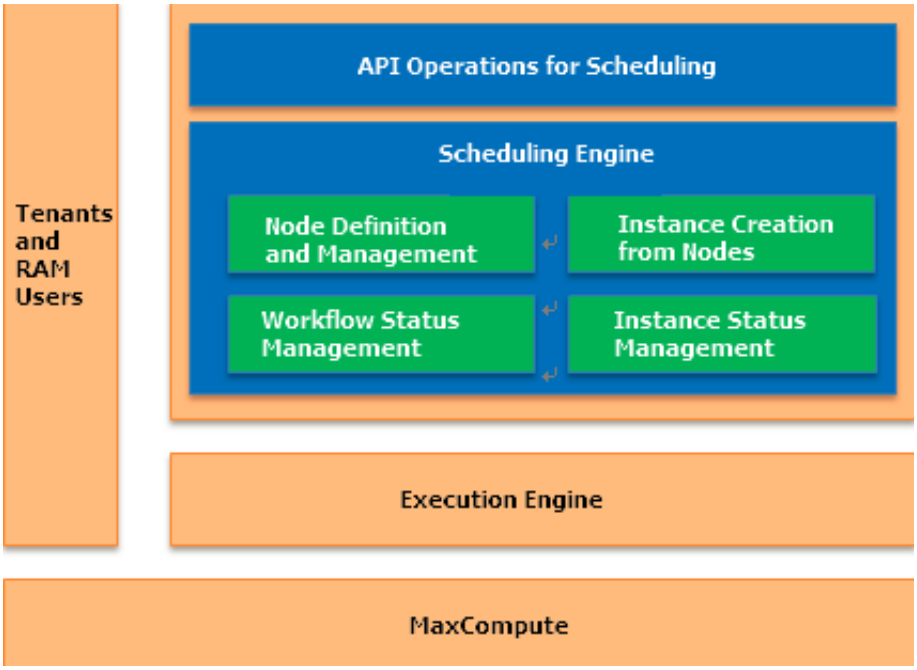
- **The scheduling system can schedule millions of jobs.**
- **The distributed execution architecture enables linear scalability of concurrent jobs.**
- **You can configure scheduling tasks at custom intervals in minutes, days, hours, weeks, or months.**
- **You can configure the dependencies of different tasks. You can choose from general dependency, cross-period dependency, or self-dependency.**
- **You can dry run a task or suspend a task.**
- **You can create and run an ad-hoc workflow.**
- **You can view the workflow in a directed acyclic graph (DAG), which provides you with a clear view while troubleshooting.**
- **You can monitor tasks in real time, and send alerts by sending SMS messages or emails.**
- **You can rerun tasks, terminate processes, set the status of tasks to Successful, or suspend tasks.**
- **You can create retroactive tasks. Instances in different periods run serially.**
- **You can view the total number of tasks, the number of task errors, the number of scheduling tasks, the top 10 scheduling tasks that consume the most resources, the top 10 scheduling tasks that take the longest to run, and the distribution of task types.**

15.4.10.2 Concepts

- **Node: A node represents a task in the scheduling system. Node properties include the node type, the code version, the specified task start time, and the dependencies between tasks.**
- **Instance: An instance is created whenever a task (node) runs. The instance has all properties that the task (node) has. In addition, the instance contains the runtime information such as the instance status and the time when the status changes.**

- **Workflow: A workflow is composed of several interdependent instances. The scheduling system consolidates all instances in a day into a workflow for unified management. A workflow has its own status, which is determined by the status of each instance in the workflow.**

15.4.10.3 Architecture



The preceding figure shows the architecture of the scheduling system and its relationship with other systems.

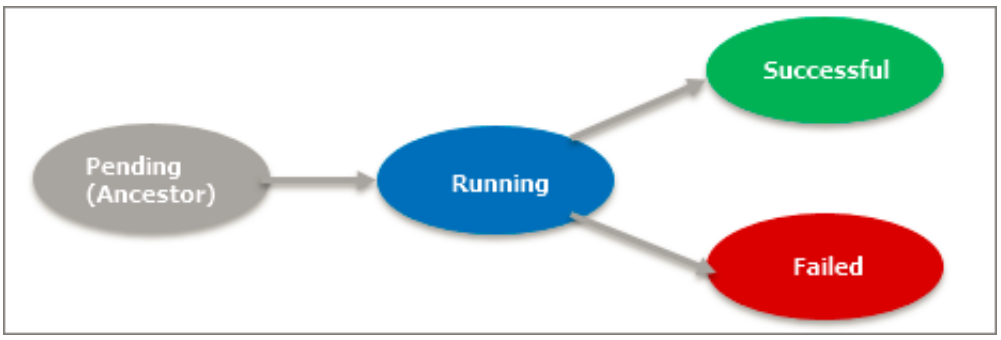
The scheduling engine is the core of the scheduling system. It contains four main modules.

- **The node definition and management module maintains node definitions submitted by users, including the code, the specified task start time, and the dependencies. An instance is generated from the node configurations at a fixed time every day.**
- **The instance status management module manages the status changes after an instance runs.**
- **The workflow status management module maintains the status changes after a workflow runs. (A workflow is a set of instances with dependencies.)**
- **The scheduling system provides APIs for other systems to perform INSERT, DELETE, UPDATE, and SELECT operations.**

The resources that are used by the scheduling system are isolated among tenants . Before a task instance runs, the scheduling system schedules the instance to the execution engine.

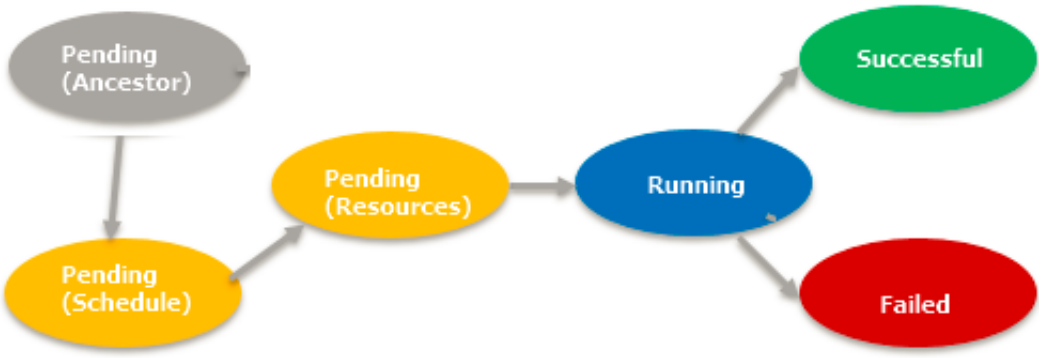
15.4.10.4 State machine

Workflow state machine



- **A workflow has four statuses: Not Running, Running, Successful, and Failed.**
- **The initial status of a workflow is Not Running. At this time, all instances in this workflow are in the Not Running status. When the workflow is invoked by the scheduling system, its status changes to Running and the root instance of the workflow runs.**
- **When an instance in the workflow fails, the status of the workflow changes to Failed.**
- **When all instances in the workflow are in the Successful status, the status of the workflow changes to Successful.**

Task instance state machine



- **A task instance has six statuses: Not Running, Waiting for Scheduled Time, Waiting for Resources, Running, Successful, and Failed.**
- **The initial status of a task instance is Not Running. When it is invoked by the scheduling system, the system checks whether all its predecessor tasks are in**

the Successful status. If yes, the status of the instance changes to Waiting for Resources.

- The task instance is invoked at the time that is specified for running the task. The instance is then sent to the execution engine and its status changes to Waiting for Resources.
- The execution engine allocates resources to the instance. The instance runs, and the scheduling system changes the status of the instance to Running. The execution engine then sends the result to the scheduling system, and then the scheduling system changes the instance status (to Successful or Failed) accordingly.

15.4.10.5 Task dependencies

You can configure dependencies for tasks based on your business requirements.

Same-period dependency

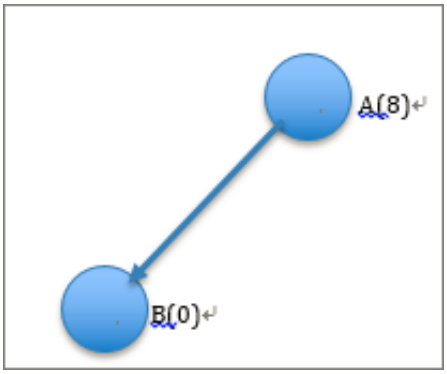
This is the most common scenario where an instance only depends on its parent instances in the same day. You can configure the following dependencies: A daily instance depends on another daily instance, a daily instance depends on an hourly instance, an hourly instance depends on a daily instance, or an hourly instance depends on another hourly instance.

If an hourly instance depends on another hourly instance, three situations can occur: The number of parent instances is equal to the number of child instances, the number of parent instances is greater than the number of child instances, or the number of parent instances is less than number of child instances. The following examples show all the situations.

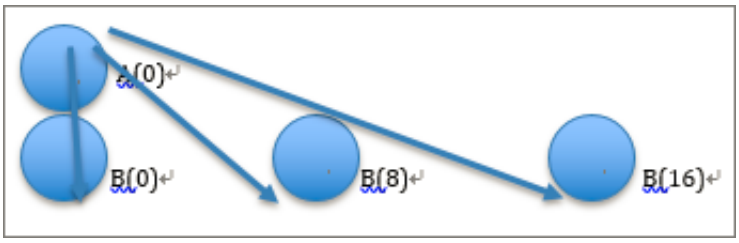
Note:

In the following examples, all A nodes are parent nodes, and all B nodes are child nodes.

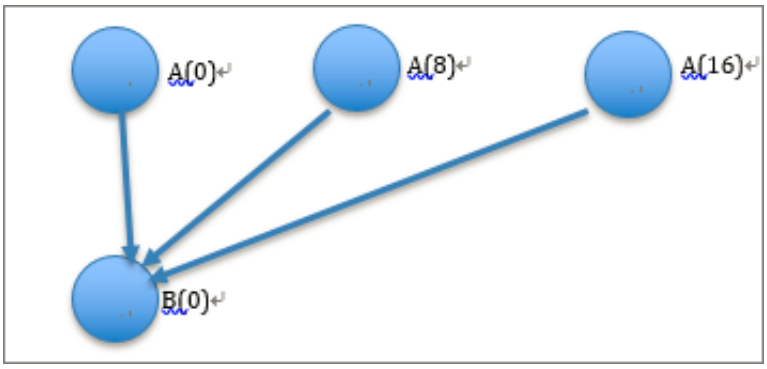
- A daily instance depends on a daily instance. The B node is specified to run at 08:00. The A node is specified to run at 00:00.



- An hourly instance depends on a daily instance. The B node is specified to run at 00:00, 08:00, and 16:00. The A node is specified to run at 00:00.

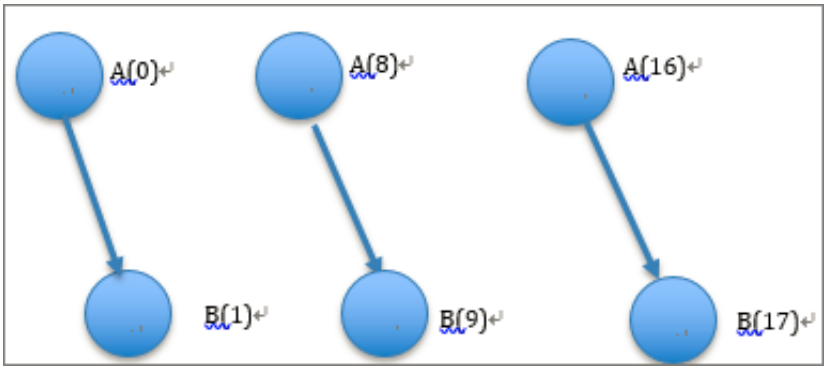


- A daily instance depends on an hourly instance. The B node is specified to run at 00:00. The A node is specified to run at 00:00, 08:00, and 16:00.

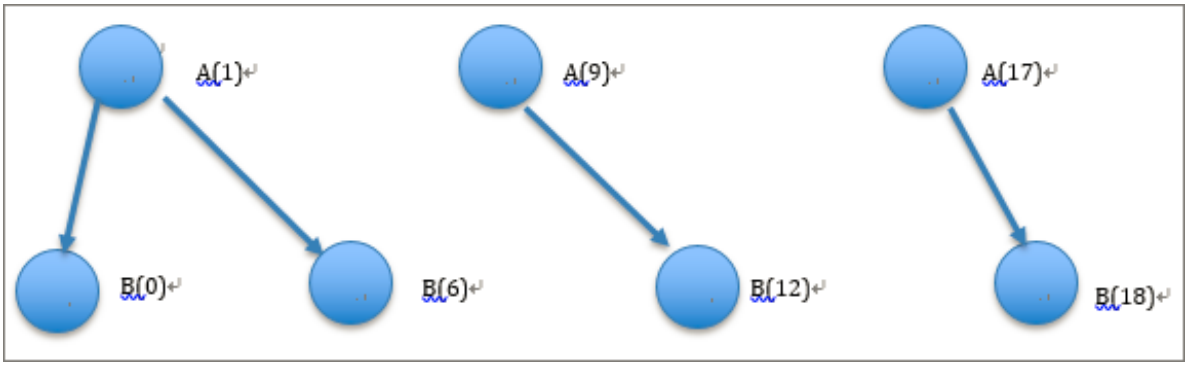


- An hourly instance depends on an hourly instance, and the number of parent instances is equal to the number of child instances. The B node is specified to

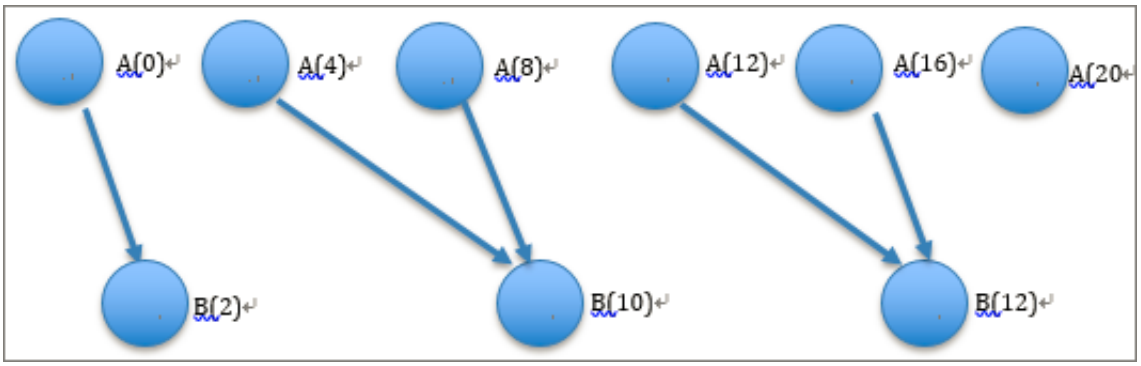
run at 01:00, 09:00, and 17:00. The A node is specified to run at 00:00, 08:00, and 16:00.



- **An hourly instance depends on an hourly instance, and the number of parent instances is less than the number of child instances. The B node is specified to run at 00:00, 06:00, 12:00, and 18:00. The A node is specified to run at 01:00, 09:00, and 17:00.**



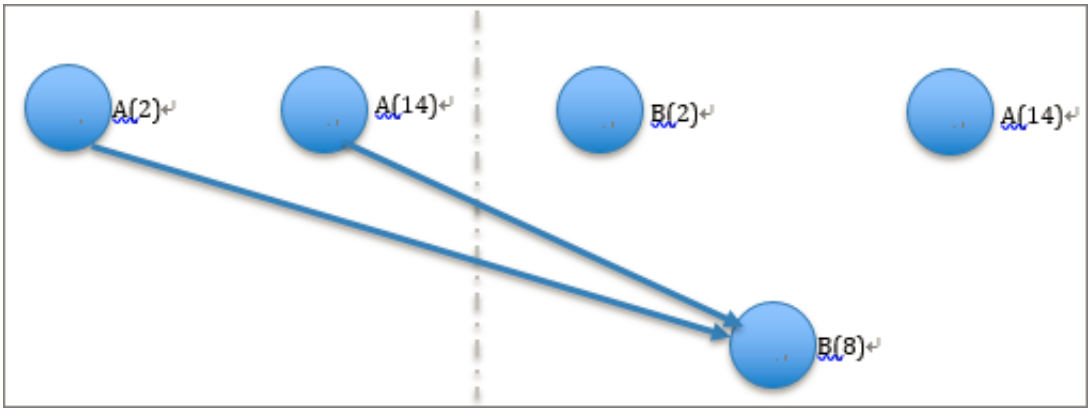
- **An hourly instance depends on an hourly instance, and the number of parent instances is greater than the number of child instances. The B node is specified to run at 02:00, 10:00, and 18:00. The A node is specified to run at 00:00, 04:00, 12:00, 16:00 and 20:00.**



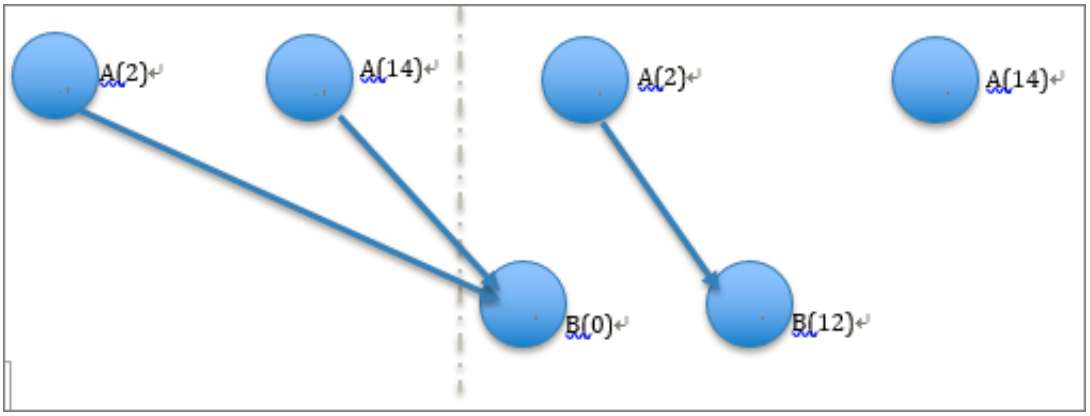
Cross-period dependency

You can configure cross-period dependency if the data processing operation requires the result of the data processing operation on the previous day.

- **In most cases, you only need to configure the dependency between the current instance and the instance in the last day. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 08:00.**

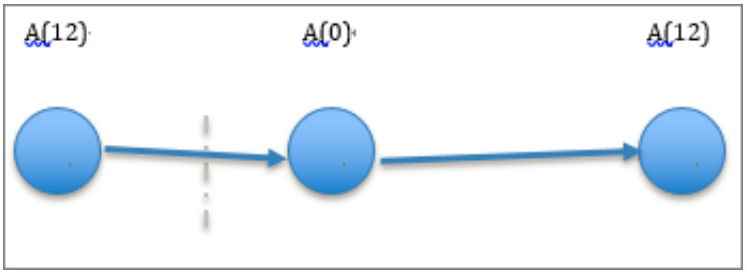


- **The same period dependency and the cross period dependency can both exist. Suppose that the A node is specified to run at 02:00 and 14:00, and the B node is specified to run at 00:00 and 12:00.**



Self-dependency

If a task instance depends on the instance that is generated from the same task in the last period, you need to configure self-dependency. The following figure shows the dependencies in the situation where the A node is specified to run at 00:00 and 12:00.



16 Realtime Compute

16.1 What is Realtime Compute?

16.1.1 Background

Realtime Compute has its beginnings in the real-time big screen service of Alibaba Group during the Double 11 Shopping Festival. The big screen service allows you to view sales data during the shopping festival in real time on big screens. With five years of experience and development, the small team that once provided the real-time big screen service and limited real-time reporting services has become an independent and reliable cloud computing team. Realtime Compute provides an end-to-end cloud solution for stream processing based on years of experience in real-time computing products, architecture, and business scenarios. We strive to help more enterprises with real-time big data processing.

We previously used the open source Storm system to support the big screen service of Alibaba Group during the Double 11 Shopping Festival. We also developed stream processing code based on Storm. In these early stages, the stream processing service was provided on a small scale. Developers used Storm APIs to create jobs for stream processing. In this scenario, developers must have proficient technical skills, handle debugging challenges, and perform large amounts of repetitive work.

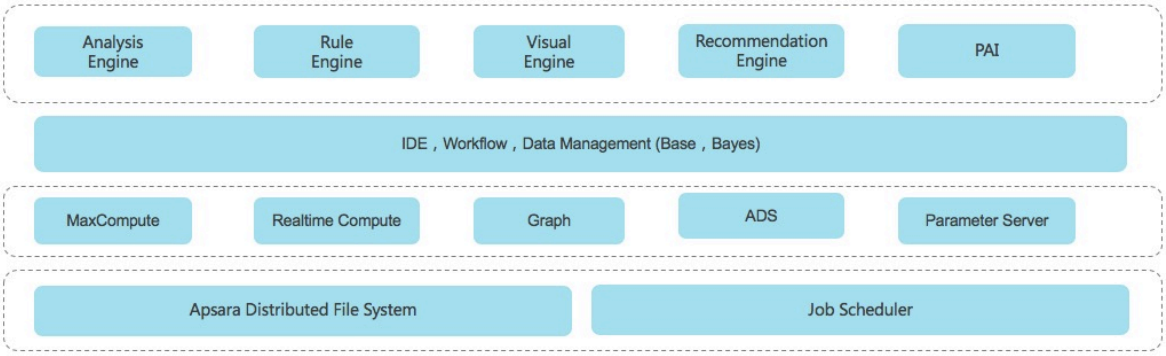
To address these challenges, we started working on data encapsulation and abstraction. Before data encapsulation and abstraction, we needed to choose an integrated processing engine for stream and batch processing from the available options: Apache Spark and Flink. The key difference of Apache Spark and Flink lies in the way they process data streams and batches. In Apache Spark, data streams are divided into micro batches, which are then processed by the Spark engine to generate the final stream of results in batches. For this method, the overhead must be increased to achieve a lower delay. Therefore, it is hard to reduce the delay of Spark Streaming to seconds or to sub-second level. In Apache Flink, batches are considered as bounded data streams that have a defined start and end. In this way, most code can be shared for stream and batch processing, which allows you to leverage the advantages of batch processing. Based on a thorough comparison

between Apache Spark and Flink, we decided to use Apache Flink as the processing engine for real-time computations over data streams. Stream processing methods can be classified as stateful computations and stateless computations. The introduction of state management allows you to easily implement complex processing logic, which is ground-breaking for stream processing.

Any emerging technology is only adopted by a small group in the beginning. With the growth of this technology and the reduction in adoption costs, it will be widely accepted. Therefore, we are working to enable stream processing technologies to be widely adopted by improving the technology and decreasing adoption costs. Apache Flink has made many improvements to the architecture, but its implementation mechanism needs to be optimized. For example, the tasks of multiple jobs may be executed by the same thread, which greatly reduces the computing performance. To resolve this issue, we introduce the YARN system. Another example is the checkpoint feature of Apache Flink. In Apache Flink, checkpoints are created to ensure data consistency, but checkpoints cannot be created when the state stored for incremental computing is excessively large. To address this challenge, Realtime Compute optimizes the checkpoint feature to efficiently manage large state. Realtime Compute has addressed many performance issues and bottlenecks to ensure the stability and scalability in the production environment. Currently, Realtime Compute is capable of supporting core businesses. We have also improved the SQL of Realtime Compute to support complex business scenarios. We are working to provide excellent user experience through constant exploration and innovation.

16.1.2 Key challenges of Realtime Compute

Realtime Compute runs on a cluster of thousands of nodes within Alibaba Group. It provides services for hundreds of real-time applications for over 20 business units of Alibaba Group, processing hundreds of billions of messages and about 1 petabyte of traffic per day. Realtime Compute has become one of the core distributed computing services of Alibaba Group.



We are working to make the following improvements:

- **Computing engine: We are working to improve the engine performance and enable the engine to support multiple semantics of processing messages.**
- **Programming interfaces: We are working to enable support for more APIs and programming languages. For example, we are working on the compatibility with open source APIs, such as Storm APIs and Beam APIs.**
- **Programming languages: We are working to enable support for more SQL syntaxes and semantics in stream analysis scenarios, such as temporal tables and complex event processing (CEP). Services: We are working to improve Realtime Compute from the following aspects: debugging, one-click deployment, hot upgrades, and training systems.**

16.2 Technical advantages

Realtime Compute uses a compute engine that is developed based on Apache Flink, which allows Realtime Compute to leverage advantages of Apache Flink and optimize the Flink Table API. You can use Flink SQL for batch and stream

processing. The application of YARN in Realtime Compute enables full compatibil ity with Flink API, which enables a large ecosystem of stream processing.

Figure 16-1: Realtime Compute and other stream processing system

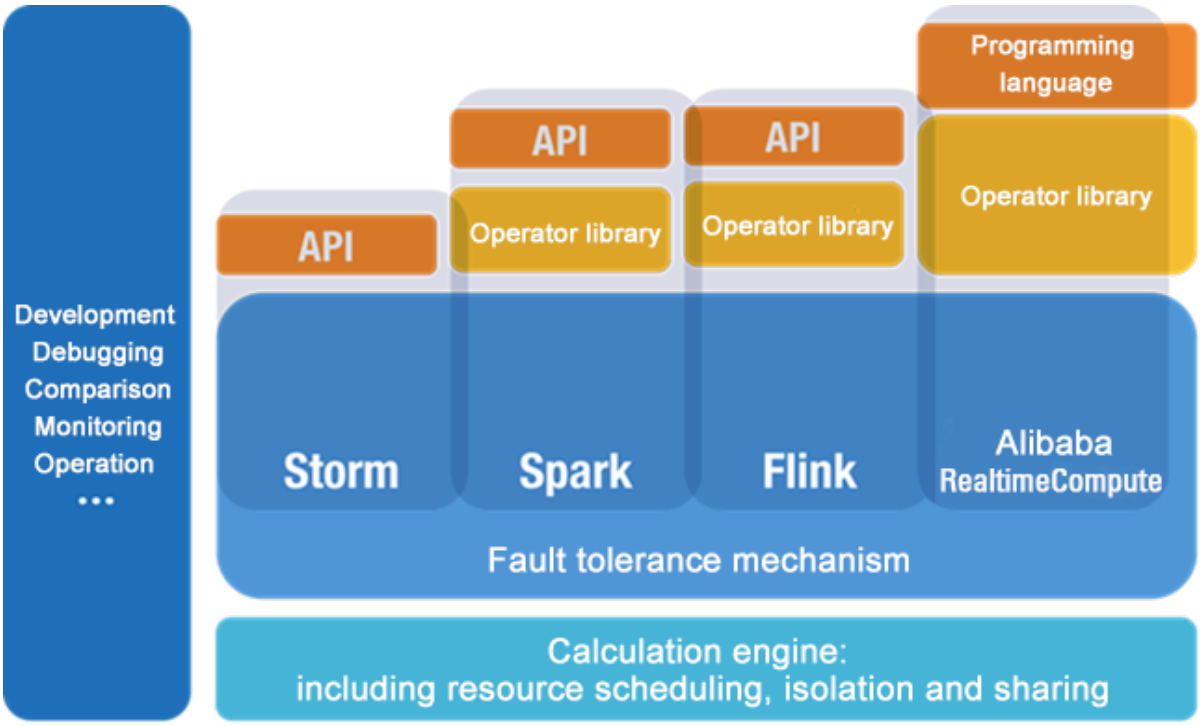


Figure 16-1: Realtime Compute and other stream processing system **shows the differences between the technologies of Realtime Compute and other stream processing systems. Based on the extensive experience of addressing challenging business scenarios, Realtime Compute provides the following benefits:**

- **Powerful stream processing functions**

  Unlike these open source systems, Realtime Compute simplifies the development process by integrating a wide range of functions. These functions are described as follows:

  - **A powerful engine is used. This engine offers the following advantages:**
    - **Provides the standard Flink SQL that enables automatic data recovery from failures. This ensures accurate data processing when failures occur.**
    - **Supports multiple types of built-in functions, such as text functions, date and time functions, and statistics functions.**
    - **Enables an accurate control over computing resources. This ensures complete isolation of each tenant's jobs.**
  - **The key performance metrics of Realtime Compute are three to four times higher than those of Apache Flink. For example, in Realtime Compute, the data processing delay is reduced to seconds or even to sub-second level. The throughput of a job reaches millions of data records per second. A cluster can contain thousands of nodes.**
  - **Realtime Compute integrates cloud-based data stores such as MaxCompute , DataHub, Log Service, ApsaraDB for RDS, and Table Store. With Realtime Compute, you can read data from and write data to these systems with the least efforts in data integration.**

- **Managed real-time computing services**

  Unlike open source or user-developed stream processing services, Realtime Compute is a fully managed stream processing engine. You can query streaming data without deploying or managing any infrastructure. With Realtime Compute , you can use streaming data processing services with a few clicks. Realtime Compute integrates services such as development, administration, monitoring , and alerting. This allows you to use cost-effective streaming data services for trial and migrate your data for deployment.

  Realtime Compute also enables complete isolation between tenants. This isolation and protection extends from the top application layer to the underlying infrastructure layer. This helps to ensure the security and privacy of your data.

- **Excellent user experience during development**

  Realtime Compute provides a standard SQL engine: Flink SQL. It also provides many built-in functions, such as the text functions, date and time functions, and statistics functions. The application of these functions greatly simplifies and accelerates the Flink-based development. With Flink SQL, even users with limited development knowledge, such as business intelligence (BI) analysts and marketers, can easily perform real-time analysis and processing of big data.

  Realtime Compute provides an end-to-end solution for stream processing, including development, administration, monitoring, and alerting. On the Realtime Compute development platform, only three steps are required to publish a job.

- **Low costs**

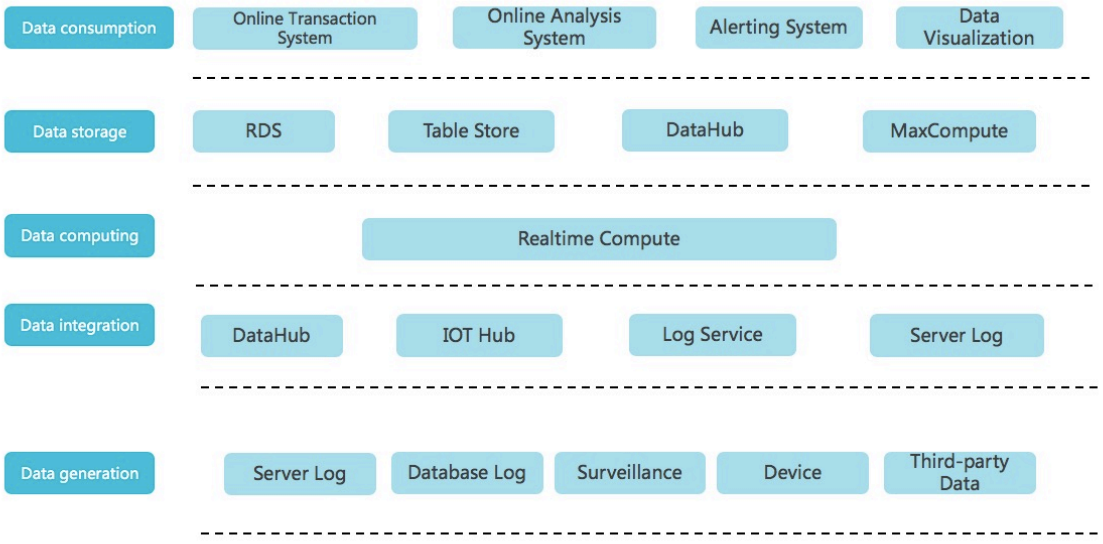
  We have made many improvements to the SQL execution engine, allowing you to create jobs more cost-effectively than to create Flink jobs. Realtime Compute is more cost-effective than open source stream frameworks in both development and production costs.

16.3 Product architecture

16.3.1 Business architecture

Realtime Compute is a lightweight SQL-enabled streaming engine for real-time processing and analysis of data streams.

Figure 16-2: Business architecture



- **Data generation**

  In this phase, streaming data is generated from sources such as server logs, database logs, sensors, and third-party systems. The generated streaming data moves on to the next phase for data integration to drive real-time computing.

- **Data integration**

  In this phase, the streaming data is integrated. You can subscribe to and publish the integrated streaming data. The following Alibaba Cloud products can be used in this phase: DataHub for big data computing, IoT Hub for connecting IoT devices, and Log Service for integrating ECS logs.

- **Data computing**

  In this phase, the streaming data, which has been subscribed to in the data integration phase, acts as inputs to drive real-time computing in Realtime Compute.

- **Data storage**

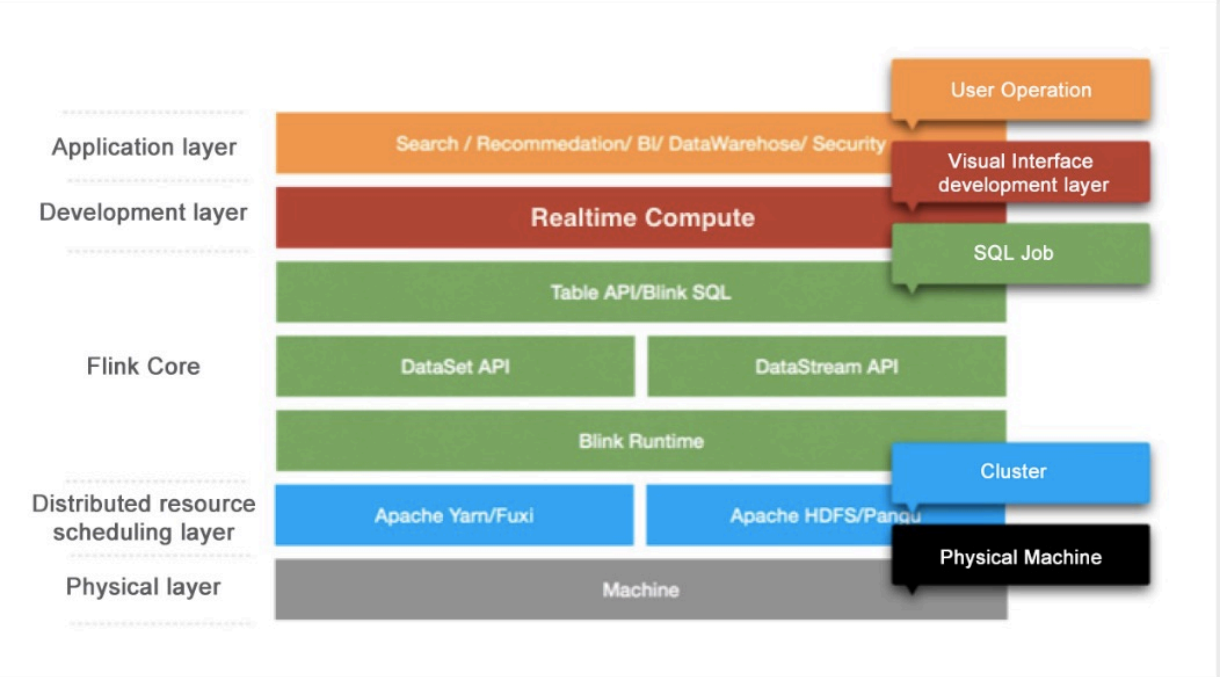
  Realtime Compute does not provide built-in data stores. Instead, it writes computing results to external data stores, such as relational databases, NoSQL databases, and online analytical processing (OLAP) systems.

- **Data consumption**

  Realtime Compute supports multiple data store types, which allows you to consume data in various ways. For example, data stores for message queues can be used to report alerts, and relational databases can be used to provide online support.

16.3.2 Technical architecture

Realtime Compute is a real-time data analysis platform for incremental computing. This platform provides statements that are similar to SQL statements and uses the MapReduceMerge (MRM) computing model for incremental computing. Realtime Compute offers a failover mechanism to ensure data accuracy when errors occur.



The Realtime Compute architecture consists of the following five layers.

- **Application layer**

  This layer allows you to create SQL files and publish jobs for real-time data processing based on a development platform. With a well-designed monitoring and alerting system, you would be notified of a processing delay for each job

in a timely manner. You can also use systems like Flink UI to view the running information of published jobs and analyze performance bottlenecks. This allows you to quickly and effectively improve job performance.

- Development layer

  This layer parses Flink SQL and generates logical and physical execution plans. The execution plans are then conceptualized as executable directed acyclic graphs (DAGs). Based on these DAGs, directed graphs that consist of various models are obtained. Directed graphs are used to implement specific business logic. A model usually contains the following three modules:

  - Map: Operations such as data filtering, distribution (GROUP), and join (MAPJOIN) are performed.
  - Reduce: Realtime Compute processes streaming data by batch, and each batch contains multiple data records.
  - Merge: You can update the state by merging the computing results of the batch, which are produced from the Reduce module, with the previous state. Checkpoints are created after N (configurable) batches have been processed . In this way, the state is stored persistently in a data store, such as Tair and Apache HBase.

- Flink Core

  This layer provides a wide range of computing models, Table API, and Flink SQL. You can use DataStream API and DataSet API at the lower sublayer. At the bottom sublayer is Flink Runtime, which schedules resources to ensure that jobs can run properly.

- Distributed resource scheduling layer

  Realtime Compute clusters run based on the Gallardo scheduling system. This system ensures that Realtime Compute runs effectively and fault tolerance is provided for recovery.

- Physical layer

  This layer provides powerful hardware devices for clusters.

16.4 Functional principles

The Blink engine of Realtime Compute is developed based on Apache Flink. For more information about the functional principles of Realtime Compute, see *Discussion on Apache Flink*.

17 Machine Learning Platform for AI

17.1 What is machine learning?

Machine learning is a process of using statistical algorithms to learn large amounts of historical data and generate an empirical model to provide business strategies.

Background

As a means of production with ever-increasing value, data has been continuously mined by developers and enterprises for valuable information. Machine learning is used to carry out this task and meets user requirements much better than traditional statistical analysis methods. Apsara Stack Machine Learning Platform for AI is developed in line with this technology trend. Machine Learning Platform for AI made its debut in 2015 as the official machine learning platform of Alibaba Cloud. Over the past three years, it has been continuously updated to provide more features, offer optimized user experience, and support more scenarios. It is among the top tier of AI cloud platforms inside and outside China.

Development status

Machine Learning Platform for AI is a proven, all-in-one platform that provides one-stop platform to implement algorithms and tasks such as data preprocess ing, feature engineering, model training, performance evaluation, and offline deployment of production applications. Machine Learning Platform for AI ranks among the top machine learning platforms domestically and internationally, and is used in a variety of sectors such as finance, energy, government, public services, customs, taxation, and the Internet.

Challenges

Machine Learning Platform for AI aims to lower the barrier to use machine learning algorithms and make AI available in each industry.

Concepts

Machine Learning Platform for AI is a set of data mining, modeling, and prediction tools. It is developed based on MaxCompute (formerly known as ODPS). Machine Learning Platform for AI supports the following functions:

- **Provides an all-in-one algorithm service covering algorithm development, sharing, model training, deployment, and monitoring.**
- **Allows you to complete the entire experiment either through the GUI or by running PAI commands. This function is intended for data miners, analysts, algorithm developers, and data explorers.**
- **In Apsara Stack, Machine Learning Platform for AI runs on MaxCompute. Machine Learning Platform for AI allows you to call algorithms to decouple the applications and compute engines after you have deployed algorithm packages in MaxCompute clusters.**
- **Provides various algorithms and reliable technical support to resolve service issues. In the Data Technology (DT) era, you can use Machine Learning Platform for AI to implement data-driven services.**

Machine Learning Platform for AI can be applied in the following scenarios:

- **Marketing: commodity recommendation, user group profiling, and precise advertising.**
- **Finance: loan delivery prediction, financial risk control, stock trend prediction, and gold price prediction.**
- **Social network sites (SNS): analysis of microblog fan leaders and social relation chains.**
- **Text: news classification, keyword extraction, document summarization, and text analysis.**
- **Unstructured data processing: image classification and image text extraction through optical character recognition (OCR).**
- **Other prediction cases: rainfall prediction and football match result prediction.**

Machine learning can be divided into three types:

- **Supervised learning: Each sample has an expected value. You can create a model and map input feature vectors to target values. Typical examples of this learning mode include regression and classification.**
- **Unsupervised learning: No samples have target values. This learning mode is used to discover potential regular patterns from data. Typical examples of this learning mode include simple clustering.**
- **Reinforcement learning: This learning mode is complex. A system constantly interacts with the external environment to obtain feedback and determines its**

own behaviors to achieve a long-term optimization of targets. Typical examples of this learning mode include AlphaGo and driverless vehicles.

17.2 Benefits

Distributed algorithm framework

- **Machine Learning Platform for AI mainly supports three engines: deep learning, parameter server, and MPI.**
- **Deep learning engine with excellent performance.**

Improved model and compilation efficiency

Collaborative optimization of models and system compilation is a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports collaborative optimization of models and system compilation.

Heterogeneous resource scheduling

For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

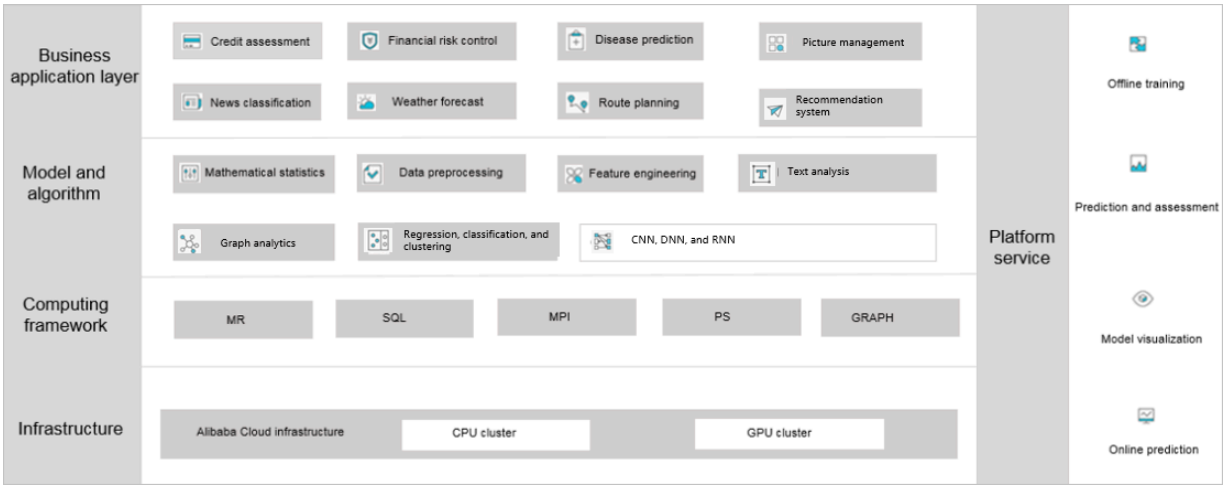
Quality algorithms

All algorithms come from the Alibaba Group algorithm system and have been tested on petabytes of service data and complex business scenarios. This ensures their sophistication and stability.

17.3 Architecture

17.3.1 System architecture

Machine Learning Platform for AI consists of multiple component systems, as shown in the following figure.



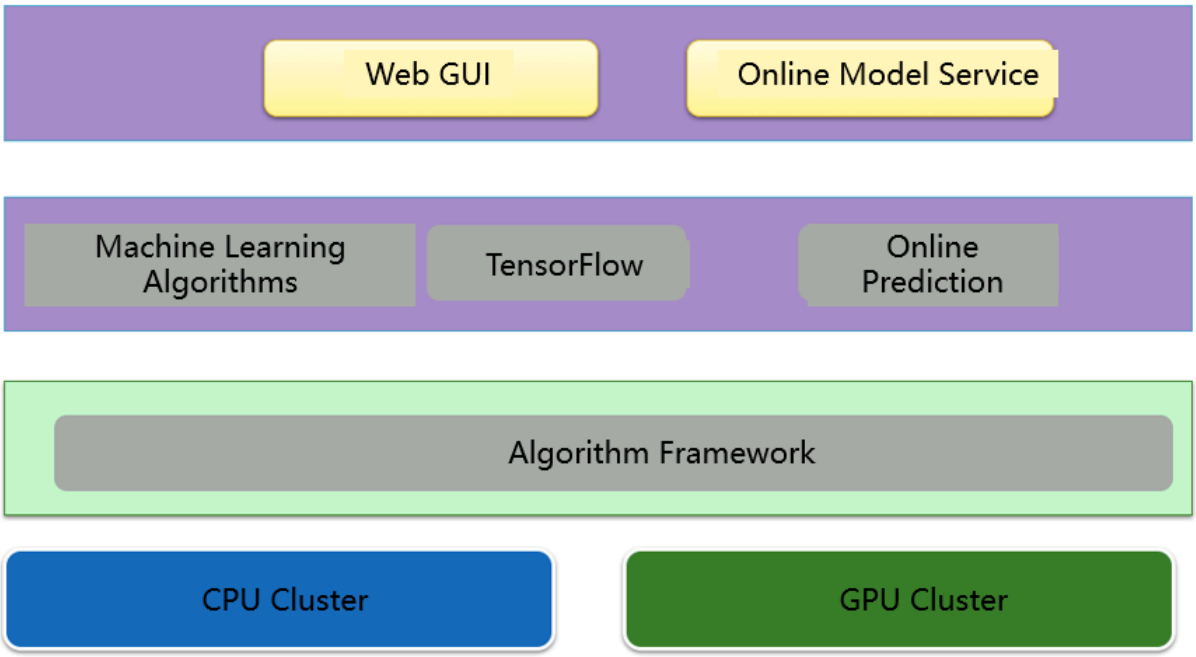
Architecture of Machine Learning Platform for AI:

- **Infrastructure layer: includes the CPU and GPU clusters.**
- **Computing framework layer: provides calculation methods such as MapReduce , SQL, and MPI. The distributed computing architecture is used to perform concurrent execution and distribution of computing tasks.**
- **Model and algorithm layer: includes basic components, such as data preprocess ing, feature engineering, and machine learning algorithms. All of the algorithm components come from the Alibaba Group algorithm system and have been tested on petabytes of service data.**
- **Service application layer: supports the search system, recommendation system , Ant Financial, and other Alibaba projects in data mining. Machine Learning Platform for AI is applicable in various industries, such as finance, medical care, education, transportation, and security.**

If you call models and algorithms in Machine Learning Platform for AI, the system converts the algorithms into compute types. For example, to join two tables, an SQL workflow is automatically generated and then delivered to MaxCompute for calculation and processing. All algorithms are stored in the underlying compute engine as plug-ins for convenient use. This decouples the algorithms from the compute engine.

17.3.2 Architecture

Machine Learning Platform for AI consists of the following components:



Component	Description
CPU cluster	The CPU cluster runs machine learning algorithms and provides computing resources such as CPU and memory resources. Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in a CPU cluster and dispatches jobs to the compute nodes.
GPU cluster	The GPU cluster runs deep learning framework jobs and provides computing resources such as GPU and graphics memory resources. · Computing resources are centrally managed by an algorithm framework. After jobs are submitted, the algorithm framework schedules compute nodes in the CPU cluster and dispatches jobs to the compute nodes. · For a task that requires multiple workers and GPUs, a virtual network is automatically created to dispatch the jobs to the compute nodes in the virtual network.
Algorithm framework	· The algorithm framework manages CPU and GPU computing resources. · The algorithm framework also provides a basic environment for running algorithms, supporting the Message Passing Interface (MPI) library, MapReduce library, Parameter Server (PS) library, algorithm package, and job isolation by user.

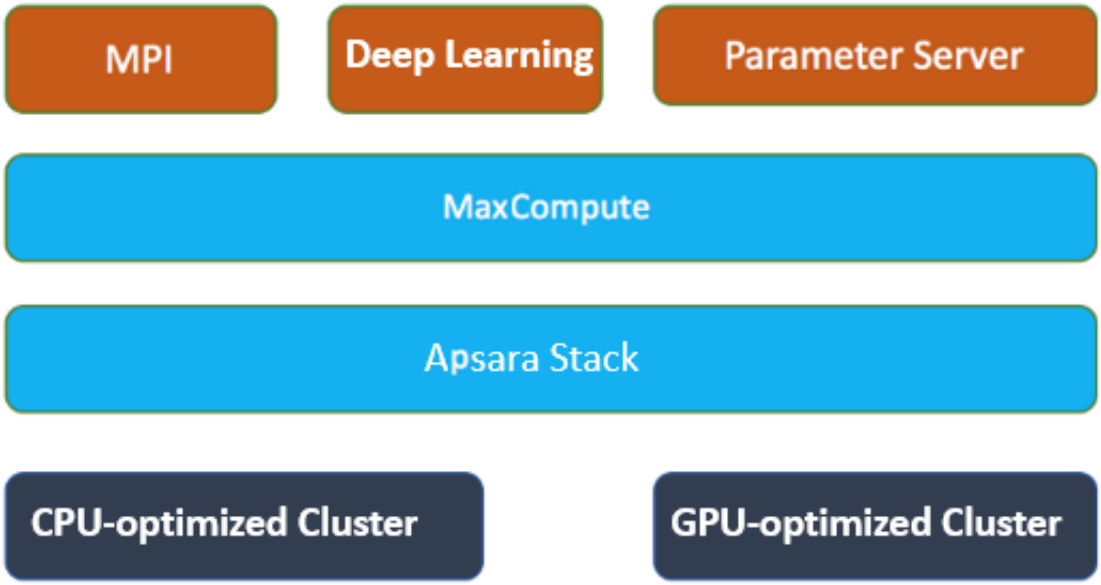
Component	Description
Machine learning algorithms	Machine learning algorithms such as data processing, classification, regression, clustering, text analysis, and network analysis are developed in the basic runtime environment of the algorithm framework. The algorithms are provided as components that can be used in experiments.
TensorFlow	· Based on the algorithm framework, the deep learning framework TensorFlow is provided. In addition, the performance and throughput of the TensorFlow open-source edition have been improved. The TensorFlow open-source 1.4 edition is supported. · You can use TensorFlow to read files from and write models to OSS buckets. · When TensorFlow is running, you can start TensorBoar d to view the status of parameter convergence during convolution.
Online model service	You can deploy machine learning models and TensorFlow-generated models as online mode services. The online model service supports model version management and blue-green deployment in the rolling upgrade mode.
Web GUI	A visual experiment management console provided by Machine Learning Platform for AI. You can perform the following actions on the Web GUI: · Create experiments, add algorithm components, and run experiments. · You can also deploy models as online model services or publish experiments to the scheduling system in DataWorks.
Call online model services	Models that are deployed as online model services provide APIs for users to call these services through the Internet.

17.4 Functions

17.4.1 Resource allocation and task scheduling

Artificial intelligent (AI) tasks typically consume considerable computing resources. Therefore, a distributed system is indispensable. A task must not occupy all resources or occupy a resource exclusively. Instead, a resource is shared by

multiple tenants. Machine Learning Platform for AI balances the efficiency of resource usage between a single task and a cluster.

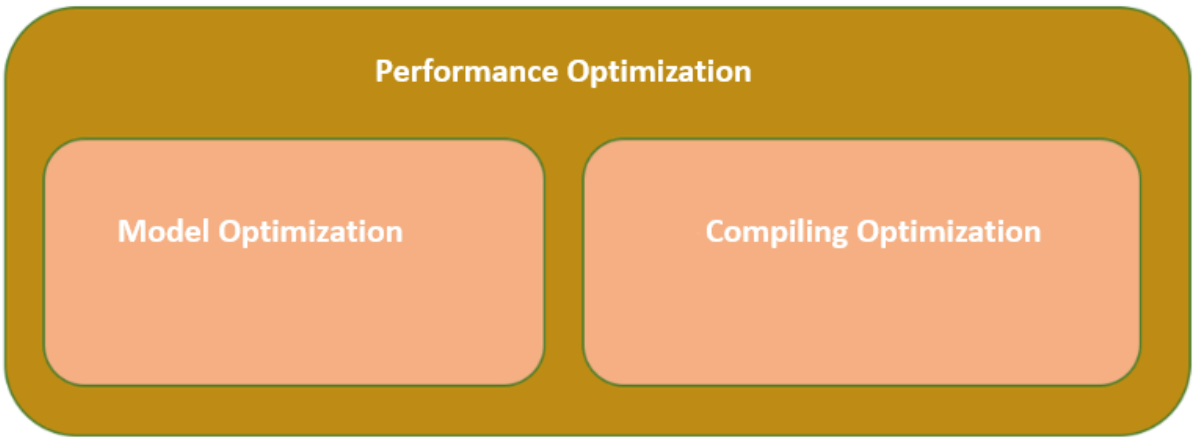


Machine Learning Platform for AI is built on the Apsara operating system and MaxCompute clusters, and is equipped with three types of compute engines: deep learning, parameter server, and MPI. AI tasks and MaxCompute tasks are deployed together to maximize the utilization of resources. For heterogeneous resources such as GPU resources required by deep learning tasks, an independent cluster is built to schedule heterogeneous computing tasks.

To allocate resources to a single task, Machine Learning Platform for AI uses the Tensorflow framework to automatically build a computing chart, allocate CPU and GPU resources, and optimize the task execution efficiency.

17.4.2 Model and compilation optimization

Collaborative optimizations of models and system compilation are a core technology provided by the modern heterogeneous computing infrastructure for AI computing services. Machine Learning Platform for AI supports the following types of optimization.



Model optimization

Many industrial service models are built based on the statistical learning theory . Model parameters can still be regularized and pruned. Besides, the AI-oriented heterogeneous computing tends to implement mixed precision to maximize the computing efficiency while guaranteeing service precision. As the hardware system develops, many technologies have been integrated in Machine Learning Platform for AI. These technologies include low bit quantization, tensor decomposition, network pruning, distillation compression, gradient compression, and hyperparam eter optimization.

Compilation optimization

Model optimization aims to minimize the computing requirements when all service requirements are met. System compilation optimization is used to adapt the specified model to the heterogeneous computing architecture and release the hardware computing resources using end-to-end optimization technologies. Compilation optimization resolves the following issues:

- **Computing requirement descriptions for service models. Machine Learning Platform for AI allows you to use advanced abstract languages to describe service models. You need only to describe the computing requirements. The system will translate the descriptions and perform automatic optimization.**
- **Hardware system independent computing chart optimization. Based on the intermediate expression of computing charts, the system implements optimizati ons that are independent of the hardware system structure. These optimizati ons include distributed splitting, mixed precision optimization, redundant computing elimination, computing mixing optimization, constant folding, efficient operator rewriting, and storage optimization of computing charts.**

- **Optimization and code generation related to the hardware system. The system performs optimization that is related to the hardware system and generates the target code. The optimization includes storage hierarchy optimization, parallel granularity reconstruction, computing and fetch streaming, assembly instructio n optimization, and automatic CodeGen space exploration and tuning.**

17.4.3 Compute engine

The compute engine provides an advanced programming language for you to compile machine learning models as needed. The engine converts the code into executable tasks at the back end, dissembles or merges the tasks, and submits the tasks to the scheduling system. Machine Learning Platform for AI supports three engines: deep learning, parameter server, and MPI.

Deep learning

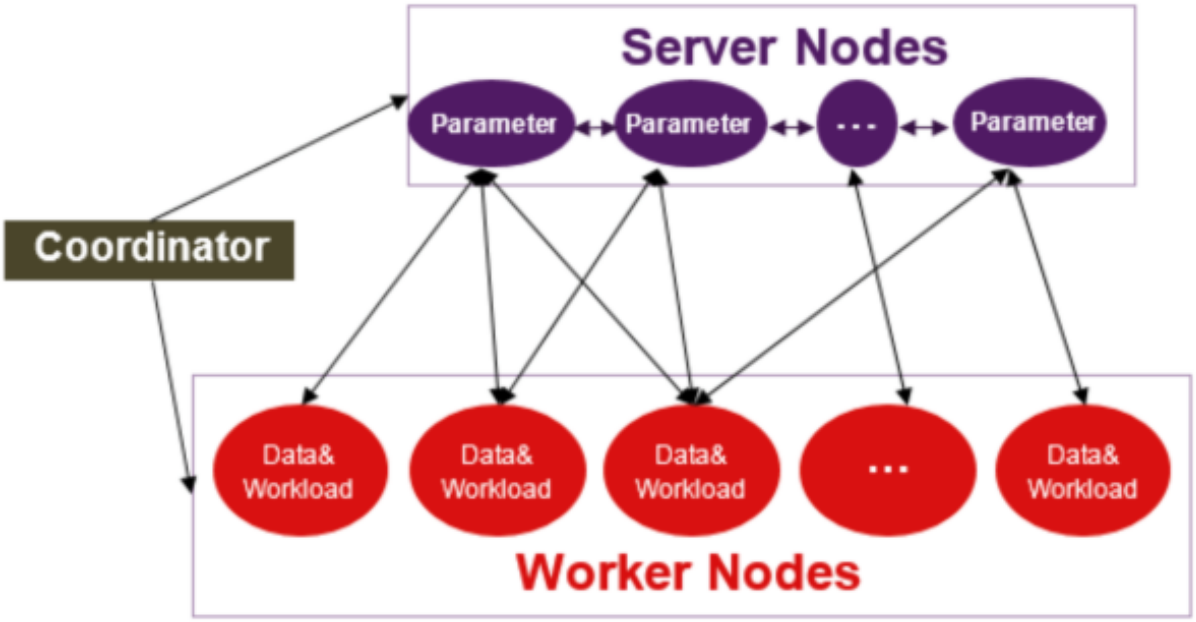
The deep learning engine is developed based on the open-source community TensorFlow. To adapt to the Apsara Stack cluster environment, the following improvements have been made to the deep learning engine:

- **Multiple basic functions are supported. These functions include image management, service resuming, permission management, and reading and writing MaxCompute and OSS data.**
- **The runtime performance of the open-source TensorFlow has been improved.**
  - **Introduces the allreduce network primitive to improve network utilization.**
  - **Replaces the native gRPC mode with the RPC framework for better performance.**
  - **Modifies the synchronization mutex mode to reduce mutex lock competition.**
- **New optimizers and operators are available.**

Parameter server

Parameter servers are a type of compute engine provided by Machine Learning Platform for AI for modeling training based on large models and large amounts of sample data. The engine allows algorithm developers to write distributed machine learning algorithm code in the same manner they write standalone code. Algorithm developers can implement distributed machine learning algorithms on the parameter server framework, and verify the algorithms based on tens of

billions of parameter and data dimensions. This shortens the development cycle and allows new algorithms to be released for big data processing.



A parameter server supports the following functions:

- Creates hash indexes for features in real time.
- Allows you to add or delete features.
- Distributed expansion.
- Globally unified checkpoint and exactly once failover.
- Sparse hash feature-based communication.
- Embedding matrix computing based on sparse hash features.

MPI

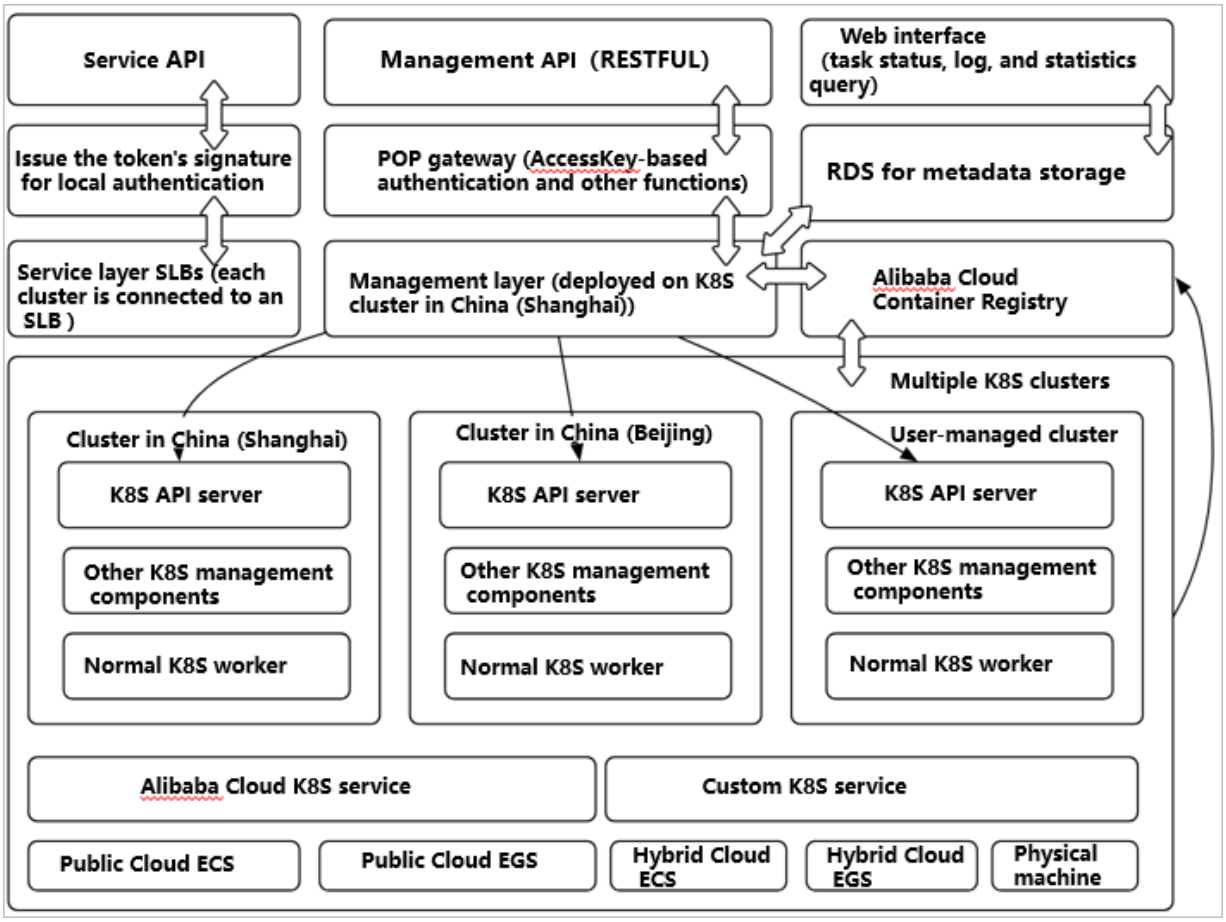
The MPI engine is a generic distributed framework used in the industry. Machine Learning Platform for AI introduces the MPI engine and integrates the MapReduce feature of MaxCompute so that you can implement classic machine learning algorithms such as logistic regression, GBDT, FM, and K-means.

17.4.4 Online prediction system

The online prediction system performs predictions tasks in the cloud using multiple types of CPUs and GPUs. The online prediction system is built based on Apsara Stack services, such as ECS, EGS, SLB, and RDS. It uses Docker to manage

resources and isolate resources. It uses open-source Kubernetes (K8s) to schedule tasks.

The overall architecture of the online prediction system is as follows:



The preceding figure shows the architecture of the online prediction system.

API layer

The online prediction service APIs are classified into two types:

- Prediction service APIs
- Prediction request APIs

The two API types are designed with different features to meet different requirements.

- Prediction service APIs are used to create, deploy, delete, and modify prediction services.
- Prediction request APIs are used to process prediction requests sent by clients and return prediction results.

Computing layer

- **Computing resources**

  All computing resources are managed by Kubernetes (K8s). Each node in a K8s cluster is an ECS instance, EGS instance, or physical server.

- **Failover**

  Failover depends on the failover solution provided by K8s. K8s allows you to configure a listening service for each container. For example, when the listening port is set to port 80:

  - **If an IRP error occurs:**

    1. **Port 80 has failed the health check. K8s sets the status of the pod (container group) to Unavailable. Traffic is forwarded to another pod.**
    2. **When a pod is restarted, the framework initializes the pod and loads the model. Port 80 is not enabled until the model has been loaded.**
    3. **Port 80 is enabled after the initialization is completed. After port 80 passes the health check at the scheduling layer, the pod is added to the traffic pool again.**

  - **If a node in a K8s cluster fails: The keepalive message exchange between the K8s primary node and the failed node fails. K8s sets the status of the failed node to Not Ready. The pod (container group) running on the node is migrated to another node. The traffic is also forwarded to another node.**

- **Rolling update**

  Rolling updates indicate application updates with zero downtime. The updates are classified into two types:

  - **User data update: User data about the model and processor is updated using an API provided by the online prediction service. The back end creates a new image version based on the current image version and updates the deployment.**
  - **IRP framework code update: The framework code is updated by creating a procedure to update all user tasks in the back end. Framework code update**

follows the rolling update procedure. Users are not aware of the update procedure.

User data and framework code are decoupled during cluster scheduling and they can be updated separately. The system packages user data and framework code into images of later versions separately and then modifies the description file of the existing application deployment. K8s performs rolling updates for running pods and dynamically switches the traffic to ensure that users are unaware of ongoing updates.

17.4.5 List of functions by module

Machine Learning Platform for AI provides a complete workflow of machine learning, such as data uploading, data processing, data visualization, model training, model deployment, model evaluation, and model utilization.

The following table describes the modules and corresponding functions.

Module	Function	Description
Data control	Data uploading	You can upload data through Machine Learning Platform for AI. When you upload data, the data is parsed, verified , and any errors are recorded and reported.
	Data table displaying	On Apsara Stack Machine Learning Platform for AI, click Data Source in the left-side navigation pane to view the uploaded data tables. You can enter a data table name in the search box and click the search icon to search for a data table. Fuzzy search is also supported.
	Data visualization	Right-click a component and choose View Data from the shortcut menu to view data in histograms, pie charts, or line charts.
Model control	Model training	On Machine Learning Platform for AI, click Run in the upper section of the canvas to train and generate a model.

Module	Function	Description
	Model visualization	On Machine Learning Platform for AI, click Models in the left-side navigation pane. Right-click a model and choose Show Model from the shortcut menu to view model parameters. Tree models and linear models can be displayed in tables.
	Model downloading	Right-click a model and choose Export PMML from the shortcut menu to generate and download a PMML file. A PMML is a standard model description file which can be parsed by a variety of open-source software.
	Model-based prediction	You can connect model generation components and prediction components . The system will automatically use the generated model for prediction.
	Model addition, deletion, modification, and query	Right-click a model and choose to add, delete, modify, or query a model.
	Online model service	You can use the online model service to deploy a model and call the correspond ing RESTful API for online prediction.
	DataWorks task scheduling	You can deploy experiments to DataStudio as DataWorks tasks and configure the system to periodically run the tasks.
	Model evaluation	You can evaluate models using confusion matrix, binary classification evaluation, clustering model evaluation , and regression model evaluation. Models are evaluated based on metrics such as F1 score, AUC, and KS. All evaluation results can be viewed in tables or charts.
Experiment control	Whole experiment lifecycle control	You can add, delete, modify, query, and copy experiments.
	Experiment visualization	Animated visualizations are used to display the entire procedure by which an experiment runs.

Module	Function	Description
	Notifications	The status of a running experiment is displayed in a prompt in the upper-right corner of the canvas, such as success and error messages.
Deep learning	Multiple deep learning frameworks	Three mainstream deep learning frameworks are supported: TensorFlow , Caffe, and MXNet. With many underlying optimizations, TensorFlow delivers better performance than other open-source frameworks.
	TensorBoard	You can view the training status of each layer in a TensorBoard job in real time and display the results visually.
	Automatic authorization	When the data source of a TensorFlow project is set to OSS, you must obtain permissions on OSS before you can run an experiment. Machine Learning Platform for AI supports automatic authorization, allowing you to obtain the read and write permissions on OSS with a single click.
	Visualized TensorFlow execution settings	The TensorFlow component is added to provide related data source settings , allowing you to run the component visually. On the Tuning tab, you can specify the number of GPUs to run with and implement parallel training with multiple GPUs easily.
	Scheduling	Deep learning jobs can be deployed and periodically executed in DataWorks.
Dashboard	Experiment history chart	You can view the experiment history on the dashboard page.
	Running experiments	You can view running experiments or delete a running experiment to save resources.
	Scheduled tasks	You can view scheduled tasks that have been deployed and add, delete, modify, and query tasks through DataStudio.

Module	Function	Description
Templates on the homepage	Machine Learning Platform for AI provides many built-in experiment templates	The experiment templates can be used for a wide range of scenarios such as product recommendations, news classification, financial risk control, haze prediction, heart disease prediction, agricultural loan delivery, and census. All these cases contain complete data sets and instructions about their use. You can also create your own experiments by using these templates.
Online prediction	Model version management	You can upload multiple versions of a model, configure them to share the same resources, and switch between those versions.
	Blue-green model deployment	The blue-green model deployment function allows you to dynamically change the proportions of the traffic forwarded between different versions of a model.
	Online model debugging	The online debugging function of Machine Learning Platform for AI allows you to debug deployed models online and view the debugging results in real time.

17.5 System metrics

Metric	Requirement
Core metrics	Provides typical machine learning algorithms, such as the data preprocessing, feature engineering, statistical analysis, classification, regression, and clustering: · Provides model evaluation algorithms. · Provides time series, text analysis, and network analysis algorithms. · Provides deep learning frameworks such as TensorFlow. · Provides the GPU job scheduling capability. · Provides the online model service and allows you to directly deploy models to the online model service. · Provides a visual console to help you use visual components to create experiments.
Function metrics	Supports reading structured and unstructured data.
	· Supports data sampling and filtering algorithms, such as the random sampling, weighted sampling, and stratified sampling. · Supports data merging algorithms, such as JOIN, UNION, and MERGE. · Supports data preprocessing algorithms, such as splitting , normalization, standardization, KV to Table, Table to KV, and adding ID columns to tables.
	· Supports the principal component analysis (PCA) algorithm. · Supports feature importance evaluation for linear and random forest models.
	Supports the following statistical analysis algorithms: the covariance, empirical probability density chart, whole table statistics, chi-square goodness of fit test, chi-square test of independence, scatter plot, correlation coefficient matrix , two sample T test, single sample T test, normality test, percentile, Pearson coefficient, and histogram.

Metric	Requirement
	· Supports the following binary classification algorithms : the Gradient Boosting Decision Tree (GBDT), Linear Support Vector Machine (SVM), and logistic regression. · Supports the following multiclass classification algorithms : K-nearest neighbors (KNN), multiclass classification for logistic regression, random forest, and naive Bayes. · Supports the GBDT, linear regression, PS-SMART regression, and PS linear regression algorithms. · Supports K-means clustering. · Supports the following evaluation algorithms: the binary classification model, regression model, clustering model, multiclass classification, and confusion matrix.
	· Supports the deep learning framework TensorFlow. · Supports TensorBoard. · Supports scheduling a deep-learning job to a GPU server.
	Supports time series algorithms such as x13_arima and x13_auto_arima.
	Supports the following text algorithms: the word frequency statistics, TF-IDF, parallel latent dirichlet allocation (PLDA), Word2Vec, word splitting, converting rows, columns, and values to KV pairs, string similarity, deprecated word filtering, text summarization, document similarity, sentence splitting, keyword extraction, ngram-count, semantic vector distance, and pointwise mutual information (PMI).
	Supports the following network analysis algorithms: the K-Core, single-source shortest path, page rank, label propagation clustering, label propagation classification , modularity, maximum connected subgraph, vertex clustering coefficient, edge clustering coefficient, counting triangle, and tree depth.
	· Supports the online model service and allows you to deploy machine learning algorithm models or deep-learning models to the service. · Provides an HTTP-based API.

Metric	Requirement
	· Provides the Web-based visual editor, which allows you to create an experiment by dragging and dropping components. · Supports releasing experiments to DataWorks for task scheduling. · Supports experiment and model management.
Compatibility/ openness	· Supports the open-source deep learning framework TensorFlow 1.4. · Supports exporting the PMML file from machine learning models. · Supports the online service model and allows you to deploy the model as an API.

18 Quick BI

18.1 What is Quick BI?

Background

With the rapid development of IoT, the volume of data is skyrocketing. It has become increasingly important to analyze and use data to generate commercial value. Data analysts urgently need an efficient data analysis tool. With Quick BI, everyone can be a data analyst. Quick BI provides online ad hoc analysis, drag-and-drop operations, and data visualization, helping you easily analyze data and monitor business. Quick BI is a tool to view data for business personnel and a booster for digital operations, solving *the last mile problem* **of big data applications.**

Status quo

Today, more and more enterprises are migrating data to the cloud. However, governments and financial institutions build local databases for security concerns. As a result, enterprise data is in distributed storage. Quick BI connects to various data sources and centrally schedules datasets to meet different requirements of on-cloud and local data use. Quick BI can connect to the following data sources:

- **MaxCompute (formerly ODPS) and ApsaraDB for RDS**
- **User-created MySQL and SQL Server databases that are hosted on ECS**
- **Data sources in VPC networks**

Challenges

Quick BI faces the following challenges:

- **How to provide better service and improve customer experience**
- **How to accelerate the construction of traditional BI and big data systems at low costs**
- **How to improve business insights with effective business monitoring and related business data analytics**

18.2 Benefits

The benefits of Quick BI can be summarized as quick response, powerful capabilities, high security, visualization, and user-friendliness.

Rapid data modeling

You can create a dataset with a few clicks, significantly reducing your reliance on professional staff.

Powerful data analysis

Quick BI generates professional workbooks that allow you to jointly analyze data online and generate reports, such as daily, weekly, and monthly reports. More than 300 regular data analysis functions allow you to easily acquire business analysis results.

Reliable data access control

Quick BI adopts an ACL system. An access object is used as a control unit for permission approval and authorization. Quick BI also incorporates a row-level permission control solution to realize more fine-grained data access control.

Multiple options for data visualization

Quick BI provides over 30 components, helping you achieve effective data visualization. In addition, it supports multi-terminal adaptation, allowing you to access multiple terminals with a single operation, which significantly improves data analysis efficiency.

Multi-user collaboration

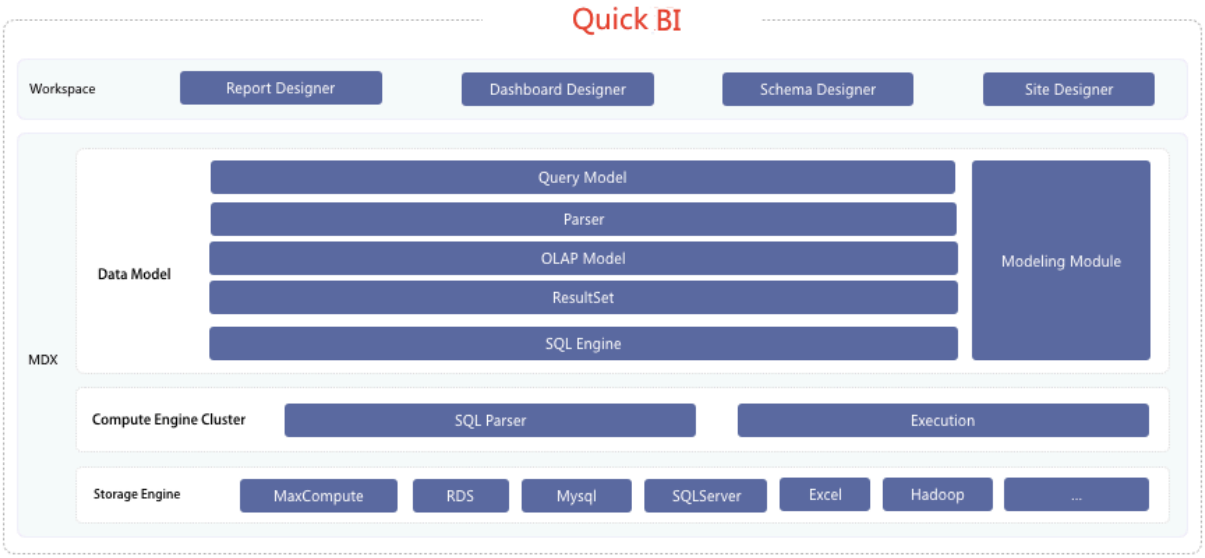
All objects are online. Enterprise users organize their business in a shared workspace. Quick BI allows members in a workspace to operate and analyze data collaboratively.

18.3 Product architecture

18.3.1 System architecture

This topic describes the system architecture of Quick BI.

The following figure shows the system architecture of Quick BI.



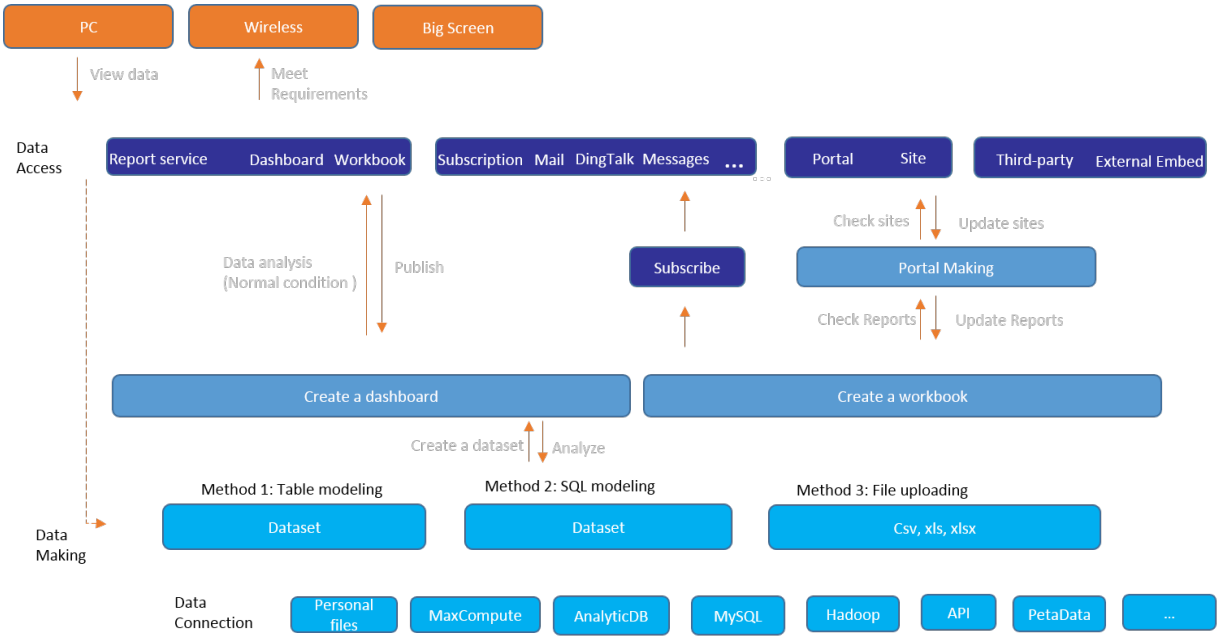
The system architecture of Quick BI consists of four function modules, including data connection, computing engine, data management, and data analysis.

- **Data connection: establishes a connection with a database to achieve ad hoc data analysis.**
- **Computing engine: efficiently generates SQL statements based on constructed datasets and user-defined conditions and executes them in a database.**
- **Data management: creates datasets that support multi-dimensional analysis based on tables, SQL statements, and local files, providing dataset management and join operations.**
- **Data analysis: provides dashboards and workbooks to support enterprise data analysis and ad hoc query.**

18.3.2 Components

This topic describes the components of Quick BI.

The following figure shows the topology of Quick BI.



Data sources

Quick BI can read data from a variety of data sources, such as relational databases , MaxCompute, and local files. All data is stored in data sources, and Quick BI does not copy data from the data sources.

Datasets

A dataset is the smallest object for data analysis of Quick BI. It can be data of a subject or data of some scenario components.

Quick BI datasets support join operations, allowing you to increase the number of dimensions or measures. For example, you can associate a dataset with another to analyze transactions. The process requires no SQL statements, helping you easily build data objects with powerful functionality.

Quick BI supports multiple granularities in the data dimension, including daily, weekly, monthly, quarterly, yearly, month-to-date, quarter-to-date, and year-to-date.

Dashboards

Quick BI dashboards provide data visualization functions, and support multicomponent filter interaction and a wide range of component settings.

Dashboards support mainstream data analysis components, such as vertical bar charts, maps, and funnel charts. They also provide a wide range of function components, such as filter bar, iFrame, and tab. Quick BI allows you to design

beautiful dashboards with flexible components, reducing your business operation costs and your reliance on specialized staff.

Workbooks

Workbooks are a unique feature of Quick BI. They offer online data analysis in a way similar to spreadsheets. A cell in a workbook is a data unit. Quick BI workbooks allow you to copy data locally and to retrieve datasets. You can link cells to each other. In addition, workbooks provide the following features:

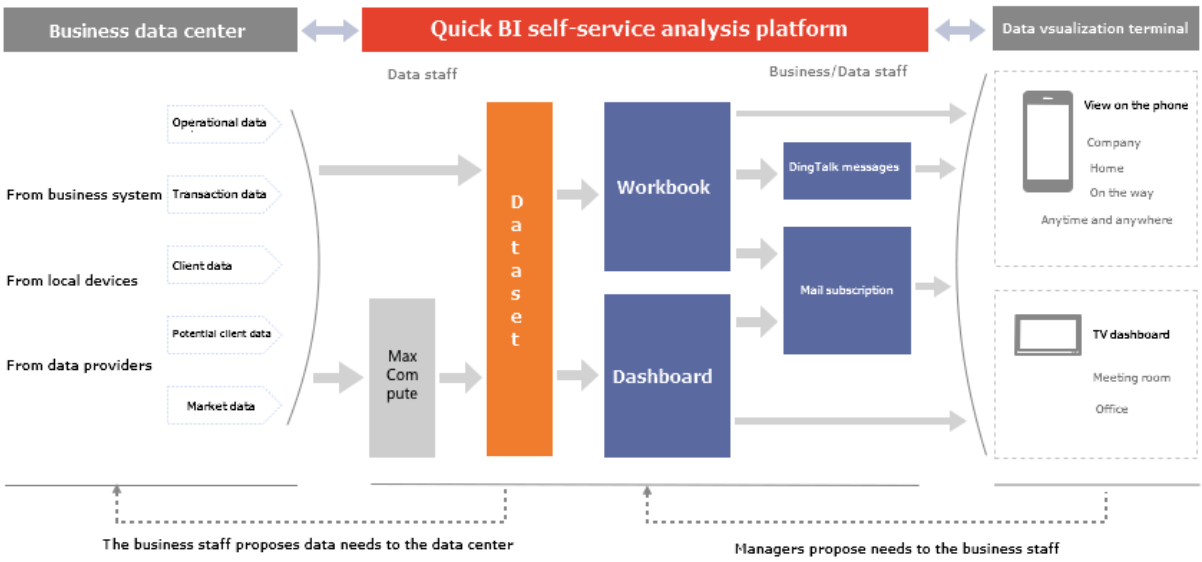
- **Built-in formulas, data aggregation, over 300 functions, and cross-sheet references**
- **Interactive drag-and-drop operations with row and column freezing**
- **Visualization of analyzed data**
- **View, edit, and preview modes**

BI portals

A BI portal is a set of dashboards and workbooks organized in the form of menus based on different scenarios, departments, and marketing plans. This allows you to have a complete view of various business states.

Monitoring dashboard

The monitoring dashboard enables you to view data. You can use the single-screen dashboard of Quick BI to create reports and display them on your TV monitor and other terminals in full-screen mode. This allows you to view the business status at a single glance.



18.3.3 Deployment

Quick BI is automatically deployed in the Apsara Infrastructure Management Framework console.

The following table lists resources required by Quick BI deployment.

Server role	Specification	Description	Scalability
Quick BI server	Two Docker containers. Each container has 16 GB memory and 8 cores.	Console page	Scale-out
Quick BI agent	Two Docker containers. Each container has 16 GB memory and 8 cores.	Computing server	Scale-out
Redis primary node	One Docker container. It has 16 GB memory and 8 cores.	Cache	Scale-out
Redis secondary node	Two Docker containers. Each container has 16 GB memory and 8 cores.	Cache	Scale-out
dbinit	One Docker container. It has 1 GB memory and one core.	Metadata initialization	None
test	One Docker container. It has 1 GB memory and one core.	Periodical monitoring system	None

18.3.4 Server roles

This topic describes server roles of Quick BI.

Quick BI consists of the following server roles:

- **base-biz-yunbi-dbinit: a database initialization component that initializes basic metadata of Quick BI. It is a prerequisite for Quick BI to run properly. The component must be at desired state.**
- **quickbi-redis-slave: the secondary node of Redis. It provides the data caching function for Quick BI to improve system query performance.**
- **quickbi-redis-master: the primary node of Redis. It provides the data caching function for Quick BI to improve system query performance.**
- **base-biz-yunbi-executor: an executor component that queries the table metadata in the connected data source and queries the table data.**
- **base-biz-yunbi: a web service component that provides web services and allows users to visit Quick BI web pages.**
- **ServiceTest: an automated testing component that executes test cases to test the overall service availability of Quick BI.**

18.4 Features

This topic describes the features of Quick BI.

Quick BI provides the following features:

- **Supports a wide range of data sources, such as MaxCompute, relational databases, and local files.**
- **Quickly analyzes offline data sources. For example, Quick BI can analyze a 100 GB file in 10 seconds.**
- **Provides complete workbooks to enable you to easily make complex reports.**
- **Enables everyone to easily learn and use.**
- **Provides you with diverse options for data visualization, and automatically identifies data properties to generate the most appropriate charts.**
- **Provides strict permission control and adopts multi-layer verification to ensure data security.**
- **Provides an OLAP analysis engine that has comprehensive functions and is easy to use.**
- **Supports collaborative operations, allowing multiple users to analyze data cooperatively.**

19 Apsara Big Data Manager (ABM)

19.1 What is Apsara Big Data Manager?

Background

In the daily use of the Alibaba Cloud big data platform, O&M engineers and data service developers often need to manage various big data products, including offline computing engines, real-time computing engines, analytic and query engines, AI platforms, and big data applications. These big data products are sometimes closely related to each other. To improve O&M and development efficiency, a one-stop O&M platform is needed to integrate these products. Apsara Big Data Manager (ABM) is developed against this background.

Supported products

Currently, ABM supports operations and maintenance of the following big data products:

- **MaxCompute**
- **DataWorks**
- **StreamCompute**
- **Quick BI**
- **DataHub**
- **Machine Learning Platform for AI**

ABM supports operations and maintenance from perspectives such as business, services, clusters, and hosts. You can also install patches for big data products, customize alert configurations, and view O&M history through the ABM console.

ABM allows on-site Apsara Stack engineers to manage big data products with ease. For example, they can view performance metrics in real time, modify runtime configurations, and check and handle alerts in a timely manner.

Challenges

The stability of a big data service may be adversely affected by an unstable product platform, and also poor service implementations, such as slow or bad SQL queries, data skews, and long tail queries. O&M engineers alone are not able to ensure

service stability. Instead, service developers and O&M engineers must work together to troubleshoot issues and improve stability.

In addition to an operations and maintenance platform for O&M engineers, ABM is evolving to provide data-based and intelligent O&M capabilities. It is exploring ways to implement cross-computing engine management and add operations and maintenance support for more products, such as big data applications, computing engines, scheduling systems, storage, operating systems, and networks.

19.2 Benefits

Based on a mature O&M mid-end, ABM can quickly connect to big data products and provide comprehensive O&M capabilities for each product.

O&M mid-end

With the O&M mid-end, ABM provides various built-in services and SDKs to enable all-around operations and maintenance capabilities. Each product can easily connect to ABM and has an exclusive site to implement operations and maintenanc e. Compared with the traditional development process, ABM provides a more visualized, configuration-based, and function-based alternative and minimizes the development costs of business customization.

The O&M mid-end provides the following services in Apsara Stack:

- Job platform: supports visualized job management, execution, and scheduling. This satisfies various needs of visualized O&M.
- Knowledge graph: supports data storage, integration, and query in different scenarios. This resolves the difficulties in integrating and querying dispersed data.
- Function as a service (FaaS): supports low-cost trial and error, fast code development, and function-based business logic management. This relieves users from complex project organization, dependency management, deployment , and scaling and allows them to focus on business.
- Application management: stores business logic and configurations in a hierarchical way, and supports highly flexible extensions. This allows users to create complex application structures with simple configurations by using JSON.

- **Inspection service: provides a universal solution for checker management and scheduling, and supports disparate alert data sources. The inspection service can be embedded into any page of an application site.**
- **Third-party system adaptation: allows users to use one SDK to call APIs of all connected third-party systems.**
- **Authorization proxy: adapts to AAS and OAM in Apsara Stack, provides capabilities such as visualized user management and authorization management, and satisfies the authorization and authentication requirements of third-party systems.**
- **Gateway service: integrates all service APIs so that external systems can call these APIs uniformly. In addition, isolation, decoupling, and scaffold capabilities are provided for authenticating and processing all requests centrally.**
- **Apsara Infrastructure Management Framework synchronization: adapts to the Apsara Infrastructure Management Framework base in Apsara Stack, and provides encapsulated interfaces for querying and managing all host data.**
- **Tunnel service: uses StarAgent to shield the differences of underlying command execution tunnels and provides a universal interface. This allows users to deliver commands and files to a large number of hosts, and aggregate and query the statuses of these hosts.**

Quick service development

Based on the O&M mid-end, ABM now supports multiple products and provides stable and reliable operations and maintenance capabilities for them. Supported products include MaxCompute and DataWorks.

19.3 Architecture

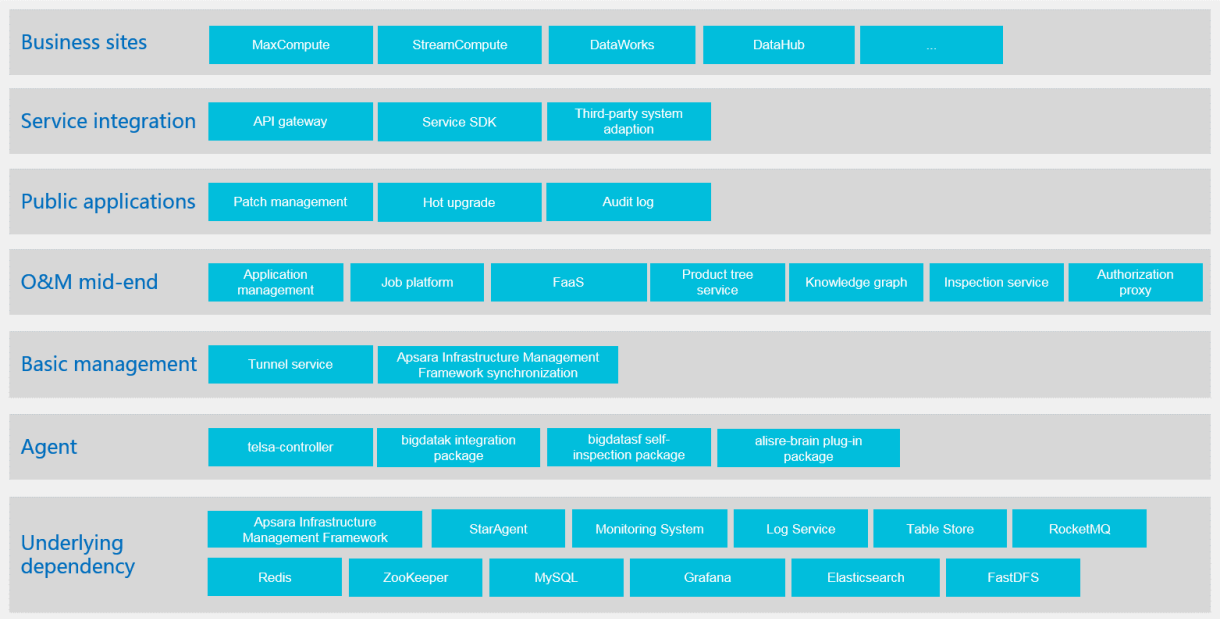
19.3.1 System architecture

This topic describes the system architecture of Apsara Big Data Manager (ABM) and the functions of each component.

ABM uses a microservice architecture that supports data integration, interface integration, and feature integration through a unified platform, and provides standard service interfaces. This architecture enables a consistent user interface , which means that O&M operations are the same for all products. This reduces training costs and lowers O&M risks.

The ABM system consists of the following components: underlying dependency, agent, basic management, O&M mid-end, public applications, service integration, and business sites.

Figure 19-1: Architecture



Underlying dependency

ABM depends on open source systems from Alibaba and third parties.

- **Uses StarAgent and Monitoring System of Alibaba to run remote commands and remote data collection instructions.**
- **Uses ZooKeeper to coordinate primary and secondary services. This guarantees high availability of services.**
- **Uses RDS to store metadata, Redis to store cache data, and Table Store to store large amounts of self-test data. This improves service throughput.**

Agent

The agent provides client SDKs, scripts, and monitoring packages to be deployed on managed servers.

O&M mid-end and basic management

The O&M mid-end and basic management components form the base of the ABM system. Each service in these two components provides different capabilities for business sites. This enables quick construction of business sites and makes the capabilities of each business site complete.

Public applications

Public applications are developed based on the O&M mid-end and designed with special purposes. These applications are adaptive to all big data products supported by ABM.

Service integration

Service integration links business sites with underlying components. It integrates interfaces of all internal services, adapts to various third-party systems, and provides a unified SDK for users.

Business sites

Business sites are built based on the O&M mid-end and cover all big data products, including MaxCompute, StreamCompute, DataWorks, and DataHub. Each business site provides comprehensive O&M capabilities for one product.

19.4 Features

19.4.1 Small file merging

This topic describes the small file merging feature of ABM for MaxCompute.

What are small files

Apsara Distributed File System stores data in blocks. The size of each block is 64 MB. Small files in this topic refer to files whose size is less than 64 MB. Reduce computing or real-time data collection through tunnels will generate a large number of small files.

Impacts of small files

- **More small files consume more instance resources. In MaxCompute, a single task instance can handle up to 120 small files. Therefore, too many small files cause a resource waste and deteriorate system performance.**
- **Too many small files cause high pressure on Apsara Distributed File System, and decrease the utilization rate of disk space.**
- **Too many small files occupy a large amount of memories of Master servers and Chunkservers in Apsara Distributed File System. When the memory usage exceeds 50% of the safety limit on a Master server of Apsara Distributed File System, the cluster stability is affected.**

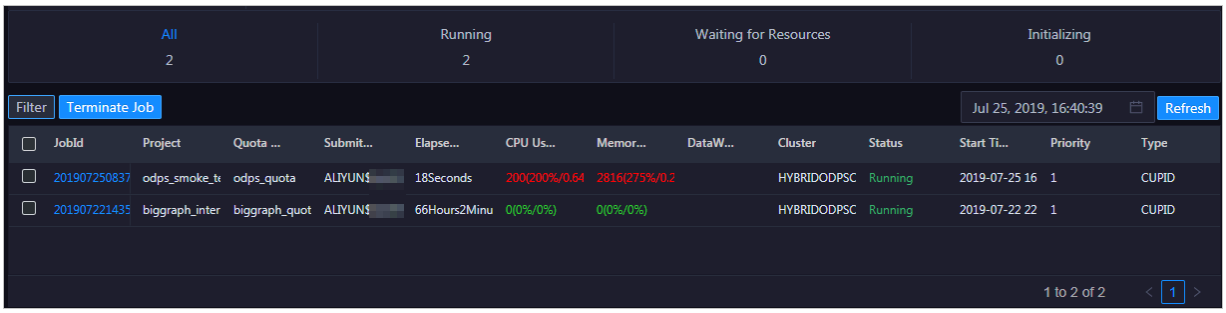
Method of merging small files

ABM uses the MaxCompute SDK to generate merge tasks for merging small files. This method increases merging concurrency to the maximum extent. Currently , you can create merge tasks by cluster or project. You can configure whether to allow merge tasks to run concurrently and specify the start and end time for each merge task.

19.4.2 Job snapshot

This topic describes the job snapshot feature of ABM for MaxCompute.

In this topic, all jobs refer to MaxCompute jobs. When a job is executed, ABM saves detailed job logs. These logs are used to generate a job snapshot. The following figure shows an example of the job snapshot page.



The job snapshot feature supports the following functions:

- **Displays information about current and historical jobs, including the resource usage and queuing status.**
- **Supports aggregating jobs from different dimensions, such as the quota group , submitter, and job status. This allows you to clearly understand the status of current jobs.**
- **Supports generating a detailed Logview page for a single job.**
- **Supports terminating jobs.**