Application Performance Management 2.0

User Guide

Issue	01
Date	2024-07-04

HUAWEI CLOUD COMPUTING TECHNOLOGIES CO., LTD.

Trademarks and Permissions


and other Huawei trademarks are the property of Huawei Technologies Co., Ltd.
All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

The purchased products, services and features are stipulated by the contract made between Huawei Cloud and the customer. All or part of the products, services and features described in this document may not be within the purchase scope or the usage scope. Unless otherwise specified in the contract, all statements, information, and recommendations in this document are provided "AS IS" without warranties, guarantees or representations of any kind, either express or implied.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Contents

1 Service Overview

1.1 What Is APM

O&M Challenges

In the cloud era, applications in the microservice architecture are increasingly diversified, bringing many application exceptions. Application O&M faces the following challenges:

- Distributed applications have complex relationships. As a result, it is hard to ensure normal application running, and quickly locate faults and performance bottlenecks.
- Users choose to leave due to poor experience. If O&M personnel cannot detect and trace services with poor experience in real time, or diagnose application exceptions in a timely manner, user experience will be greatly affected.
- There are a large number of widely distributed applications in the service system. Calls across systems, regions, and applications are frequent. Enterprises urgently need to reduce application management and O&M costs and improve O&M efficiency.

Introduction to APM

Application Performance Management (APM) helps O&M personnel quickly identify application performance bottlenecks and locate root causes of faults, ensuring user experience.

You only need to install Agents for applications so that APM can monitor them in an all-round manner. APM can quickly locate error APIs and slow APIs, reproduce calling parameters, and detect system bottlenecks, facilitating online diagnosis. Currently, APM supports Java applications. The following table lists the application monitoring capabilities of APM.

Table 1-1 APM monitoring capabilities

Capability	Description
Non-intrusive collection of application performance data	You do not need to modify application code. Instead, you only need to deploy an APM Agent package and modify application startup parameters to monitor applications.
Application metric monitoring	APM automatically monitors application metrics, such as JVM, JavaMethod, URL, Exception, Tomcat, HttpClient, MySQL, Redis, and Kafka.
Application topology	APM automatically generates call relationships between distributed applications based on dynamic analysis and intelligent computing of remote procedure call (RPC) information.
Tracing	After multiple applications are connected to APM, APM automatically samples requests, and collects the call relationships between services and the health status of intermediate calls for automatic tracing.
Metric drill-down analysis	APM enables you to drill down and analyze metrics such as application response time, number of requests, and error rate, and view metrics by application, component, environment, database, middleware, or other dimensions.
Error or slow URL tracing	APM identifies error or slow URLs based on URL tracing, and automatically associates them with corresponding APIs, such as SQL and MQ APIs.

1. Access to APM: Applications need to implement AK/SK authentication to connect to APM.
2. O&M data collection: APM can collect data about applications, basic resources, and user experience from Agents in non-intrusive mode.
3. Service implementation: APM supports application metric monitoring, application topology, tracing, and intelligent alarm reporting.
4. Service expansion:
   - You can quickly diagnose application performance exceptions based on the application topology and tracing of APM, and make judgments based on the application O&M metrics of Application Operations Management (AOM).
   - After identifying performance bottlenecks, you can use CodeArts PerfTest to implement association analysis and generate performance reports.
   - Based on the historical metric data learned by using intelligent algorithms, APM associates metrics for analysis from multiple dimensions, extracts the context data of both normal and abnormal services for comparison, and locates root causes through cluster analysis.

Advantages

Ease of Use

Connects to applications without having to modify code, and collects data in a non-intrusive mode.

- APM Agents collect service call, service inventory, and call KPI data.

High Performance

Delivers high throughput (hundreds of millions of API calls), ensuring premium experience.

Open Ecosystem

Provides open APIs to query O&M data, offers collection standards, and supports independent development.

Intelligent Analysis

Reports alarms using Artificial Intelligence (AI) threshold detection and machine learning based on historical baseline data, and supports root cause analysis.

1.2 Functions

APM manages cloud application performance. It provides application metric monitoring, tracing, application topology, resource tag management, URL tracing, intelligent alarm reporting, tag/Agent/configuration/system management, and application monitoring.

Application Metric Monitoring

This function enables you to monitor the overall health status of applications. APM Agents collect metrics of JVM, GC, service calls, exceptions, external calls, database access, and middleware of Java applications, helping you monitor application running.

Tracing

APM comprehensively monitors calls and displays service execution traces and statuses, helping you quickly locate performance bottlenecks and faults.

- In the displayed trace list, click the target trace to view its basic information.

- On the trace details page, you can view the trace's complete information, including the local method stack and remote call relationships.

Application Topology

There are two types of application topologies:

- Single-component topology: topology of a single component under a certain environment. You can also view the call relationships of direct and indirect upstream and downstream components.
- Global application topology: topology of some or all components under an application.

The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults. The application relationships, call data (service and instance metrics), and health status are clearly displayed.

URL Tracing

If you need to find out the call relationships of an important application (for example, calling an e-commerce system's API to create orders), use URL tracing analysis. In APM, URL tracing consumes a large number of resources. Therefore, an entry URL will not be added for tracing by default. However, you can set that if necessary. APM has a limit on the total number of URLs added for tracing. It focuses on tracing the downstream calls for the APIs that are added for tracing. Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and detect problems more precisely.

Resource Tag Management

You can tag resources under your account for classification.

Tag Management

You can add tags for different environments and applications for easy management.

Intelligent Alarm Reporting

When an application connected to APM meets a preset alarm condition, an alarm is triggered and reported. In this way, you can quickly learn about service exceptions and rectify faults to prevent loss.

APM allows you to configure alarm templates. You can create multiple alarm policies under a template and bind them to nodes.

With intelligent alarm reporting, you can receive alarms by SMS, email, or function.

Agent Management

You can view the deployment and running statuses of the Agents that are connected to APM, and to stop, start, or delete them.

Configuration Management

Configuration Management consists of **Collection Center** and **Data Masking**.

- **Collection Center**: displays collectors in a centralized manner. You can view and manage various collectors, metrics, and collection parameters supported by APM.
- **Data Masking**: You can set policies to mask the data reported using APM APIs.

System Management

System Management consists of **Access Keys**, **General Configuration**, and **Agent Count**.

- **Access Keys**: Access Key ID (AK) and Secret Access Key (SK) are your long-term identity credentials. JavaAgents report data with an AK. An AK is used together with an SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.
- **General Configuration**: You can determine whether to collect data through bytecode instrumentation, and specify the slow request threshold and maximum number of rows to collect.
- **Agent Count**: APM can count the Agents used by tenants. You can view the number of Agents by time, region, or Agent type.

1.3 Application Scenarios

APM is widely used. The following lists some typical scenarios.

Diagnosis of Application Exceptions

Pain Points

In the distributed microservice architecture, enterprises can develop diverse applications efficiently, but face great challenges in traditional O&M and diagnosis. An e-commerce application may face the following problems:

- Difficult fault locating

  After receiving the feedback from customers, customer service personnel submit problems to technical personnel for troubleshooting. In the distributed microservice architecture, a request usually undergoes multiple services/nodes before a result is returned. If a fault occurs, O&M personnel need to repeatedly view logs on multiple hosts to locate the fault. Even for simple problems, troubleshooting requires cooperation from multiple teams.
- Difficult architecture sort-out

  When service logic becomes complex, it is difficult to find out the downstream services (databases, HTTP APIs, and caches) that an application depends on,

and external services that depend on the application from the code perspective. It is also difficult to sort out the service logic, manage the architecture, and plan capacities. For example, enterprises are hard to determine the number of hosts required in their activities.

Service Implementation

APM can diagnose exceptions in large distributed applications. When an application breaks down or a request fails, you can locate faults in minutes through topologies and drill-downs.

- Visible topology: Abnormal application instances can be automatically discovered on the topology.
- Tracing: You can locate root causes in code through drill-downs after identifying abnormal applications.
- SQL analysis: APM displays graphs of key metrics (such as number of SQL statement calls, latency, and number of errors), and supports analysis of database performance problems caused by abnormal SQL statements.

User Experience Management

Pain Points

In the Internet era where user experience is of crucial importance, you cannot obtain user access information even if backend services run stably. It is much more difficult to locate frontend problems that occur occasionally. After a system goes online, if users cannot access the system due to errors and APM fails to obtain the information in time, lots of users will choose to leave. If users report page problems, how can APM reproduce the problems immediately? How can error details be obtained for fast troubleshooting?

Service Implementation

APM analyzes the complete process (user request > server > database > server > user request) of application transactions in real time, enabling you to monitor comprehensive user experience in real time. For transactions with poor user experience, locate problems through topologies and tracing.

- Application KPI analysis: KPIs such as throughput, latency, and call success rate are displayed, so that you can monitor user experience easily.
- Full-link performance tracing: Web services, caches, and databases are traced, so that you can detect performance bottlenecks quickly.

Intelligent Diagnosis

Pain Points

Massive services bring abundant but unassociated application O&M data, including hundreds of monitoring metrics, KPI data, and tracing data. How can metric and alarm data be associated for analysis from the application, component, or URL tracing perspective? How can possible causes be provided for exceptions based on the historical data and O&M experience library?

Service Implementation

APM supports automatic detection of faults using machine learning algorithms, and intelligent diagnosis. When an exception is found during URL tracing, APM

learns historical metric data based on intelligent algorithms, associates exception metrics for multi-dimensional analysis, extracts characteristics of context data (such as resources, parameters, and call structures) for both normal and abnormal services, and locate root causes through cluster analysis.

1.4 Basic Concepts

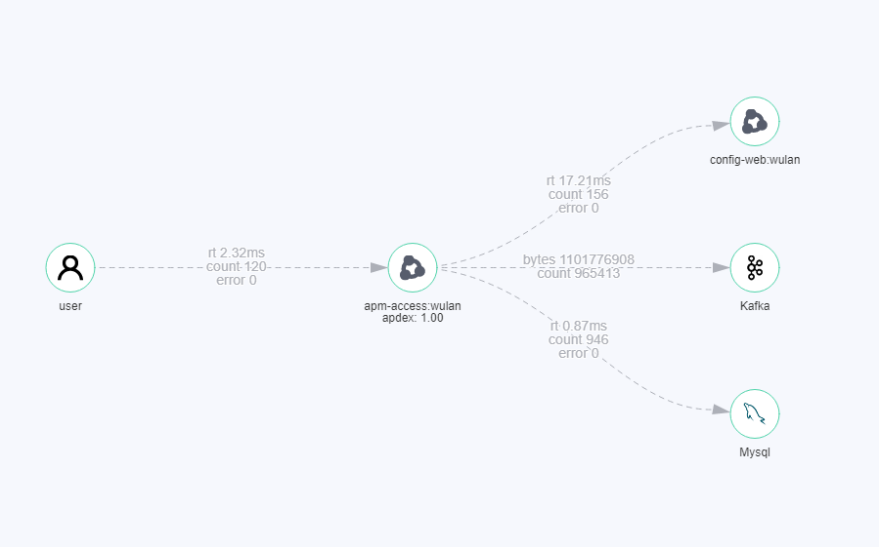
Application Topology

A topology graphically displays call and dependency relationships between applications. It is composed of circles, lines with arrows, and resources. Each line with an arrow represents a call relationship. The number of requests, average response time, and the number of errors are displayed above the line. Different colors indicate different RT ranges, helping you quickly detect and locate faults.

NOTE

- Database: When the database call time is greater than or equal to 100 ms, the value turns yellow. When this time is greater than or equal to 200 ms, the value turns red.
- Cache: When the cache call time is greater than or equal to 10 ms, the value turns yellow. When this time is greater than or equal to 30 ms, the value turns red.
- Other API calls: When the API call time is greater than or equal to 500 ms, the value turns yellow. When this time is greater than or equal to 1000 ms, the value turns red.
- If the number of errors is greater than 0, the value turns red.

Figure 1-1 Application topology



Tracing

By tracing and recording application calls, APM displays the execution traces and statuses of application requests in systems, so that you can quickly locate performance bottlenecks and faults.

APM Agent

APM Agents use bytecode enhancement technology to collect application performance data in real time. They run on the server where applications are deployed. For details about data collection and usage, see **Data Collection**. Before using APM, ensure that APM Agents have been installed.

URL Tracing

URL tracing is to trace the call relationship of an application. For example, the complete process of calling an e-commerce system's API to create orders is "user request > web server > database > web server > user request."

If a URL is added for tracing, APM will focus on tracing its downstream calls. Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and detect problems more precisely.

Apdex

Apdex is an open standard developed by the Apdex alliance. It defines a standard method to measure application performance. The Apdex standard converts the application response time into user satisfaction with application performance in the range of 0 to 1.

- Apdex principle

Apdex defines the threshold "T" for application response time. "T" is determined based on performance expectations. Based on the actual response time and "T", user experience can be categorized as follows:

Satisfied: indicates that the actual response time is shorter than or equal to "T". For example, if "T" is 1.5s and the actual response time is 1s, user experience is satisfied.

Tolerable: indicates that the actual response time is greater than "T", but shorter than or equal to "4T". For example, if "T" is 1s, the tolerable upper threshold for the response time is 4s.

Frustrated: indicates that the actual response time is greater than "4T".

- Apdex calculation

  In APM, the Apdex threshold is the maximum response time that makes users satisfied. The application response latency is the service latency. The Apdex value ranges from 0 to 1 and is calculated as follows:

  Apdex = (Number of satisfied samples + Number of tolerable samples x 0.5)/ Total number of samples

CMDB

Configuration Management Database (CMDB) structures and displays application resource configuration, so that you can better monitor and manage applications. It consists of:

- **Application** (global concept): refers to a logical unit. You can view the same application information in all regions. For example, an independent functional module under an account can be regarded as an application.
- **Sub-application** (global concept): similar to a folder. You can create up to three layers of sub-applications under an application.
- **Component** (global concept): refers to a program or microservice. It is generally used together with environments. A component can contain one or more environments. For example, an order app can be deployed in the function test environment, pressure test environment, pre-release environment, or live network environment.
- **Environment**: Components or programs with different configurations are deployed in different environments. Each environment has its own region attribute. You can filter environments by region. You can also add one or more tags to an environment and filter environments by tag.
- **Instance**: refers to a process in an environment. It is named in the format of "host name+IP address+instance name." An environment is usually deployed on different hosts or containers. If an environment is deployed on one host, differentiation by instance is supported.
- **Environment tag**: an attribute for filtering environments. Different environments may have the same tag. Tags carry public configuration capabilities. For example, the configuration set on a tag can be shared by the environments with the same tag. Tags defined for environments of one application cannot be applied to other applications.

1.5 Edition Differences

APM provides basic and enterprise editions and supports interconnection with Java applications. The following table lists the functions supported by each edition.

Edition	Free	Enterprise
Description	Free of charge. Up to 10 Agents can be connected. Reactivate them every 15 days.	All functions are open.
Data storage duration	7 days	30 days
Application topology	√	√
Tracing	√	√
Metric monitoring	√	√
URL tracing analysis	√	√
Alarm	√	√

Edition	Free	Enterprise
CMDB	√	√
Note: √ indicates supported, and **x** indicates not supported.		

NOTE

The enterprise edition cannot be downgraded back to the free edition (basic edition).

1.6 Permissions Management

If you need to assign different permissions to employees in your enterprise to access your APM resources, Identity and Access Management (IAM) is a good choice for fine-grained permissions management. IAM provides identity authentication, permissions management, and access control, helping you secure access to your cloud resources.

With IAM, you can use your account to create IAM users for your employees, and assign permissions to the users to control their access to specific resources. For example, some software developers in your enterprise need to use APM resources but cannot delete them or perform any high-risk operations. To achieve this result, you can create IAM users for the software developers and grant them only the permissions required for using APM resources.

If your account does not need individual IAM users for permissions management, you may skip over this chapter.

IAM can be used free of charge. You pay only for the resources in your account. For more information about IAM, see **IAM Service Overview**.

NOTICE

Traces and Agent statistics do not involve your entity resources. To ensure statistics integrity, authorized users can check the trace and Agent statistics of the tenant, including those in other enterprise projects.

APM Permissions

By default, new IAM users do not have any permissions assigned. You need to add a user to one or more groups, and assign permissions policies or roles to these groups. The user then inherits permissions from the groups it is a member of. This process is called authorization. After authorization, the user can perform specified operations on APM.

APM is a global service. By default, the APM permissions granted to a user take effect in all regions supported by APM. APM resources are isolated by tenant. All users under a tenant share resources. To isolate resources, use enterprise projects.

APM is a global service and can be accessed without specifying a physical region.

Table 1-2 lists all the system permissions supported by APM.

Table 1-2 System permissions supported by APM

Role	Description	Category
APM FullAccess	Full permissions for APM	System-defined policy
APM ReadOnlyAccess	Read-only permissions for APM	System-defined policy

Table 1-3 lists the common operations supported by each system-defined policy or role of APM. Choose policies or roles as required.

Table 1-3 Common operations supported by each system-defined policy or role of APM

Operation	APM FullAccess	APM ReadOnlyAccess
Querying the alarm list	√	√
Querying alarm details	√	√
Querying alarm notification details	√	√
Obtaining application configuration	√	√
Creating application configuration	√	x
Deleting application configuration	√	x
Modifying application configuration	√	x
Querying a tag	√	√
Adding a tag	√	x
Deleting a tag	√	x
Modifying a tag	√	x
Querying a resource tag	√	√
Adding a resource tag	√	x
Deleting a resource tag	√	x
Modifying a resource tag	√	x

Operation	APM FullAccess	APM ReadOnlyAccess
Querying an alarm template	√	√
Adding an alarm template	√	x
Deleting an alarm template	√	x
Modifying an alarm template	√	x
Obtaining a notification	√	√
Deleting a notification	√	x
Adding a notification	√	x
Modifying a notification	√	x
Obtaining URL tracing configuration	√	√
Deleting URL tracing configuration	√	x
Adding a URL for tracing	√	x
Modifying URL tracing configuration	√	x
Querying a URL tracing view	√	√
Obtaining the URL tracing list	√	√
Obtaining the global topology	√	√
Querying a sub-application	√	√
Querying environment configuration	√	√
Adding environment configuration	√	x
Deleting environment configuration	√	x
Modifying environment configuration	√	x
Obtaining an instance	√	√

Operation	APM FullAccess	APM ReadOnlyAccess
Deleting an instance	√	x
Modifying an instance	√	x
Querying a monitoring item	√	√
Modifying a monitoring item	√	x
Obtaining collection status	√	√
Obtaining a custom alarm policy	√	√
Deleting a custom alarm policy	√	x
Modifying a custom alarm policy	√	x
Creating a custom alarm policy	√	x
Obtaining the environment topology	√	√
Obtaining a metric view	√	√
Obtaining the trace list	√	√
Obtaining trace details	√	√
Obtaining collector information	√	√
Obtaining an access key	√	x
Modifying an access key	√	x
Deleting an access key	√	x
Adding an access key	√	x
Obtaining general configuration	√	√
Modifying general configuration	√	x
Querying Agent statistics	√	√

1.7 Metric Overview

1.7.1 Metric Overview

A metric describes resource performance data or status. It consists of the metric type, name, and description, data type, and default aggregation mode.

NOTE

For the default aggregation mode, **LAST** indicates the value of the latest metric collected. **SUM** indicates the sum of collected metrics. **MAX** indicates the maximum value of collected metrics. **AVG** indicates the average value of collected metrics.

1.7.2 Exception

This section describes the types, names, and meanings of exception metrics collected by APM.

Table 1-4 Exception collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Determine Trace Exception upon Log Error Detection	radio	JAVA	true	2.0.0	-	Whether to mark a trace as abnormal after a log error is collected.

Table 1-5 Exception metric description

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception logs (**exception**: statistics about all exception logs)	className	Exception Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	logType	Log Type	Exception log type	-	ENUM	LAST
	count	Count	Number of times that an exception has occurred	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack	-	CLOB	LAST
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
Log version (**logVersion**: package version of the log component)	logType	Log Type	Log type	-	ENUM	LAST
	version	Log Version	Log version	-	STRING	LAST

1.7.3 Basic Monitoring

1.7.3.1 GC Monitoring

This section describes the types, names, and meanings of GC metrics collected by APM.

Table 1-6 GC metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
GC statistics (**gc**)	fullGCCount	Full GC Times	Number of full GC times in a collection period	-	INT	SUM
	fullGCCountTotal	Total Full GC Times	Total number of full GC times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	fullGCTime	Full GC Time	Full GC duration in a collection period	ms	INT	SUM
	fullGCTimeTotal	Total Full GC Time	Total full GC duration	ms	INT	SUM
	fullGCMBeanName	Full GC Recycler	Name of the full GC recycler	-	STRING	LAST
	youngGCCount	Young GC Times	Number of young GC times in a collection period	-	INT	SUM
	youngGCCountTotal	Total Young GC Times	Total number of young GC times	-	INT	SUM
	youngGCTime	Young GC Time	Young GC duration in a collection period	ms	INT	SUM
	youngGCTimeTotal	Total Young GC Time	Total young GC duration	ms	INT	SUM
	youngGCMBeanName	Young GC Recycler	Name of the young GC recycler	-	STRING	LAST
GC details (**gcdetail**)	action	GC Type	GC type, which can be **major** or **minor**	-	ENUM	LAST
	cause	GC Cause	GC cause	-	ENUM	LAST
	name	GC Name	GC collector name	-	STRING	LAST
	count	Count	Number of times that GC has occurred	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	GC Time	GC duration	ms	INT	SUM
	maxTime	Max. GC Time	Time consumed by the slowest GC	ms	INT	MAX
	detail	GC Details	Details about the slowest GC	-	CLOB	LAST

1.7.3.2 JavaMethod

This section describes the types, names, and meanings of JavaMethod metrics collected by APM.

Table 1-7 JavaMethod collection parameter

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Method Interception Configuration	obj_array	JAVA	-	2.0.0	-	Specify methods to intercept. Use commas (,) to separate methods. If this parameter is left blank, all public methods will be intercepted by default.

Table 1-8 JavaMethod metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Java method (**method**: Method call statistics are collected based on the configured Java method names.)	class	Class	Class	-	ENUM	LAST
	method	Method	Method	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM
	lastError	Error Message	Error information of the method	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the method	ms	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	totalTime	Total RT	Total response time of the method	ms	INT	SUM

1.7.3.3 JVM Monitoring

This section describes the types, names, and meanings of JVM metrics collected by APM.

Table 1-9 Collection parameters for JVM monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Trace Stack Collection Threshold	integer	JAVA	0	2.0.4	-	Stacks will be automatically printed when the request latency exceeds the threshold.

Table 1-10 JVM monitoring metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Class loading (**classLoading**: JVM class loading statistics)	loadedClassCount	Loaded Classes	Number of loaded classes	-	INT	SUM
	totalLoadedClassCount	Total Loaded Classes	Total number of loaded classes	-	INT	SUM
	unloadedClassCount	Unloaded Classes	Number of unloaded classes	-	INT	SUM
Compilation (**compile**: JVM class compilation time statistics)	compilationTime	Compilation Time	Compilation time in a collection period	ms	INT	SUM
	totalCompilationTime	Total Compilation Time	Total compilation time	ms	INT	SUM
CPU (**CPU**: CPU usage statistics of JVM processes)	cpuRatio	CPU Usage	CPU usage of the Java process	%	DOUBLE	AVG
	cpuRatioMax	Max. CPU Usage	Maximum CPU usage of the Java process	%	DOUBLE	MAX
	cpuTimeInterval	CPU Time	CPU time of the Java process in the collection interval	ns	INT	SUM
	processorCount	Processors	Number of processors	-	INT	SUM
	systemTimeInterval	Collection Interval	Collection interval	ns	INT	SUM
	totalCpuTime	Total CPU Time	Total CPU time	ns	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Memory (**memory**: JVM memory statistics)	directMemoryUsage	Direct Memory	Used direct memory	M	INT	AVG
	directMemoryCapacity	Direct Memory Capacity	Total direct memory capacity	M	INT	AVG
	heapMemoryUsage	Heap Memory	Used heap memory	M	INT	AVG
	nonHeapMemoryUsage	Non-Heap Memory	Used non-heap memory	M	INT	AVG
	objectPendingFinalizationCount	Objects Being Recycled	Number of objects that are being recycled at the time of collection	-	INT	SUM
Memory pool (**memoryPool**: statistics collected by JVM memory pool)	committed	Available Memory	Available memory	Byte	INT	SUM
	init	Initialized Memory	Initialized memory	Byte	INT	SUM
	max	Max. Memory	Maximum memory	Byte	INT	SUM
	name	Memory Pool Name	Memory pool name	-	ENUM	LAST
	used	Used Memory	Used memory	Byte	INT	SUM
Thread (**thread**: JVM thread statistics)	currentThreadCpuTime	Thread CPU Time	CPU time of the current thread	-	INT	SUM
	currentThreadUserTime	Thread User Time	User time of the current thread	-	INT	SUM
	daemonThreadCount	Daemon Threads	Number of daemon threads	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	deadlockedThreadsCount	Deadlock Threads	Number of deadlock threads	-	INT	SUM
	monitorDeadlockedThreads	Current Deadlock Threads	ID list of current deadlock threads	-	INT	SUM
	peakThreadCount	Max. Threads Executed	Maximum number of threads executed	-	INT	SUM
	threadCount	Current Threads	Number of current threads	-	INT	SUM
	totalStartedThreadCount	Total Threads	Total number of threads that are started since the Java process is started	-	INT	SUM
	newThreadCount	Initial Threads	Number of threads in the initial state	-	INT	SUM
	runnableThreadCount	Running Threads	Number of running threads	-	INT	SUM
	blockedThreadCount	Blocked Threads	Number of blocked threads	-	INT	SUM
	waitingThreadCount	Pending Threads	Number of pending threads	-	INT	SUM
	timedWaitingThreadCount	Timed-out Threads	Number of threads that timed out	-	INT	SUM
	terminatedThreadCount	Terminated Threads	Number of terminated threads	-	INT	SUM

1.7.3.4 JVMInfo

This section describes the types, names, and meanings of JVMInfo metrics collected by APM.

1.7.3.5 Netty Memory

This section describes the types, names, and meanings of Netty memory metrics collected by APM.

Table 1-11 Netty memory metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Memory (**memory**: memory metrics)	directMemoryUsage	directMemoryUsage	Used direct memory	-	INT	AVG
	maxDirectMemory	maxDirectMemory	Maximum direct memory	-	INT	MAX
Exception (**exception**)	causeType	Class	Class	-	ENUM	LAST
	exceptionType	ExceptionType	Exception type	-	ENUM	LAST
	count	Count	Count	-	INT	SUM
	message	ExceptionMessage	Exception message	-	STRING	LAST
	stackTrace	ExceptionStack	Exception stack	-	CLOB	LAST

1.7.3.6 Threads

This section describes the types, names, and meanings of thread metrics collected by APM.

Table 1-12 Thread collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Max. Rows of Thread Details	integer	JAVA	1	2.3.19	-	Maximum number of rows of thread details. You can set it to up to 50.

Table 1-13 Thread metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Thread details (**threadDetail**)	threadName	Thread Name	Thread name	-	ENUM	LAST
	memory	Memory	Memory	-	INT	SUM
	stack	Thread Stack	Thread stack	-	CLOB	LAST
	ids	Thread ID	Thread ID	-	STRING	LAST
	cpuTime	Thread CPU Time	Thread CPU time	ms	INT	SUM
	count	Threads	Number of threads	-	INT	LAST

1.7.4 Databases

1.7.4.1 C3P0 Connection Pool Monitoring

This section describes the types, names, and meanings of C3P0 connection pool metrics collected by APM.

Table 1-14 Collection parameters for C3P0 connection pool monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.1.3	-	Threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported.
Obtain Pool Info or Not	radio	JAVA	false	2.1.3	-	Whether to obtain pool information when getting connections

Table 1-15 C3P0 connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Data source (**dataSource**)	url	URL	URL	-	ENUM	LAST
	driverClass	Driver	Driver	-	STRING	LAST
	initialPoolSize	initialPoolSize	Number of initialized connections	-	INT	LAST
	minPoolSize	minPoolSize	Minimum connection pool size	-	INT	LAST
	maxPoolSize	maxPoolSize	Maximum connection pool size	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	numIdleConnections	numIdleConnections	Number of idle connections	-	INT	LAST
	numBusyConnections	numBusyConnections	Number of busy connections	-	INT	LAST
	numConnections	numConnections	Total number of connections	-	INT	LAST
	maxIdleTime	maxIdleTime	Maximum connection idle time	-	INT	LAST
	idleConnectionTestPeriod	idleConnectionTestPeriod	Interval for checking for idle connections	-	INT	LAST
	testConnectionOnCheckout	testConnectionOnCheckout	Connection validity check during check-out	-	STRING	LAST
	testConnectionOnCheckin	Connection Validity Check During Check-In	Connection validity check during check-in	-	STRING	LAST
	acquireRetryAttempts	Connection Retries	Number of Connection retry times	-	INT	LAST
	acquireRetryDelay	Connection Retry Interval	Connection retry interval	-	INT	LAST
	acquireIncrement	Connections Created If No Connection Exists	Number of connections created if no connection exists	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection details (**connection**)	url	Connection Address	Connection address	-	ENUM	LAST
	invokeCount	Calls	Number of calls	-	INT	LAST
	totalTime	Total Time	Total time	-	INT	LAST
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
Version (**version**)	version	Version	Version	-	STRING	LAST
Exception (**exception**: C3P0 call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

1.7.4.2 Cassandra Monitoring

This section describes the types, names, and meanings of Cassandra metrics collected by APM.

Table 1-16 Collection parameters for Cassandra monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.2.9	-	Threshold for reporting borrowConnection() method traces. If the threshold is not exceeded, such traces will not be reported.
Collect Original CQL Statement or Not	radio	JAVA	false	2.2.9	-	Whether to collect original CQL statements

Table 1-17 Cassandra metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
CQL call (**Cql**)	cql	cql	Executed CQL Statement	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	maxTime	Maximum response time	-	INT	MAX
	queryRowCount	Read Rows	Number of read rows	-	INT	SUM
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	totalTime	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Cassandra node call (**node**)	node	Node Address	Node address	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	maxTime	Maximum response time	-	INT	MAX
	totalTime	totalTime	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Cassandra cluster call (**cluster**)	nodes	Cluster Node	Cluster node information	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	maxTime	Maximum response time	-	INT	MAX
	totalTime	totalTime	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Connection details (**connection**)	host	Connected Host	Connected host	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Exception (**exception**: Cassandra call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	stackTrace	Exception stack information	-	CLOB	LAST
Cassandra summary (**total**: summary of Cassandra call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	queryRowCount	Total Read Rows	Total number of read rows	-	INT	SUM
	errorCount	Total Errors	Total number of errors	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
Cassandra version (**version**)	version	Version	Version	-	STRING	LAST

1.7.4.3 ClickHouse Database

This section describes the types, names, and meanings of ClickHouse database metrics collected by APM.

Table 1-18 ClickHouse database collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Collect Original SQL Statement or Not	radio	JAVA	false	2.0.0	-	Whether to collect and report original SQL statements

Table 1-19 ClickHouse database metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Database connection (**connection**: APM counts SQL call statistics by database.)	db	Database	Database name	-	ENUM	LAST
	createdCount	Created Connections	Number of connections created by the database	-	INT	SUM
	currentCount	Current Connections	Current number of connections of the database	-	INT	SUM
	destroyedCount	Destroyed Connections	Number of the database's connections that have been destroyed	-	INT	SUM
	errorCount	Errors	Number of errors that the database encounters	-	INT	SUM
	invokeCount	Calls	Number of times that the database is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the database	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	readRowCount	Read Rows	Number of rows read from the database	-	INT	SUM
	updatedRowCount	Updated Rows	Number of rows updated in the database	-	INT	SUM
	totalTime	Total RT	Total response time of the database	-	INT	SUM
	slowestSql	Slowest SQL	Slowest SQL statement of the database in the collection period	-	STRING	LAST
Exception (**exception**: exception statistics about SQL calls)	causeType	Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Message	Exception message	-	STRING	LAST
	sql	Exception SQL	SQL statement that encounters an exception	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Version (**version**: ClickHouse package version)	version	Version	Driver package version	-	STRING	LAST
SQL monitoring (**sql**: APM counts call statistics by SQL.)	sql	SQL ID	Unique ID of the SQL statement, which is used for alarm configuration	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the SQL statement	-	INT	MAX
	errorCount	Errors	Number of errors that the SQL statement encounters	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of times that the SQL statement is called	-	INT	SUM
	lastError	Error Message	SQL error information	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the SQL statement	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	readRowCount	Read Rows	Number of read rows of the SQL statement	-	INT	SUM
	runningCount	Ongoing Executions	Number of SQL statements that are being executed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	sqlString	SQL Statement	SQL statement	-	STRING	LAST
	totalTime	Total RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Number of updated rows of the SQL statement	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Summary (**total**: summary about SQL statement call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	errorCount	Errors	Total number of errors	-	INT	SUM
	readRowCount	Read Rows	Total number of read rows	-	INT	SUM
	totalTime	RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Total number of updated rows	-	INT	SUM

1.7.4.4 DBCP Connection Pool Monitoring

This section describes the types, names, and meanings of DBCP connection pool metrics collected by APM.

Table 1-20 DBCP connection pool collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.1.3	-	Threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported.

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Obtain Pool Info or Not	radio	JAVA	false	2.1.3	-	Whether to obtain pool information when getting connections

Table 1-21 DBCP connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Data source (**dataSource**)	url	url	url	-	ENUM	LAST
	driverClassName	Driver	Driver	-	STRING	LAST
	initialSize	Initialized Connections	Number of initialized connections	-	INT	LAST
	minIdle	Min. Idle Connections	Minimum number of idle connections in the pool	-	INT	LAST
	maxIdle	Max. Idle Connections	Maximum number of idle connections in the pool	-	INT	LAST
	maxTotal	Max. RT	Maximum response time	-	INT	LAST
	numIdle	Idle Connections	Number of idle connections	-	INT	LAST
	numActive	Active Connections	Number of active connections	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxWaitMillis	Max. Time for Waiting Connection to Be Reclaimed	Maximum time for a waiting connection to be reclaimed (when no connection is available) before an exception is thrown	-	INT	LAST
	testOnCreate	Validity Check Upon Connection Creation	Whether to check the validity of a connection after it is created	-	STRING	LAST
	testOnBorrow	Validity Check Before Obtaining Connection	Check whether a connection is valid before obtaining it from the connection pool.	-	STRING	LAST
	testWhileIdle	Idle Connection Validity Check	Whether to verify the validity of an idle connection when an application applies for it from the pool	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	timeBetweenEvictionRunsMillis	Interval for Checking Connection Validity	If **testOnBorrow** is set to **false** and **testWhileIdle** is set to **true**, the application checks whether the idle time of a connection is greater than **timeBetweenEvictionRunsMillis** before obtaining the connection. If it is greater than that value, the application checks whether the connection is valid.	-	INT	LAST
	removeAbandonedOnBorrow	Remove Discarded Connections When Obtaining Connections	Whether to remove discarded connections when obtaining connections. (The following conditions must be met: "getNumActive() > getMaxTotal() – 3" and "getNumIdle() < 2")	-	STRING	LAST
	removeAbandonedOnMaintenance	Remove Discarded Connections During Maintenance	Whether to remove discarded connections in the maintenance cycle (when the eviction ends)	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	removeAbandonedTimeout	Connection Removal Timeout	If a connection is not used within the specified timeout, it is regarded as a discarded connection and can be removed.	-	INT	LAST
Connection details (**connection**)	url	Connection Address	Connection address	-	ENUM	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
Version (**version**)	version	Version	Version	-	STRING	LAST
Exception (**exception**: exception statistics of DBCP calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

1.7.4.5 Druid Connection Pool Monitoring

This section describes the types, names, and meanings of Druid connection pool metrics collected by APM.

Table 1-22 Druid connection pool collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.1.3	-	Threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported.
Obtain Pool Info or Not	radio	JAVA	false	2.1.3	-	Whether to obtain pool information when getting connections

Table 1-23 Druid connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Data source (**dataSource**)	url	url	url	-	ENUM	LAST
	dbType	Database Type	Database type	-	STRING	LAST
	driverClassName	Driver	Driver	-	STRING	LAST
	initialSize	Initialized Connections	Number of initialized connections	-	INT	LAST
	minIdle	Min. Idle Connections	Minimum number of idle connections in the pool	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxIdle	Max. Idle Connections	Maximum number of idle connections in the pool	ms	INT	LAST
	maxActive	Max. Pool Size	Maximum connection pool size	-	INT	LAST
	waitThreadCount	Waiting Threads	Number of waiting threads	-	INT	LAST
	maxWaitThreadCount	Max. Waiting Threads	Maximum number of waiting threads	-	INT	LAST
	poolingCount	Pool Connections	Number of connections in the pool	-	INT	LAST
	poolingPeak	Max. Pool Connections	Maximum number of connections in the pool	-	INT	MAX
	activeCount	Active Connections	Number of active connections	-	INT	LAST
	activePeak	Max. Active Connections	Maximum number of active connections	-	INT	MAX
	logicConnectCount	Total Connections	Total number of connections	-	INT	SUM
	maxWait	Max. Waiting Time	Maximum waiting time of a connection	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	removeAbandoned	Automatically Reclaim Timeout Connections	Whether to automatically reclaim timeout connections	-	STRING	LAST
	removeAbandonedCount	Timeout Connection Reclaims	Number of times that timeout connections are reclaimed	-	INT	LAST
	removeAbandonedTimeoutMillis	Max. Connection Usage Duration	If a connection in the pool is not returned within the specified duration, the connection will be reclaimed.	-	INT	LAST
	testWhileIdle	Idle Connection Validity Check	Whether to verify the validity of an idle connection when an application applies for it from the pool	-	STRING	LAST
	testOnBorrow	Validity Check Before Obtaining Connection	Check whether a connection is valid before obtaining it from the connection pool.	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	testOnReturn	Validity Check Upon Connection Return	Whether to verify the validity of a connection when it is returned	-	STRING	LAST
	minEvictableIdleTimeMillis	Allowed Idle Time for Connection	Idle time that is allowed for connections in the pool	ms	INT	LAST
	timeBetweenEvictionRunsMillis	Interval for Checking Idle Connection Validity	Interval for checking the validity of idle connections	-	INT	LAST
Connection details (**connection**)	url	Connection Address	Connection address	-	ENUM	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
Version (**version**)	version	Version	Version	-	STRING	LAST
Exception (**exception**: exception statistics of Druid calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

1.7.4.6 EsRestClient Monitoring

This section describes the types, names, and meanings of EsRestClient metrics collected by APM.

Table 1-24 EsRestClient collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Index Normalization Configuration	obj_array	JAVA	-	2.0.0	-	Implement "regex" matching and normalize the URL index.

Table 1-25 EsRestClient metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of EsRestClient calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Client information (**clientInfo**)	clientId	Client ID	Client ID	-	ENUM	LAST
	RestClientVersion	RestClient Version	RestClient version	-	STRING	LAST
	RestHighLevelClientVersion	RestHighLevelClient Version	RestHighLevelClient version	-	STRING	LAST
	poolId	HttpAsyncClient Connection Pool ID	HttpAsyncClient Connection pool ID	-	STRING	LAST
	esNodes	Cluster Node Information Set on Client	Cluster node information set on the client	-	STRING	LAST
	esDeadNodes	Disconnected Node	Disconnected node of the cluster	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
URL monitoring (**esClient**: APM counts URL call statistics by URL.)	clientId	clientId	RESTClient ID	-	ENUM	LAST
	url	URL	Called URL	-	ENUM	LAST
	method	HTTP Method	HTTP method of the URL	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the URL	-	INT	MAX
	errorCount	Errors	Number of call errors of the URL	-	INT	SUM
	definitiveFailureCount	Request Errors	Number of request errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	hostUri	hostUri	host uri	-	STRING	LAST
	invokeCount	Calls	Number of times that the URL is called	-	INT	SUM
	lastError	Error Message	Error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the called URL	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time of the called URL	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	retryCount	Retries	Request retry times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Status code statistics (**code**: APM counts URL call statistics by status code.)	code	Status Code	Status code	-	ENUM	LAST
	url	URL	URL that returns the status code	-	STRING	LAST
	count	Count	Number of times that the status code has occurred	-	INT	SUM
EsRestClient summary (**total**: summary of EsRestClient call statistics)	definitiveFailureCount	Total Request Errors	Total number of request errors	-	INT	SUM
	invokeCount	Calls	Total number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
	retryCount	Total Request Retries	Total number of request retries	-	INT	SUM
EsRestClient node call monitoring (**serverNode**)	serverAddr	Server Node	Server node information	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
URL monitoring (**invocation**: APM counts URL call statistics by URL.)	url	URL	Called URL	-	ENUM	LAST
	method	HTTP Method	HTTP method of the URL	-	ENUM	LAST
	client	Client Type	EsRestClient type	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the URL	-	INT	MAX
	errorCount	Errors	Number of call errors of the URL	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	hostUri	Call Address	Called URL address	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	Calls	Number of times that the URL is called	-	INT	SUM
	lastError	Error Message	Error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the called URL	-	INT	MAX
	responseCloseCount	responseCloseCount	Number of closed responses when the URL is called	-	INT	SUM
	totalTime	Total RT	Total response time of the called URL	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	envId	Cluster ID	Cluster ID corresponding to the called URL	-	STRING	LAST
EsRestClient cluster call (**cluster**)	esNodes	Cluster Node	Cluster node information	-	ENUM	LAST
	clientCount	Created RestClients	Number of RestClients that have been created	-	INT	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

1.7.4.7 GaussDB Database

This section describes the types, names, and meanings of GaussDB database metrics collected by APM.

Table 1-26 GaussDB database collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Collect Original SQL Statement or Not	radio	JAVA	false	2.2.8	-	Whether to collect and report original SQL statements

Table 1-27 GaussDB database metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Database connection (**connection**: APM counts SQL call statistics by database.)	db	Database	Database name	-	ENUM	LAST
	createdCount	Created Connections	Number of connections created by the database	-	INT	SUM
	currentCount	Current Connections	Current number of connections of the database	-	INT	SUM
	destroyedCount	Destroyed Connections	Number of the database's connections that have been destroyed	-	INT	SUM
	errorCount	Errors	Number of errors that the database encounters	-	INT	SUM
	invokeCount	Calls	Number of times that the database is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the database	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	ranges	Custom RT Range	Custom response time range	-	STRING	LAST
	readRowCount	Read Rows	Number of rows read from the database	-	INT	SUM
	updatedRowCount	Updated Rows	Number of rows updated in the database	-	INT	SUM
	totalTime	Total RT	Total response time of the database	-	INT	SUM
	slowestSql	Slowest SQL	Slowest SQL statement of the database in the collection period	-	STRING	LAST
Exception (**exception**: exception statistics about SQL calls)	causeType	Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Message	Exception message	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	sql	Exception SQL	SQL statement that encounters an exception	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Version (**version**: GaussDB package version)	version	Version	Driver package version	-	STRING	LAST
SQL monitoring (**sql**: APM counts call statistics by SQL.)	sql	SQL ID	Unique ID of the SQL statement, which is used for alarm configuration	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the SQL statement	-	INT	MAX
	errorCount	Errors	Number of errors that the SQL statement encounters	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of times that the SQL statement is called	-	INT	SUM
	lastError	Error Message	SQL error information	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the SQL statement	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	readRowCount	Read Rows	Number of read rows of the SQL statement	-	INT	SUM
	runningCount	Ongoing Executions	Number of SQL statements that are being executed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	sqlString	SQL Statement	SQL statement	-	STRING	LAST
	totalTime	Total RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Number of updated rows of the SQL statement	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	ranges	Custom RT Range	Custom response time range	-	STRING	LAST
Summary (**total**: summary about SQL statement call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	errorCount	Errors	Total number of errors	-	INT	SUM
	readRowCount	Read Rows	Total number of read rows	-	INT	SUM
	totalTime	RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Total number of updated rows	-	INT	SUM

1.7.4.8 HBase Monitoring

This section describes the types, names, and meanings of HBase metrics collected by APM.

Table 1-28 HBase metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of HBase calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
HBase call monitoring (**client**)	namespaceTable	Namespace:Table name	Namespace and table name corresponding to the HBase operation	-	ENUM	LAST
	command	Command	Command run on the HBase server	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	queryRowCount	Read Rows	Number of read rows	-	INT	SUM
	updateRowCount	Updated Rows	Number of updated rows	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	totalTime	Total RT	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
HBase version (**version**)	version	Version	Version	-	STRING	LAST
HBase summary (**total**: summary of HBase call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	queryRowCount	Total Read Rows	Total number of read rows	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	updateRowCount	Total Updated Rows	Total number of updated rows	-	INT	SUM
	errorCount	Total Errors	Total number of errors	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
HBase node call monitoring (**serverNode**: HBase server RPC call statistics)	serverAddr	Server Node	Server node information	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	totalTime	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
HBase cluster call monitoring (**cluster**: HBase cluster RPC call information)	clusterId	Cluster ID	Cluster ID	-	ENUM	LAST
	cachedServers	Client Cache Node Address	Client cache node address	-	STRING	LAST
	zkNodes	ZooKeeper Connection Address	ZooKeeper connection address	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

1.7.4.9 Hikari Connection Pool Monitoring

This section describes the types, names, and meanings of Hikari connection pool metrics collected by APM.

Table 1-29 Hikari connection pool collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.1.0	-	Threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported.
Obtain Pool Info or Not	radio	JAVA	false	2.1.0	-	Whether to obtain pool information when getting connections

Table 1-30 Hikari connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Data source (**dataSource**)	url	url	url	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maximumPoolSize	Max. Connections Allowed	Maximum number of connections that are allowed	-	INT	LAST
	leakDetectionThreshold	Max. Pool Size	Maximum connection pool size	-	INT	LAST
	validationTimeout	Waiting Threads	Number of waiting threads	-	INT	LAST
	maxLifetime	Maximum Waiting Threads	Maximum number of waiting threads	-	INT	LAST
	poolingCount	Pool Connections	Number of connections in the pool	-	INT	LAST
	poolingPeak	Max. Connections	Maximum number of connections in the pool	-	INT	MAX
	activeCount	Active Connections	Number of active connections	-	INT	LAST
	activePeak	Max. Active Connections	Maximum number of active connections	-	INT	MAX
	logicConnectCount	Total Connections	Total number of connections	-	INT	SUM
	maxWait	Max. Waiting Time	Max. Waiting Time	ms	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	removeAbandoned	Automatically Reclaim Timeout Connections	Whether to automatically reclaim timeout connections	-	STRING	LAST
	removeAbandonedCount	Timeout Connection Reclaims	Number of times that timeout connections are reclaimed	-	INT	LAST
	removeAbandonedTimeoutMillis	Max. Connection Usage Duration	If a connection in the pool is not returned within the specified duration, the connection will be reclaimed.	-	INT	LAST
	testWhileIdle	Idle Connection Validity Check	Whether to verify the validity of an idle connection when an application applies for it from the pool	-	STRING	LAST
	testOnBorrow	Validity Check Before Obtaining Connection	Check whether a connection is valid before obtaining it from the connection pool.	-	STRING	LAST
	testOnReturn	Validity Check Upon Connection Return	Whether to verify the validity of a connection when it is returned	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	minEvictableIdleTimeMillis	Allowed Connection Idle Time	Idle time that is allowed for connections in the pool	ms	INT	LAST
	timeBetweenEvictionRunsMillis	Interval for Checking Idle Connection Validity	Interval for checking the validity of idle connections	-	INT	LAST
	driverName	Driver	Driver	-	STRING	LAST
	totalConnections	Total Connections	Total number of connections	-	INT	LAST
	activeConnections	Active Connections	Number of active connections	-	INT	LAST
	idleConnections	Idle Connections	Number of idle connections	-	INT	LAST
	threadsAwaitingConnection	Waiting Connections	Number of waiting connections	-	INT	LAST
Connection details (**connection**)	url	Connection Address	Connection address	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Exception (**exception**: Hikari call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Version (**version**)	version	Version	Version	-	STRING	LAST

1.7.4.10 Jetcd Monitoring

This section describes the types, names, and meanings of Jetcd metrics collected by APM.

Table 1-31 Jetcd collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Parse Value or Not	radio	JAVA	false	2.2.8	-	Whether to parse the value of the key-value pair. If it is not parsed, the value will be replaced with a question mark (?).

Table 1-32 Jetcd metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: Jetcd call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Jetcd UnaryRpc call monitoring (**naryRpc**)	endpoints	Cluster Address	Address of the etcd cluster	-	ENUM	LAST
	request	Request Type	Request type of the etcd API	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Jetcd Watch callback monitoring (**watcher**)	listener	listener	Listener class name corresponding to WatchImpl	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Jetcd KeepAlive callback monitoring (**KeepAlive**)	observers	observers	StreamObserver class name corresponding to KeepAlive	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Jetcd Election Observe callback monitoring (**election Observe**)	listener	Listener	Listener bound to the observe call	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Jetcd summary (**total**)	errorCount	Errors	Total number of errors	-	INT	SUM
	invokeCount	Calls	Total number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
Jetcd version (**version**)	version	Version	Version	-	STRING	LAST

1.7.4.11 MongoDB Monitoring

This section describes the types, names, and meanings of MongoDB metrics collected by APM.

Table 1-33 Collection parameters for MongoDB monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
TraceReportTimeSpanThreshold(ms)	integer	JAVA	1	2.1.13	-	Threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported.
isParseOriginalCommand	radio	JAVA	false	2.2.2	-	Indicates whether to collect original Mongo JSON commands.

Table 1-34 MongoDB metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection details (**connection**)	host	Connected Host	Connected host	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	openedCount	Opened Connections	Number of opened connections	-	INT	SUM
	closedCount	Closed Connections	Number of closed connections	-	INT	SUM
	idleClosedCount	Connections Closed Due to Idling	Number of connections that are closed due to long idle time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lifeClosedCount	Connections Closed Due to Keepalive Timeout	Number of connections that are closed due to keepalive timeout	-	INT	SUM
	errorClosedCount	Connections Closed Due to Errors	Number of connections that are closed due to errors	-	INT	SUM
	staleClosedCount	Connections Closed Due to Pool Clearing	Number of connections that are closed due to pool clearing	-	INT	SUM
	poolClosedClosedCount	Connections Closed Due to Pool Closure	Number of connections that are closed due to pool closure	-	INT	SUM
Exception (**exception**: exception statistics of MongoDB calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Cluster information (**clusterInfo**)	clusterId	Cluster ID	Cluster ID	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	hosts	Cluster Node Information Set on Client	Cluster node information set on the client	-	STRING	LAST
MongoDB call monitoring (**client**)	namespace	Namespace	Namespace corresponding to the MongoDB operation	-	ENUM	LAST
	command	Command	Command run on the MongoDB server	-	ENUM	LAST
	concurrentMax	Max. concurrency	Maximum concurrency	-	INT	MAX
	queryCount	Read Rows	Number of read rows	-	INT	SUM
	updateCount	Updated Rows	Number of updated rows	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
MongoDB version (**version**)	version	Version	Version	-	STRING	LAST
MongoDB summary (**total**: summary of MongoDB call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	queryCount	Total Read Rows	Total number of read rows	-	INT	SUM
	updateCount	Total Updated Rows	Total number of updated rows	-	INT	SUM
	errorCount	Total Errors	Total number of errors	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time	-	INT	SUM
MongoDB cluster call (**cluster**)	nodes	Cluster Node	Cluster node information	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Connection pool monitoring (**connectionPool**)	host	Connection Address	Connection address	-	ENUM	LAST
	maxSize	Max. Pool Size	Maximum connection pool size	-	INT	AVG
	minSize	Min. Pool Size	Minimum connection pool size	-	INT	AVG
	availableCount	Idle Connections	Number of idle connections	-	INT	AVG
	inUseCount	Active Connections	Number of active connections	-	INT	AVG
	maxWaitTimeMs	Max. Waiting Time (ms)	Maximum waiting time of a connection (ms)	-	INT	AVG
	maxConnectionLifeTimeMs	Max. Keepalive Time	Maximum keepalive time of a connection	-	INT	AVG
	maxConnectionIdleTimeMs	Max. Idle Time	Maximum idle time of a connection	-	INT	AVG

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
MongoDB node call monitoring (**serverNode**)	serverAddr	Node Address	Node address	-	ENUM	LAST
	type	Node Type	Node type	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

1.7.4.12 MySQL Database

This section describes the types, names, and meanings of MySQL database metrics collected by APM.

Table 1-35 MySQL database collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Collect Original SQL	radio	JAVA	false	2.0.0	-	Whether to collect and report original SQL statements
shardTableName	array	JAVA	-	2.2.2	-	Table name specified for SQL statement aggregation. Tables starting with this name will be aggregated into the same table.

1.7.4.13 ObsClient Monitoring

This section describes the types, names, and meanings of ObsClient metrics collected by APM.

Table 1-36 ObsClient metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of ObsClient calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
URL monitoring (**obsClientInvocation**: APM counts URL call statistics by URL.)	client	client	client	-	ENUM	LAST
	url	url	Called URL	-	ENUM	LAST
	method	HTTP Method	HTTP method of the URL	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the URL	-	INT	MAX
	errorCount	Errors	Number of call errors of the URL	-	INT	SUM
	hostUri	hostUri	hostUri	-	STRING	LAST
	invokeCount	Calls	Number of times that the URL is called	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the called URL	-	INT	MAX
	responseCloseCount	Closed Responses	Number of responses that are closed	-	INT	SUM
	totalTime	Total RT	Total response time of the called URL	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Status code statistics (**code**: APM counts URL call statistics by status code.)	code	Status Code	Status code	-	ENUM	LAST
	url	URL	URL that returns the status code	-	STRING	LAST
	count	Count	Number of times that the status code has occurred	-	INT	SUM
ObsClient summary (**total**)	errorCount	Total Request Errors	Total number of request errors	-	INT	SUM
	invokeCount	Calls	Total number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM

1.7.4.14 Oracle Database

This section describes the types, names, and meanings of Oracle database metrics collected by APM.

Table 1-37 Oracle database collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Collect Original SQL Statement or Not	radio	JAVA	false	2.2.9	-	Whether to collect and report original SQL statements

Table 1-38 Oracle database metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Database connection (**connection**: APM counts SQL call statistics by database.)	db	Database	Database name	-	ENUM	LAST
	createdCount	Created Connections	Number of connections created by the database	-	INT	SUM
	currentCount	Current Connections	Current number of connections of the database	-	INT	SUM
	destroyedCount	Destroyed Connections	Number of the database's connections that have been destroyed	-	INT	SUM
	errorCount	Errors	Number of errors that the database encounters	-	INT	SUM
	invokeCount	Calls	Number of times that the database is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the database	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	ranges	Custom RT Range	Custom response time range	-	STRING	LAST
	readRowCount	Read Rows	Number of rows read from the database	-	INT	SUM
	updatedRowCount	Updated Rows	Number of rows updated in the database	-	INT	SUM
	totalTime	Total RT	Total response time of the database	-	INT	SUM
	slowestSql	Slowest SQL	Slowest SQL statement of the database in the collection period	-	STRING	LAST
Exception (**exception**: APM counts SQL call statistics by database.)	causeType	Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Message	Exception message	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	sql	Exception SQL	SQL statement that encounters an exception	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Version (**version**: Oracle package version)	version	Version	Driver package version	-	STRING	LAST
SQL monitoring (**sql**: APM counts call statistics by SQL.)	sql	SQL ID	Unique ID of the SQL statement, which is used for alarm configuration	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the SQL statement	-	INT	MAX
	errorCount	Errors	Number of errors that the SQL statement encounters	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of times that the SQL statement is called	-	INT	SUM
	lastError	Error Message	SQL error information	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the SQL statement	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	readRowCount	Read Rows	Number of read rows of the SQL statement	-	INT	SUM
	runningCount	Ongoing Executions	Number of SQL statements that are being executed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	sqlString	SQL Statement	SQL statement	-	STRING	LAST
	totalTime	Total RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Number of updated rows of the SQL statement	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	ranges	Custom RT Range	Custom response time range	-	STRING	LAST
Summary (**total**: summary about SQL statement call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	errorCount	Errors	Total number of errors	-	INT	SUM
	readRowCount	Read Rows	Total number of read rows	-	INT	SUM
	totalTime	RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Total number of updated rows	-	INT	SUM

1.7.4.15 PostgreSQL Database

This section describes the types, names, and meanings of PostgreSQL database metrics collected by APM.

Table 1-39 PostgreSQL database collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Collect Original SQL Statement or Not	radio	JAVA	false	2.0.0	-	Whether to collect and report original SQL statements

Table 1-40 PostgreSQL database metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Database connection (**connection**: APM counts SQL call statistics by database.)	db	Database	Database name	-	ENUM	LAST
	createdCount	Created Connections	Number of connections created by the database	-	INT	SUM
	currentCount	Current Connections	Current number of connections of the database	-	INT	SUM
	destroyedCount	Destroyed Connections	Number of the database's connections that have been destroyed	-	INT	SUM
	errorCount	Errors	Number of errors that the database encounters	-	INT	SUM
	invokeCount	Calls	Number of times that the database is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the database	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	readRowCount	Read Rows	Number of rows read from the database	-	INT	SUM
	updatedRowCount	Updated Rows	Number of rows updated in the database	-	INT	SUM
	totalTime	Total RT	Total response time of the database	-	INT	SUM
	slowestSql	Slowest SQL	Slowest SQL statement of the database in the collection period	-	STRING	LAST
Exception (**exception**: exception statistics about SQL calls)	causeType	Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Message	Exception message	-	STRING	LAST
	sql	Exception SQL	SQL statement that encounters an exception	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Version (**version**: PostgreSQL package version)	version	Version	Driver package version	-	STRING	LAST
SQL monitoring (**sql**: APM counts call statistics by SQL.)	sql	SQL ID	Unique ID of the SQL statement, which is used for alarm configuration	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the SQL statement	-	INT	MAX
	errorCount	Errors	Number of errors that the SQL statement encounters	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of times that the SQL statement is called	-	INT	SUM
	lastError	Error Message	SQL error information	-	STRING	LAST
	maxTime	Max. RT	Maximum response time of the SQL statement	-	INT	MAX
	readRowCount	Read Rows	Number of read rows of the SQL statement	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of SQL statements that are being executed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	sqlString	SQL Statement	SQL statement	-	STRING	LAST
	totalTime	Total RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Number of updated rows of the SQL statement	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Summary (**total**: summary about SQL statement call statistics)	invokeCount	Calls	Total number of calls	-	INT	SUM
	errorCount	Errors	Total number of errors	-	INT	SUM
	readRowCount	Read Rows	Total number of read rows	-	INT	SUM
	totalTime	RT	Total response time	-	INT	SUM
	updatedRowCount	Updated Rows	Total number of updated rows	-	INT	SUM

1.7.5 URLs

1.7.5.1 CSEProvider Cluster Monitoring

This section describes the types, names, and meanings of CSEProvider cluster metrics collected by APM.

Table 1-41 CSEProvider collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Blocklist Configuration	obj_array	JAVA	-	2.0.0	-	URLs in the blacklist will not be collected. There are four modes: "startwith", "endwith", "include", and "regex".

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Max. Status Code Length	integer	JAVA	0	2.0.0	-	The system parses the body content within the specified length and obtains the corresponding service status code.
Key for Status Code Parsing	array	JAVA	-	2.0.0	-	Key based on which the corresponding body content is to be obtained and reported as the service status code
Normal Status Code	array	JAVA	-	2.0.0	-	If the obtained status code is not within the range, the request is regarded as an error trace.
Slow Request Threshold	integer	JAVA	800	2.0.0	-	Slow request threshold. If the threshold is crossed, a URL will be regarded as a slow URL. The system will automatically increase the sampling ratio for it.

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Slow URL Threshold	obj_array	JAVA	-	2.0.0	-	Slow request threshold. If this threshold is crossed, the URL is defined as a slow URL. In that case, the system automatically increases the sampling ratio for it. There are four sampling policies: 1. Full sampling; 2. Percentage sampling; 3. Fixed-quantity sampling per minute; 4. Automatic sampling.
Key for Header Value Interception	array	JAVA	-	2.0.0	-	Key based on which header value content is to be intercepted

Table 1-42 CSEProvider cluster metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
CSEProvider cluster monitoring (**cluster**: APM counts call statistics based on the ID of the caller's cluster.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM
	invokeCount	Calls	Number of cluster calls	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	Max. RT	Maximum response time for calling the cluster	ms	INT	MAX
	totalTime	Total RT	Total response time for calling the cluster	ms	INT	SUM
CSEProvider call details (**detail**: APM counts call statistics by URL.)	qualifiedName	Call URL	Called URL of CSEProvider	-	ENUM	LAST
	method	HTTP Method	HTTP method of the called CSEProvider URL	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the CSEProvider URL	-	INT	MAX
	errorCount	Errors	Number of errors occur when the CSEProvider URL is called	-	INT	SUM
	invokeCount	Calls	Number of times that the CSEProvider URL is called	-	INT	SUM
	lastError	Error Message	Call error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for calling the CSEProvider URL	ms	INT	MAX
	totalTime	Total RT	Total response time for calling the CSEProvider URL	ms	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
CSEProvider summary (**total**: summary of all URL statistics)	errorCount	Errors	Total number of CSEProvider call errors	-	INT	SUM
	invokeCount	Calls	Total number of CSEProvider calls	-	INT	SUM
	totalTime	Total RT	Total response time of calling CSEProvider	ms	INT	SUM
Status code monitoring (**statuscode**: APM counts URL call statistics based on the status code returned.)	code	Status Code	HTTP status code	-	ENUM	LAST
	count	Count	Number of times that the status code has occurred	-	INT	SUM
	url	Sample URL	Sample URL which returns the status code	-	STRING	LAST

1.7.5.2 DubboProvider Monitoring

This section describes the types, names, and meanings of DubboProvider metrics collected by APM.

Table 1-43 Dubbo server metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Method call (**invocation**)	serviceUniqueName	serviceUniqueName	Unique service identifier (group+interface+version)	-	ENUM	LAST
	method	method	Method	-	ENUM	LAST
	source	Source	Call source	-	ENUM	LAST
	lastError	lastError	Error message	-	STRING	LAST
	slowTraceId	slowTraceId	Slowest trace ID	-	STRING	LAST
	errorTraceId	errorTraceId	Error trace ID	-	STRING	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM
	invokeCount	invokeCount	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
Host summary (**cluster**)	cluster	cluster	Host	-	ENUM	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM
	invokeCount	invokeCount	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
Return code summary (**resultCode**)	code	code	Return code	-	ENUM	LAST
	count	count	Number of calls	-	INT	SUM
	lastMethod	lastMethod	Last exception type	-	STRING	LAST
Summary (**total**)	lastError	lastError	Error message	-	STRING	LAST
	slowTraceId	slowTraceId	Slowest trace ID	-	STRING	LAST
	errorTraceId	errorTraceId	Error trace ID	-	STRING	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM
	invokeCount	invokeCount	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
Thread pool (**threadPool**)	poolId	poolId	Unique ID of a thread pool	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	poolType	poolType	Custom Dubbo thread pool type, such as fixed, cached, or limited	-	STRING	LAST
	activeCount	activeCount	Number of active threads	-	INT	SUM
	corePoolSize	corePoolSize	Number of core threads	-	INT	SUM
	maximumPoolSize	maximumPoolSize	Maximum number of core threads	-	INT	SUM
	poolSize	poolSize	Size of the thread pool	-	INT	SUM
	queueSize	queueSize	Size of the waiting queue	-	INT	SUM
	taskCount	taskCount	Number of tasks	-	INT	SUM
Client version (**version**)	version	version	Client version	-	STRING	LAST

1.7.5.3 FunctionGraph Monitoring

This section describes the types, names, and meanings of FunctionGraph metrics collected by APM.

Table 1-44 Collection parameters for FunctionGraph monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Sampling Type	radio	JAVA	4	1.0.0	-	Sampling type. Options: full sampling, percentage sampling, fixed-quantity sampling per minute, and intelligent sampling (default).
Sampling Ratio	integer	JAVA	10	1.0.0	-	Percentage of samples to the total number of trace data records
Samples/Minute	integer	JAVA	1000	1.0.0	-	Number of trace data records collected every minute.
Slow Request Threshold	integer	JAVA	800	2.0.0	-	Slow request threshold. If the threshold is crossed, the method is regarded as a slow method. In that case, the trace sampling ratio will be increased by default.

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Method Configuration	obj_array	JAVA	-	2.0.0	-	Configure the slow request threshold and sampling ratio for each method separately. The following sampling policies can be set: percentage sampling, fixed-quantity sampling per minute, and automatic sampling.

Table 1-45 FunctionGraph metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Method monitoring (**detail**: APM counts URL call statistics by function method.)	method	Method	Request method	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the method in a collection period	ms	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	totalTime	Total RT	Total response time of the method	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Cluster call (**cluster**: APM counts URL call statistics based on the cluster ID of the caller.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	totalTime	Total RT	Total response time for calling the cluster	ms	INT	SUM
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM

1.7.5.4 URL Monitoring

This section describes the types, names, and meanings of URL metrics collected by APM.

Table 1-46 Collection parameters for URL monitoring

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Key for Header Value Interception	array	JAVA	-	2.0.0	-	Key based on which header value content is to be intercepted
Key for Parameter Value Interception	array	JAVA	-	2.0.0	-	Key based on which parameter value content is to be intercepted

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Key for Cookie Value Interception	array	JAVA	-	2.0.0	-	Key based on which cookie value content is to be intercepted
URL Collection Configuration	obj_array	JAVA	-	2.0.0	-	URL collection configuration, based on which RESTful URLs are normalized. There are four modes: "startwith", "endwith", "include", and "regex".
Blocklist Configuration	obj_array	JAVA	-	2.0.0	-	URLs that match the specified rule will not be collected. There are four modes: "startwith", "endwith", "include", and "regex".
Service Code Length	integer	JAVA	0	2.0.0	-	Maximum length of the body content to be collected for service code parsing

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Key for Service Code Interception	array	JAVA	-	2.0.0	-	Specify a key. Then the system parses the key in the body (JSON) and obtains the service status code based on the key.
Normal Service Code	array	JAVA	-	2.0.0	-	Normal service code range. If a service code is not within this range, the corresponding trace is regarded as an error trace.
Slow Request Threshold	integer	JAVA	800	2.0.0	-	Slow request threshold. If the threshold is crossed, the URL is regarded as a slow URL. In that case, the trace sampling ratio will be increased by default.

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
URL Configuration	obj_array	JAVA	-	2.0.0	-	Configure the slow request threshold and sampling ratio for each URL separately. The following sampling policies can be set: percentage sampling, fixed-quantity sampling per minute, and automatic sampling.
Error Code	radio	JAVA	500	2.0.0	-	Status codes that are counted as errors
Auto URL Normalization	radio	JAVA	false	2.3.11	-	Whether the URL will be automatically normalized

1.7.6 External Calls

1.7.6.1 ApacheHttpAsyncClient Connection Pool

This section describes the types, names, and meanings of ApacheHttpAsyncClient connection pool metrics collected by APM.

Table 1-47 ApacheHttpAsyncClient collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Threshold (ms) for Reporting Connection Trace	integer	JAVA	1	2.1.6	-	Threshold (ms) for reporting connection traces
Obtain Pool Info or Not	radio	JAVA	1	2.1.6	-	Whether to obtain pool information when getting connections

Table 1-48 ApacheHttpAsyncClient connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection pool (**connectionPool**: statistics about ApacheHttpAsyncClient connections in different states)	poolId	Connection Pool ID	ApacheHttpAsyncClient connection pool ID	-	ENUM	LAST
	available	Idle Connections	Number of idle connections in the connection pool	-	INT	SUM
	leased	Occupied Connections	Number of connections occupied	-	INT	SUM
	max	Max. Connections	Maximum number of connections in the connection pool	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	pending	Pending Connections	Number of pending connections in the connection pool	-	INT	SUM
Connection pool route (**collectionPoolRoute**: APM counts connection statistics by pool route.)	poolId	Connection Pool ID	ApacheHttpAsyncClient connection pool ID	-	ENUM	LAST
	route	Route	Routing information of the connection pool	-	ENUM	LAST
	available	Idle Connections	Number of idle connections in the connection pool	-	INT	SUM
	leased	Occupied Connections	Number of connections occupied	-	INT	SUM
	max	Max. Connections	Maximum number of connections in the connection pool	-	INT	MAX
	pending	Pending Connections	Number of pending connections in the connection pool	-	INT	SUM
Connection details (**connection**)	route	Route	Route	-	ENUM	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total Time	Total time	-	INT	SUM
	errorCount	Errors	Number of errors	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX

1.7.6.2 ApacheHttpClient Connection Pool

This section describes the types, names, and meanings of ApacheHttpClient connection pool metrics collected by APM.

Table 1-49 ApacheHttpClient connection pool metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection pool (**connectionPool**: statistics about ApacheHttpclient connections in different states)	poolId	Connection Pool ID	ApacheHttpclient connection pool ID	-	ENUM	LAST
	available	Idle Connections	Number of idle connections in the connection pool	-	INT	SUM
	leased	Occupied Connections	Number of connections occupied	-	INT	SUM
	max	Max. Connections	Maximum number of connections in the connection pool	-	INT	MAX
	pending	Pending Connections	Number of pending connections in the connection pool	-	INT	SUM
Connection pool route (**collectionPoolRoute**: APM counts connection statistics by pool route.)	poolId	Connection Pool ID	ApacheHttpClient connection pool ID	-	ENUM	LAST
	route	Route	Routing information of the connection pool	-	ENUM	LAST
	available	Idle Connections	Number of idle connections in the connection pool	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	leased	Occupied Connections	Number of connections occupied	-	INT	SUM
	max	Max. Connections	Maximum number of connections in the connection pool	-	INT	MAX
	pending	Pending Connections	Number of pending connections in the connection pool	-	INT	SUM

1.7.6.3 CSEConsumer Cluster Monitoring

This section describes the types, names, and meanings of CSEConsumer cluster metrics collected by APM.

Table 1-50 CSEConsumer cluster metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
CSEConsumer cluster monitoring (**cluster**: APM counts call statistics based on the ID of the cluster called by CSEConsumer.)	clusterId	Cluster ID	ID of the cluster where the called service is located	-	ENUM	LAST
	errorCount	Errors	Number of errors	-	INT	SUM
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time for calling the cluster	ms	INT	MAX
	totalTime	Total RT	Total response time for calling the cluster	ms	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
CSEConsumer call details (**detail**: APM counts the call statistics based on the called URL.)	qualifiedName	Call URL	CSEConsumer call URL	-	ENUM	LAST
	method	HTTP Method	HTTP method for CSEConsumer calling	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum number of concurrent CSEConsumer calls	-	INT	MAX
	errorCount	Errors	Number of CSEConsumer call errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the error trace in a collection period	-	STRING	LAST
	slowTraceId	Slowest Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of CSEConsumer calls	-	INT	SUM
	lastError	Error Message	Call error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for CSEConsumer calling	ms	INT	MAX
	totalTime	Total RT	Total response time for CSEConsumer calling	ms	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
CSEConsumer summary (**total**: summary of CSEConsumer call statistics)	errorCount	Errors	Total number of CSEConsumer call errors	-	INT	SUM
	invokeCount	Calls	Total number of CSEConsumer calls	-	INT	SUM
	totalTime	Total RT	Total response time for CSEConsumer calling	-	INT	SUM

1.7.6.4 DubboConsumer Monitoring

This section describes the types, names, and meanings of DubboConsumer metrics collected by APM.

Table 1-51 DubboConsumer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Method call (**invocation**)	serviceUniqueName	serviceUniqueName	Unique service identifier (group+interface+version)	-	ENUM	LAST
	method	method	Method	-	ENUM	LAST
	lastError	lastError	Error message	-	STRING	LAST
	slowTraceId	slowTraceId	Slowest trace ID	-	STRING	LAST
	errorTraceId	errorTraceId	Error trace ID	-	STRING	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM
	invokeCount	invokeCount	Number of calls	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
	source	Source	Call source	-	ENUM	LAST
Host summary (**cluster**)	cluster	cluster	Host	-	ENUM	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	invokeCount	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
Return code summary (**resultCode**)	code	code	Return code	-	ENUM	LAST
	count	count	Number of calls	-	INT	SUM
	lastMethod	lastMethod	Last exception type	-	STRING	LAST
Summary (**total**)	lastError	lastError	Error message	-	STRING	LAST
	slowTraceId	slowTraceId	Slowest trace ID	-	STRING	LAST
	errorTraceId	errorTraceId	Error trace ID	-	STRING	LAST
	range1	range1	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	range2	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	range3	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	range4	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	range5	Number of requests with 1–10s response time	-	INT	SUM
	range6	range6	Number of requests with response time longer than 10s	-	INT	SUM
	invokeCount	invokeCount	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	errorCount	Number of errors	-	INT	SUM
	runningCount	runningCount	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	concurrentMax	Maximum concurrency	-	INT	MAX
Thread pool (**threadPool**)	poolId	poolId	Unique ID of a thread pool	-	ENUM	LAST
	poolType	poolType	Custom Dubbo thread pool type, such as fixed, cached, or limited	-	STRING	LAST
	activeCount	activeCount	Number of active threads	-	INT	SUM
	corePoolSize	corePoolSize	Number of core threads	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maximumPoolSize	maximumPoolSize	Maximum number of core threads	-	INT	SUM
	poolSize	poolSize	Size of the thread pool	-	INT	SUM
	queueSize	queueSize	Size of the waiting queue	-	INT	SUM
	taskCount	taskCount	Number of tasks	-	INT	SUM
Client version (**version**)	version	version	Version	-	STRING	LAST

1.7.6.5 HttpClient Monitoring

This section describes the types, names, and meanings of HttpClient metrics collected by APM.

Table 1-52 HttpClient collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
URL Normalization Configuration	obj_array	JAVA	-	2.0.0	-	URL normalization configuration, based on which some RESTful URLs are normalized. There are four modes: "startwith", "endwith", "include", and "regex".

Table 1-53 HttpClient metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: HttpClient call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
Cluster metrics (**hostInvocation**: APM counts HttpClient URL call statistics by the called party's cluster.)	envId	Cluster ID	Cluster ID of the called party	-	ENUM	LAST
	hostUri	Called Address	Called address	-	STRING	LAST
	errorCount	Errors	Number of errors that occur when the cluster URL is called	-	INT	SUM
	invokeCount	Calls	Number of times that the cluster URL is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time for calling the cluster URL	ms	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time for calling the cluster URL	ms	INT	SUM
	responseCloseCount	Closed Responses	Number of closed responses when the cluster URL is called	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
HttpClient version (**info**: HttpClient package version statistics)	httpClientVersion	HttpClient Version	Version of the HttpClient package	-	STRING	LAST
	httpCoreVersion	HttpCore Version	Version of the HttpCore package	-	STRING	LAST
URL monitoring (**invocation**: APM counts URL call statistics by URL.)	url	url	Called URL	-	ENUM	LAST
	method	HTTP Method	HTTP method of the URL	-	ENUM	LAST
	client	Client Type	HTTP client type	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the URL	-	INT	MAX
	errorCount	Errors	Number of call errors of the URL	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	hostUri	Called Address	Called URL address	-	STRING	LAST
	invokeCount	Calls	Number of times that the URL is called	-	INT	SUM
	lastError	Error Message	Error details	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	Max. RT	Maximum response time of the called URL	ms	INT	MAX
	responseCloseCount	responseCloseCount	Number of closed responses when the URL is called	-	INT	SUM
	totalTime	Total RT	Total response time of the called URL	ms	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	envId	Cluster ID	Cluster ID corresponding to the called URL	-	STRING	LAST
HttpClient summary (**total**: summary about HttpClient call statistics)	errorCount	Errors	Total number of errors	-	INT	SUM
	invokeCount	Calls	Total number of calls	-	INT	SUM
	responseCloseCount	Closed Responses	Total number of responses that are closed	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
Status code statistics (**code**: HttpClient call exception statistics)	code	Status Code	Status code	-	ENUM	LAST
	url	URL	URL that returns the status code	-	STRING	LAST
	count	Count	Number of times that the status code has occurred	-	INT	SUM

1.7.7 Cache

1.7.7.1 Redis Method Call

This section describes the types, names, and meanings of Redis method call metrics collected by APM.

Table 1-54 Redis method call collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Parameter Parsing	radio	JAVA	false	2.0.0	-	Whether to parse Redis parameters and return values
Length	integer	JAVA	1000	2.0.0	-	Maximum length of parameters to be parsed
Port Differentiation	radio	JAVA	false	2.0.0	-	Whether to distinguish Redis ports

Table 1-55 Call metrics

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Call details (**detail**)	host	Host	Host	-	ENUM	LAST
	action	Method	Method	-	ENUM	LAST
	lastError	Error Message	Error message	-	STRING	LAST
	slowTraceId	Slowest Trace ID	Slowest trace ID	-	STRING	LAST
	errorTraceId	Error Trace ID	Error trace ID	-	STRING	LAST
	range1	0–5 ms	Number of requests with 0–5 ms response time	-	INT	SUM

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	5–10 ms	Number of requests with 5–10 ms response time	-	INT	SUM
	range3	10–50 ms	Number of requests with 10–50 ms response time	-	INT	SUM
	range4	50–100 ms	Number of requests with 50–100 ms response time	-	INT	SUM
	range5	100–1000 ms	Number of requests with 100–1000 ms response time	-	INT	SUM
	range6	> 1s	Number of requests with response time longer than 1s	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	hits	Hits	Hits of methods including GET, HGET, and EXPIRE	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	runningCount	Ongoing Executions	Number of tasks that are being executed	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	blobCount	Calls with Large Field Returned	Number of calls with more than 1000 bytes returned	-	INT	SUM
	getInvokeCount	GET Calls	Number of times that GET methods including GET, HGET, and EXPIRE have been called	-	INT	SUM
	traffic	Traffic	Call traffic	-	INT	SUM
Host summary (**host**)	host	Host	Host	-	ENUM	LAST
	lastError	Error Message	Error message	-	STRING	LAST
	slowTraceId	Slowest Trace ID	Slowest trace ID	-	STRING	LAST
	errorTraceId	Error Trace ID	Error trace ID	-	STRING	LAST
	range1	0–5 ms	Number of requests with 0–5 ms response time	-	INT	SUM
	range2	5–10 ms	Number of requests with 5–10 ms response time	-	INT	SUM
	range3	10–50 ms	Number of requests with 10–50 ms response time	-	INT	SUM
	range4	50–100 ms	Number of requests with 50–100 ms response time	-	INT	SUM
	range5	100–1000 ms	Number of requests with 100–1000 ms response time	-	INT	SUM

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 1s	Number of requests with response time longer than 1s	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	hits	Hits	Hits of methods including GET, HGET, and EXPIRE	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	runningCount	Ongoing Executions	Number of tasks that are being executed	-	INT	SUM
	blobCount	Calls with Large Field Returned	Number of calls with more than 1000 bytes returned	-	INT	SUM
	getInvokeCount	GET Calls	Number of times that GET methods including GET, HGET, and EXPIRE have been called	-	INT	SUM
	traffic	Traffic	Call traffic	-	INT	SUM
Method summary (**action**)	action	Method	Method	-	ENUM	LAST
	lastError	Last Exception Type	Last exception type	-	STRING	LAST

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	slowTraceId	Slowest Trace ID	Slowest trace ID	-	STRING	LAST
	errorTraceId	Error Trace ID	Error trace ID	-	STRING	LAST
	range1	0–5 ms	Number of requests with 0–5 ms response time	-	INT	SUM
	range2	5–10 ms	Number of requests with 5–10 ms response time	-	INT	SUM
	range3	10–50 ms	Number of requests with 10–50 ms response time	-	INT	SUM
	range4	50–100 ms	Number of requests with 50–100 ms response time	-	INT	SUM
	range5	100–1000 ms	Number of requests with 100–1000 ms response time	-	INT	SUM
	range6	> 1s	Number of requests with response time longer than 1s	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	hits	Hits	Hits of methods including GET, HGET, and EXPIRE	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Ongoing executions	-	INT	SUM
	blobCount	Calls with Large Field Returned	Number of calls with more than 1000 bytes returned	-	INT	SUM
	getInvokeCount	GET Calls	Number of times that GET methods including GET, HGET, and EXPIRE have been called	-	INT	SUM
	traffic	Traffic	Traffic	-	INT	SUM
Summary (**total**)	lastError	Last Exception Type	Last exception type	-	STRING	LAST
	slowTraceId	Slowest Trace ID	Slowest trace ID	-	STRING	LAST
	errorTraceId	Error Trace ID	Error trace ID	-	STRING	LAST
	range1	0–5 ms	Number of requests with 0–5 ms response time	-	INT	SUM
	range2	5–10 ms	Number of requests with 5–10 ms response time	-	INT	SUM
	range3	10–50 ms	Number of requests with 10–50 ms response time	-	INT	SUM
	range4	50–100 ms	Number of requests with 50–100 ms response time	-	INT	SUM

Name	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	100–1000 ms	Number of requests with 100–1000 ms response time	-	INT	SUM
	range6	> 1s	Number of requests with response time longer than 1s	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	hits	Hits	Hits of methods including GET, HGET, and EXPIRE	-	INT	SUM
	totalTime	Total RT	Total response time	ms	INT	SUM
	maxTime	Max. RT	Maximum response time	ms	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	runningCount	Ongoing Executions	Number of tasks that are being executed	-	INT	SUM
	blobCount	Calls with Large Field Returned	Number of calls with more than 1000 bytes returned	-	INT	SUM
	getInvokeCount	GET Calls	Number of times that GET methods including GET, HGET, and EXPIRE have been called	-	INT	SUM
	traffic	Traffic	Traffic	-	INT	SUM

1.7.7.2 Jedis Monitoring

This section describes the types, names, and meanings of Jedis metrics collected by APM.

Table 1-56 Jedis metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection pool (**jedisPool**)	pool	Pool	Unique identifier of a connection pool (host name +port number)	-	ENUM	LAST
	maxTotal	maxTotal	Maximum number of connections	-	INT	MAX
	maxIdle	maxIdle	Maximum number of idle connections	-	INT	MAX
	minIdle	minIdle	Minimum number of idle connections	-	INT	MIN
	numActive	numActive	Number of active connections	-	INT	SUM
	numIdle	numIdle	Number of idle connections	-	INT	SUM
	numWaiters	numWaiters	Number of waiting connections	-	INT	SUM
	createdCount	createdCount	Number of connections that have been created	-	INT	SUM
	destroyedCount	destroyedCount	Number of connections that have been destroyed	-	INT	SUM
	borrowedCount	borrowedCount	Number of borrowed connections	-	INT	SUM
	maxWaitMillis	maxWaitMillis	Maximum waiting time (ms)	ms	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxBorrowWaitTimeMillis	maxBorrowWaitTimeMillis	Maximum waiting time of borrowed connections (ms)	ms	INT	MAX
	meanActiveTimeMillis	meanActiveTimeMillis	Average activation time of connections (ms)	ms	INT	SUM
	meanBorrowWaitTimeMillis	meanBorrowWaitTimeMillis	Average waiting time of borrowed connections	ms	INT	SUM
Active/standby switchover (**switch**)	from	from	Source host	-	STRING	LAST
	to	to	Target host	-	STRING	LAST
	switchTimes	switchTimes	Number of switchovers	-	INT	SUM
Client information (**clientInfo**)	version	version	Client version	-	STRING	LAST
	mode	mode	Redis mode (standalone or cluster)	-	STRING	LAST
	nodes	nodes	Number of master Redis nodes	-	STRING	LAST

1.7.7.3 Lettuce Client

This section describes the types, names, and meanings of Lettuce client metrics collected by APM.

Table 1-57 Lettuce client metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Client information (**clientInfo**)	version	version	Client version	-	STRING	LAST
	mode	Mode	Redis mode (standalone or cluster)	-	STRING	LAST
	nodes	nodes	Number of master Redis nodes	-	STRING	LAST
Active/standby switchover (**switch**)	from	from	Source host	-	STRING	LAST
	to	to	Target host	-	STRING	LAST
	switchTimes	switch Times	Number of switchovers	-	INT	SUM

1.7.8 Agent Monitoring

This section describes the types, names, and meanings of Agent monitoring metrics collected by APM.

Table 1-58 Agent monitoring metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Probe data (**detail**: probe data metric set)	type	Data Type	Type of data reported by the probe	-	ENUM	LAST
	discardBytes	Discarded Bytes	Number of discarded bytes	Byte	INT	SUM
	discardCount	Discard Times	Number of times that the type of data is discarded	-	INT	SUM
	errorBytes	Bytes Not Sent	Number of bytes that fail to be sent	Byte	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Send Failures	Number of times that the type of data fails to be sent	-	INT	SUM
	maxBytes	Max. Bytes	Maximum number of sent bytes	Byte	INT	MAX
	maxQueueSize	Max. Queue Size	Maximum length of the sending queue	-	INT	MAX
	sendBytes	Sent Bytes	Number of successfully sent bytes	Byte	INT	SUM
	sendCount	Successful Send Times	Number of times that the type of data is successfully sent	-	INT	SUM
	sendTotalTime	Total Send Time	Total sending time of the data type	ms	INT	SUM
	slowTime	Max. Send Time	Maximum sending time of the data type	ms	INT	MAX
Exception (**exception**: exception metric set)	causeType	Exception Class	Exception class	-	ENUM	LAST
	type	Type	Exception type	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Message	Exception message	-	STRING	LAST
	stackTrace	Stack	Exception stack	-	CLOB	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Server connection monitoring (**transfer**: server connection monitoring metric set)	host	Host	Host information	-	ENUM	LAST
	connectIp	Connection IP Address	Connection IP address	-	STRING	LAST
	ipList	IP Addresses	All IP addresses	-	STRING	LAST
	isConnected	Connected or Not	Connected or not	-	INT	LAST
	rt	RT	Response time	ms	INT	AVG
Queue monitoring (**repository**: queue monitoring metric set)	monitorQueueSize	Size of Monitoring Data Queue	Size of the monitoring data queue	-	INT	SUM
	monitorObjectSize	Memory Size of Monitoring Data	Memory size of the monitoring data	-	INT	SUM
	traceQueueSize	Size of Trace Data Queue	Size of the trace data queue	-	INT	SUM
	traceObjectSize	Memory Size of Trace Data	Memory size of the trace data	-	INT	SUM

1.7.9 Tomcat Monitoring

This section describes the types, names, and meanings of Tomcat metrics collected by APM.

Table 1-59 Tomcat metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Tomcat information (**tomcatInfo**: Tomcat package version statistics)	tomcatVersion	Tomcat Version	Tomcat version	-	STRING	LAST
Tomcat port monitoring (**tomcat**: APM counts statistics of Tomcat threads and connections by Tomcat port.)	name	Port Name	Port name	-	ENUM	LAST
	currentThreadCount	Current Threads	Number of current threads on the port	-	INT	AVG
	currentThreadsBusy	Busy Threads	Number of busy threads on the port at the time of collection	-	INT	AVG
	currentThreadsBusyMax	Max. Busy Threads	Maximum number of busy threads on the port in a collection period	-	INT	MAX
	maxThreads	Max. Threads	Maximum number of threads on the port	-	INT	MAX
	maxConnections	Max. Connections	Maximum number of connections on the port	-	INT	MAX
	connectionCount	Current Connections	Number of current connections of the port at the time of collection	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	connectionCountMax	Max. Connections	Maximum number of connections on the port in a collection period	-	INT	MAX

1.7.10 Message Queues

1.7.10.1 KafkaConsumer Monitoring

This section describes the types, names, and meanings of KafkaConsumer metrics collected by APM.

Table 1-60 KafkaConsumer monitoring collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Kafka Slow Request Threshold	integer	JAVA	800	2.1.14	-	The sampling ratio will increase if the slow request threshold is crossed.
Kafka Consumption Method Configuration	obj_array	JAVA	-	2.1.14	-	Kafka consumption method configuration

Table 1-61 KafkaConsumer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Topic (**topic**: Kafka topic monitoring data)	id	id	Client ID and IP address	-	ENUM	LAST
	topic	topic	Kafka topic name	-	ENUM	LAST
	bytesConsumedRate	Bytes Consumed/s	Number of bytes consumed per second	Byte	INT	AVG
	fetchSizeAvg	Avg. Bytes Fetched	Average number of bytes fetched for a request	Byte	INT	AVG
	fetchSizeMax	Max. Bytes Fetched	Maximum number of bytes fetched for a request	Byte	INT	MAX
	recordsConsumedRate	Messages Consumed/s	Number of messages consumed per second	-	INT	AVG
	recordsPerRequestAvg	Avg. Messages of Single Request	Average number of messages of a single request	-	INT	AVG
	seqIds	Producer-generated SN	Sequence number generated by the producer	-	STRING	LAST
	recordConsumedTotal	Total Consumption Times	Total number of consumption times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	bytesConsumedTotal	Total Consumed Bytes	Total number of bytes that have been consumed	-	INT	SUM
Fetch monitoring (**fetch**: Kafka fetch monitoring data)	id	id	Client ID and IP address	-	ENUM	LAST
	bytesConsumedRate	Bytes Consumed/s	Number of bytes consumed per second	Byte	INT	AVG
	fetchLatencyAvg	Avg. Request Latency	Average request latency	ms	INT	AVG
	fetchLatencyMax	Max. Request Latency	Maximum request latency	ms	INT	MAX
	fetchRate	Requests/s	Number of requests per second	-	INT	AVG
	fetchSizeAvg	Avg. Bytes Fetched	Average number of bytes fetched for a request	Byte	INT	AVG
	fetchSizeMax	Max. Bytes Fetched	Maximum number of bytes fetched for a request	Byte	INT	MAX
	recordsConsumedRate	Messages Consumed/s	Number of messages consumed per second	-	INT	AVG
	recordsLagMax	Max. Accumulated Messages	Maximum number of accumulated messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	recordsPerRequestAvg	Avg. Messages of Single Request	Average number of messages of a single request	-	INT	AVG
	seqIds	Producer-generated SN	Sequence number generated by the producer	-	STRING	LAST
	recordConsumedTotal	Total Consumption Times	Total number of consumption times	-	INT	SUM
	bytesConsumedTotal	Total Consumed Bytes	Total number of bytes that have been consumed	-	INT	SUM
Partition (**partition**: Kafka partition data)	id	id	Client ID and IP address	-	ENUM	LAST
	partition	partition	Kafka partition name	-	ENUM	LAST
	recordsLag	Accumulated Messages	Number of accumulated messages	-	INT	LAST
	recordsLagAvg	Avg. Accumulated Messages	Average number of accumulated messages	-	INT	AVG
	recordsLagMax	Max. Accumulated Messages	Maximum number of accumulated messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	seqIds	Producer-generated SN	Sequence number generated by the producer	-	STRING	LAST
Kafka consumption method monitoring (**consumer**)	method	Method	Consumption method	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error details	-	STRING	LAST
	maxTime	Max. RT	Maximum response time in a collection period	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with more than 10s response time	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
KafkaConsumer summary (**total**)	recordConsumedTotal	Total Consumption Times	Total number of consumption times	-	INT	SUM
	bytesConsumedTotal	Total Consumed Bytes	Total number of bytes that have been consumed	-	INT	SUM
	recordsLag	Total Accumulated Messages	Total number of messages that have been accumulated	-	INT	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics about Kafka consumption)	causeType	Exception Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Exception Message	Exception message	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack	-	CLOB	LAST

1.7.10.2 KafkaProducer Monitoring

This section describes the types, names, and meanings of KafkaProducer metrics collected by APM.

Table 1-62 KafkaProducer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Topic (**topic**: Kafka topic monitoring data)	id	id	Client ID and IP address	-	ENUM	LAST
	topic	topic	Kafka topic name	-	ENUM	LAST
	byteRate	Bytes Sent/s	Number of bytes sent per second	Byte	INT	AVG
	recordErrorRate	Errors/s	Number of errors per second	-	INT	AVG
	recordRetryRate	Retries/s	Number of retries per second	-	INT	AVG

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	recordSendRate	Messages sent/s	Number of messages sent per second	-	INT	AVG
	seqIds	Producer-generated SN	Sequence number generated by the producer	-	STRING	LAST
	recordSendTotal	Total Send Times	Total number of send times	-	INT	SUM
	byteTotal	Total Sent Bytes	Total number of bytes that have been sent	-	INT	SUM
KafkaProducer summary (**total**)	recordSendTotal	Total Send Times	Total number of send times	-	INT	SUM
	byteTotal	Total Sent Bytes	Total number of bytes that have been sent	-	INT	SUM
Exception (**exception**: exception statistics about Kafka byte sending)	causeType	Exception Class	Exception class	-	ENUM	LAST
	exceptionType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of exceptions	-	INT	SUM
	message	Exception Message	Exception message	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack	-	CLOB	LAST
Send methods (**doSendMethod**)	topic	topic	topic	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Errors	Number of errors	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with more than 10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time	-	INT	SUM

1.7.10.3 RabbitMqCommon Monitoring

This section describes the types, names, and meanings of RabbitMqCommon metrics collected by APM.

1.7.10.4 RabbitMqConsumer Monitoring

This section describes the types, names, and meanings of RabbitMqConsumer metrics collected by APM.

Table 1-63 Call metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of RabbitMqConsumer calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Push-mode consumption monitoring (**pushConsume**: APM counts statistics about push-mode message consumption.)	pushConsumeIdentifier	Identifier	Push-mode consumption identifier	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	consumedMsgCount	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Consumed	Maximum number of bytes consumed each time	-	INT	MAX
	manualAckCount	ACK Messages	Number of ACK messages	-	INT	SUM
	rejectCount	Rejected Messages	Number of rejected messages	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	requeueCount	Re-queued Messages	Number of re-queued messages	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	unackedMsgCount	Unacked Messages in Channel	Number of unacknowledged messages in the channel	-	INT	LAST
Connection monitoring (**connectionConsume**: APM counts message consumption statistics by connection.)	connection	connection	Consumer connection information	-	ENUM	LAST
	connectionCount	Current Connections	Current number of connections	-	INT	LAST
	channelCount	Current Channels	Current number of channels	-	INT	LAST
	connectionCreated	Created Connections	Number of connections that have been created	-	INT	SUM
	connectionClosed	Destroyed Connections	Number of connections that have been destroyed	-	INT	SUM
	channelCreated	Created Channels	Number of channels that have been created	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	channelClosed	Destroyed Channels	Number of channels that have been destroyed	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	consumedMsgCount	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Consumed	Maximum number of bytes consumed each time	-	INT	MAX
	manualAckCount	ACK Messages	Number of ACK messages	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	rejectCount	Rejected Messages	Number of rejected messages	-	INT	SUM
	requeueCount	Re-queued Messages	Number of re-queued messages	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	unackedMsgCount	Unacked Messages	Number of messages that have not been acknowledged in a connection	-	INT	LAST
Total monitoring (**total**: APM counts message consumption statistics by client.)	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	consumedMsgCount	consumedMsgCount	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Consumed	Maximum number of bytes consumed each time	-	INT	MAX
	manualAckCount	ACK messages	Number of ACK messages	-	INT	SUM
	rejectCount	Rejected Messages	Number of rejected messages	-	INT	SUM
	requeueCount	Re-queued Messages	Number of re-queued messages	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	unackedMsgCount	Unacked Messages	Number of unacknowledged messages on the client	-	INT	LAST

1.7.10.5 RabbitMqProducer Monitoring

This section describes the types, names, and meanings of RabbitMqProducer metrics collected by APM.

Table 1-64 RabbitMqProducer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of RabbitMqProducer calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Exchange monitoring (**exchange Publish**: APM counts message push statistics by exchange.)	connection	connection	Producer connection information	-	ENUM	LAST
	exchange	exchange	Exchange name	-	ENUM	LAST
	concurrentMax	Maximum concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Pushed	Maximum number of bytes in each push	-	INT	MAX
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Connection monitoring (**connectionPublish**: APM counts message push statistics by connection.)	connection	connection	Producer connection information	-	ENUM	LAST
	connectionCount	Current Connections	Current number of connections	-	INT	LAST
	channelCount	Current Channels	Current number of channels	-	INT	LAST
	connectionCreated	Created Connections	Number of connections that have been created	-	INT	SUM
	connectionClosed	Destroyed Connections	Number of connections that have been destroyed	-	INT	SUM
	channelCreated	Created Channels	Number of channels that have been created	-	INT	SUM
	channelClosed	Destroyed Channels	Number of channels that have been destroyed	-	INT	SUM
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Pushed	Maximum number of bytes in each push	-	INT	MAX
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Total monitoring (**total**: APM counts message push statistics by client.)	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	maxSingleMsgBytes	Max. Bytes Pushed	Maximum number of bytes in each push	-	INT	MAX
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

1.7.10.6 RocketMqConsumer Monitoring

This section describes the types, names, and meanings of RocketMqConsumer metrics collected by APM.

Table 1-65 RocketMqConsumer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of RocketMqConsumer calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Consumption pool monitoring (**consumeServicePool**)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	consumerGroup	Consumer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	currentConsumeRequestQueueSize	Current Size of Consumption Request Queue	Current size of the consumption request queue	-	INT	AVG
	maxConsumeRequestQueueSize	Max. Size of Consumption Request Queue	Maximum size of the consumption request queue	-	INT	MAX
	currentConsumingThreadCount	Current Consumption Threads	Current number of consumption threads	-	INT	AVG
	maxConsumingPoolSize	Max. Consumption Threads	Maximum number of consumption threads	-	INT	MAX
Message listener monitoring (**consumeListener**: APM counts message consumption statistics by MessageListener.)	consumeListener	MessageListener	Registered message listener, which is the callback function for message consumption	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	Messages Consumed	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	reconsumeTimes	Message Re-consumption Times	Number of message re-consumption times	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Client ID monitoring (**clientIdConsume**: APM counts message consumption statistics by client ID.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	Messages Consumed	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	reconsumeTimes	Message re-consumption times	Number of message re-consumption times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Topic monitoring (**topicConsume**: APM counts message consumption statistics by topic.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	topic	Topic	Topic for message consumption	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	consumedMsgCount	Messages Consumed	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	reconsumeTimes	Message Re-consumption Times	Number of message re-consumption times	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being consumed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for consuming messages	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Queue monitoring (**queueConsume**: APM counts message consumption statistics by queue.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	queue	Message Queue	Message queue ID	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent consumption	-	INT	MAX
	errorCount	Errors	Number of message consumption errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	Messages Consumed	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	reconsumeTimes	Message Re-consumption Times	Number of message re-consumption times	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message consumption	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for consuming messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of messages that are being pulled at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pulling messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Client ID monitoring (**clientIdPull**: APM counts message pull statistics by client ID.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent pulling	-	INT	MAX
	errorCount	Errors	Number of message pull errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of pull calls	-	INT	SUM
	pulledMsgCount	Messages Pulled	Number of messages that have been pulled	-	INT	SUM
	pulledBytes	Bytes Pulled	Number of bytes that have been pulled	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message pulling	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	Max. RT	Maximum response time for pulling messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pulled at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pulling messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Topic monitoring (**topicPull**: APM counts message pull statistics by topic.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	topic	Topic	Topic for pulling messages	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent pulling	-	INT	MAX
	errorCount	Errors	Number of message pull errors	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of pull calls	-	INT	SUM
	pulledMsgCount	Messages Pulled	Number of messages that have been pulled	-	INT	SUM
	pulledBytes	Bytes Pulled	Number of bytes that have been pulled	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message pulling	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pulling messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pulled at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	totalTime	Total RT	Total response time for pulling messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Queue monitoring (**queuePull**: APM counts message pull statistics by queue.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Consumer Group	Consumer group	-	ENUM	LAST
	queue	Message Queue	Message queue ID	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent pulling	-	INT	MAX
	errorCount	Errors	Number of message pull errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of pull calls	-	INT	SUM
	pulledMsgCount	Messages Pulled	Number of messages that have been pulled	-	INT	SUM
	pulledBytes	Bytes Pulled	Number of bytes that have been pulled	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message pulling	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pulling messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of messages that are being pulled at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pulling messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Total monitoring (**total**: APM counts message consumption statistics by client.)	consumeErrorCount	Consumption Errors	Number of message consumption errors	-	INT	SUM
	consumeInvokeCount	consumeInvokeCount	Number of consumption call times	-	INT	SUM
	consumedMsgCount	consumedMsgCount	Number of messages that have been consumed	-	INT	SUM
	consumedBytes	Bytes Consumed	Number of bytes that have been consumed	-	INT	SUM
	consumeTotalTime	Total RT for Message Consumption	Total response time for consuming messages	-	INT	SUM
	reconsumeTimes	Message Re-consumption Times	Number of message re-consumption times	-	INT	SUM
	pullErrorCount	Pull Errors	Number of message pull errors	-	INT	SUM
	pullInvokeCount	pullInvokeCount	Number of pull calls	-	INT	SUM
	pulledMsgCount	pulledMsgCount	Number of messages that have been pulled	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	pulledBytes	Bytes Pulled	Number of bytes that have been pulled	-	INT	SUM
	pullTotalTime	Total Pull RT	Total response time for pulling messages	-	INT	SUM

1.7.10.7 RocketMqProducer Monitoring

This section describes the types, names, and meanings of RocketMqProducer metrics collected by APM.

Table 1-66 RocketMqProducer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: exception statistics of RabbitMq Producer calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Client ID monitoring (**clientIdPublish**: APM counts message push statistics by client ID.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Producer Group	Producer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Topic monitoring (**topicPublish**: APM counts message push statistics by topic.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Producer Group	Producer group	-	ENUM	LAST
	topic	Topic	Topic to which a message is pushed	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Queue monitoring (**queuePublish**: APM counts message push statistics by queue.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Producer Group	Producer group	-	ENUM	LAST
	queue	Message Queue	Message queue ID	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Broker monitoring (**brokerPublish**: APM counts message push statistics by broker.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Producer Group	Producer group	-	ENUM	LAST
	broker	broker	Broker address	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Information about the error that has occurred during message pushing	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Transaction monitoring (**transactionPublish**: APM counts transaction message push statistics by client.)	clientId	clientId	Client instance ID	-	ENUM	LAST
	group	Producer Group	Producer group	-	ENUM	LAST
	pid	pid	PID	-	STRING	LAST
	concurrentMax	Max. Concurrency	Maximum number of transaction messages for concurrent push	-	INT	MAX
	errorCount	Errors	Number of transaction message push errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	lastError	Error Message	Error information generated when an error has occurred during transaction message push	-	STRING	LAST
	maxTime	Max. RT	Maximum response time for pushing transaction messages	-	INT	MAX
	runningCount	Ongoing Executions	Number of transaction messages that are being pushed at the time of collection	-	INT	SUM
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	totalTime	Total RT	Total response time for pushing transaction messages	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–200 ms	Number of requests with 100–200 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	200–1000 ms	Number of requests with 200–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Total monitoring (**total**: APM counts message push statistics by client.)	errorCount	Errors	Number of message push errors	-	INT	SUM
	invokeCount	invokeCount	Number of message push calls	-	INT	SUM
	publishedMsgCount	publishedMsgCount	Number of push messages	-	INT	SUM
	publishedBytes	Push Bytes	Number of push bytes	-	INT	SUM
	totalTime	Total RT	Total response time for pushing messages	-	INT	SUM

1.7.11 RPC

1.7.11.1 GRPCClient Monitoring

This section describes the types, names, and meanings of GRPCClient metrics collected by APM.

Table 1-67 GRPCClient monitoring metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Method monitoring (**detail**: APM counts URL call statistics by method.)	method	Method	Request method	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the method in a collection period	-	INT	MAX
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	totalTime	Total RT	Total response time of the method	-	INT	SUM
Cluster call (**cluster**: APM counts URL call statistics based on the cluster ID of the caller.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	totalTime	Total RT	Total response time for calling the cluster	-	INT	SUM
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM

1.7.11.2 GRPCServer Monitoring

This section describes the types, names, and meanings of GRPCServer metrics collected by APM.

Table 1-68 GRPCServer monitoring collection parameters

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Sampling Type	radio	JAVA	4	1.0.0	-	Sampling type. Options: full sampling, percentage sampling, fixed-quantity sampling per minute, and intelligent sampling (default).
Sampling Ratio	integer	JAVA	10	1.0.0	-	Percentage of samples to the total number of trace data records
Samples/ Minute	integer	JAVA	1000	1.0.0	-	Number of trace data records collected every minute.
Slow Request Threshold	integer	JAVA	800	2.0.0	-	Slow request threshold. If the threshold is crossed, the method is regarded as a slow method. In that case, the trace sampling ratio will be increased by default.

Parameter	Data Type	Application Type	Default	Supported Start Agent Version	Supported End Agent Version	Description
Method Configuration	obj_array	JAVA	-	2.0.0	-	Configure the slow request threshold and sampling ratio for each method separately. The following sampling policies can be set: percentage sampling, fixed-quantity sampling per minute, and automatic sampling.

Table 1-69 GRPCServer monitoring metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Method monitoring (**detail**: APM counts URL call statistics by method.)	method	Method	Request method	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the method in a collection period	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	totalTime	Total RT	Total response time of the method	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Cluster call (**cluster**: APM counts URL call statistics based on the cluster ID of the caller.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	totalTime	Total RT	Total response time for calling the cluster	-	INT	SUM
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM

1.7.12 IoT

1.7.12.1 CoapClient Monitoring

This section describes the types, names, and meanings of CoapClient metrics collected by APM.

Table 1-70 Call metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
URL monitoring (**detail**: APM counts URL call statistics by URL, packet type, and request type.)	url	URL	Request URL	-	ENUM	LAST
	requestType	Packet Type	Packet type	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	maxTime	Max. RT	Maximum response time of the method in a collection period	-	INT	MAX
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	totalTime	Total RT	Total response time of the method	-	INT	SUM
	method	Request Type	Request type	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	lastError	Error Message	Error message	-	STRING	LAST
Status code (**statusInfo**: APM counts URL call statistics based on the status code returned.)	statusInfo	Status Code	Status code	-	ENUM	LAST
	count	Calls	Number of times that the status code has occurred	-	INT	SUM
	url	Sample URL	Sample URL which returns the status code in a collection period	-	STRING	LAST
Cluster call (CON packets) (**cluster_con**: APM counts URL call statistics (CON packets) based on the cluster ID.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	totalTime	Total RT	Total response time for calling the cluster	-	INT	SUM
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
CoapClient version (**version**)	version	Version	Version	-	STRING	LAST

1.7.12.2 CoapServer Monitoring

This section describes the types, names, and meanings of CoapServer metrics collected by APM.

Table 1-71 CoapServer metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
URL monitoring (**detail**: APM counts URL call statistics by URL, packet type, and request type.)	url	URL	Request URL	-	ENUM	LAST
	requestType	Packet Type	Packet type	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency of the method	-	INT	MAX
	errorCount	Errors	Number of times that the method fails to be called	-	INT	SUM
	invokeCount	Calls	Number of times that the method is called	-	INT	SUM
	maxTime	Max. RT	Maximum response time of the method in a collection period	-	INT	MAX
	runningCount	Ongoing Executions	Number of executions of the method at the time of collection	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
	totalTime	Total RT	Total response time of the method	-	INT	SUM
	method	Request Type	Request type	-	ENUM	LAST
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	lastError	Error Message	Error message	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Status code (**statusInfo**: APM counts URL call statistics based on the status code returned.)	statusInfo	Status Code	Status code	-	ENUM	LAST
	count	Calls	Number of times that the status code has occurred	-	INT	SUM
	url	url	URL corresponding to the status code	-	STRING	LAST
Cluster call (**cluster**: APM counts URL call statistics based on the cluster ID of the caller.)	clusterId	Cluster ID	Cluster ID of the caller	-	ENUM	LAST
	invokeCount	Calls	Number of times the cluster is called	-	INT	SUM
	totalTime	Total RT	Total response time for calling the cluster	-	INT	SUM
	errorCount	Errors	Number of times that the cluster fails to be called	-	INT	SUM
	clientErrorCount	Client Errors	Number of client errors	-	INT	SUM
	serverErrorCount	Server Errors	Number of server errors	-	INT	SUM
CoapServer version (**version**)	version	Version	Version	-	STRING	LAST

1.7.12.3 MoquetteBroker Monitoring

This section describes the types, names, and meanings of MoquetteBroker metrics collected by APM.

Table 1-72 MoquetteBroker metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: Moquette Broker call exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
Moquette Broker version (**version**)	version	Version	Version	-	STRING	LAST
Moquette Broker topic summary (**total**: Moquette Broker topic summary)	msgSentCount	Message Sending Times	Total number of message sending times	-	INT	SUM
	bytesSent	Bytes Sent	Total number of bytes sent	-	INT	SUM
	msgReceivedCount	Message Receiving Times	Total number of message receiving times	-	INT	SUM
	bytesReceived	Bytes Received	Total number of bytes received	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Moquette Broker topic-based monitoring (**brokerTopic**)	topic	Topic	Topic	-	ENUM	LAST
	subscribeCount	Subscriptions	Number of subscriptions	-	INT	SUM
	msgSentCount	Message Sending Times	Number of message sending times	-	INT	SUM
	bytesSent	Bytes Sent	Number of bytes sent	-	INT	SUM
	msgReceivedCount	Message Receiving Times	Number of message receiving times	-	INT	SUM
	bytesReceived	Bytes Received	Number of bytes received	-	INT	SUM

1.7.12.4 PahoPublisher Monitoring

This section describes the types, names, and meanings of PahoPublisher metrics collected by APM.

Table 1-73 PahoPublisher metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Monitoring of PUBLISH packets sent by PahoPublisher (**message**)	uri	service Uri	URI of the MQTT server connected to PahoPublisher	-	ENUM	LAST
	msgType	Packet Type	Type of the packet that is sent	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Node-based monitoring of PUBLISH packets sent by PahoPublisher (**uriMessage**)	uri	service Uri	URI of the MQTT server connected to PahoPublisher	-	ENUM	LAST
	errorCount	Errors	Number of errors	-	INT	SUM
	invokeCount	Calls	Number of calls	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
Exception (**exception**: exception statistics of PahoPublisher calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
PahoPublisher monitoring by topic (**clientPublish**)	clientId	clientId	clientId	-	ENUM	LAST
	topic	Topic	Topic	-	ENUM	LAST
	msgSentCount	Message Sending Times	Number of message sending times	-	INT	SUM
	bytesSent	Bytes Sent	Number of bytes sent	-	INT	SUM
PahoPublisher version (**version**)	version	Version	Version	-	STRING	LAST
PahoPublisher topic summary (**total**)	msgSentCount	Message Sending Times	Total number of message sending times	-	INT	SUM
	bytesSent	Bytes Sent	Total number of bytes sent	-	INT	SUM

1.7.12.5 PahoSubscriber Monitoring

This section describes the types, names, and meanings of PahoSubscriber metrics collected by APM.

Table 1-74 PahoSubscriber metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Monitoring of PUBLISH packets received by PahoSubscriber (**message**)	uri	serviceUri	URI of the MQTT server connected to PahoSubscriber	-	ENUM	LAST
	msgType	Packet Type	Type of the packet that is sent	-	ENUM	LAST
	concurrentMax	Max. Concurrency	Maximum concurrency	-	INT	MAX
	errorCount	Errors	Number of errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of calls	-	INT	SUM
	lastError	Error Message	Error message	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
Node-based monitoring of PUBLISH packets received by PahoSubscriber (**uriMessage**)	uri	serviceUri	URI of the MQTT server connected to PahoSubscriber	-	ENUM	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	errorCount	Errors	Errors	-	INT	SUM
	invokeCount	Number of calls	Calls	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM
Exception (**exception**: exception statistics of PahoSubscriber calls)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST
PahoSubscriber topic-based monitoring (**clientReceive**)	clientId	clientId	clientId	-	ENUM	LAST
	topic	Topic	Topic	-	ENUM	LAST
	msgReceivedCount	Message Receiving Times	Number of message receiving times	-	INT	SUM
	bytesReceived	Bytes Received	Number of bytes received	-	INT	SUM
PahoSubscriber version (**version**)	version	Version	Version	-	STRING	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
PahoSubscriber topic summary (**total**)	msgReceivedCount	Message Receiving Times	Total number of message receiving times	-	INT	SUM
	bytesReceived	Bytes Received	Total number of bytes received	-	INT	SUM

1.7.13 Communication Protocol

This section describes the types, names, and meanings of WebSocket metrics collected by APM.

Table 1-75 WebSocket metrics

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
Exception (**exception**: WebSocket exception statistics)	exceptionType	Exception Type	Exception type	-	ENUM	LAST
	causeType	Exception Class	Exception class	-	ENUM	LAST
	count	Count	Number of times the exception has occurred	-	INT	SUM
	message	Exception Message	Message returned when the exception occurred	-	STRING	LAST
	stackTrace	Exception Stack	Exception stack information	-	CLOB	LAST

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
WebSocket message monitoring (**message**: WebSocket message processing information)	url	url	URL corresponding to WebSocket	-	ENUM	LAST
	errorCount	Errors	Number of message processing errors	-	INT	SUM
	errorTraceId	Error Trace ID	ID of the trace that encounters an error in a collection period	-	STRING	LAST
	slowTraceId	Slow Trace ID	ID of the slowest trace in a collection period	-	STRING	LAST
	invokeCount	Calls	Number of times that the message processing method is called	-	INT	SUM
	traffic	Traffic	Traffic	-	INT	SUM
	createSessionCount	Created Connections	Number of connections that have been created	-	INT	SUM
	closeSessionCount	Closed Connections	Number of closed connections	-	INT	SUM
	closeReason	Close Reason	Cause of the connection closure	-	STRING	LAST
	maxTime	Max. RT	Maximum response time	-	INT	MAX
	totalTime	Total RT	Total response time	-	INT	SUM

Category	Metric	Name	Description	Unit	Data Type	Default Aggregation Mode
	range1	0–10 ms	Number of requests with 0–10 ms response time	-	INT	SUM
	range2	10–100 ms	Number of requests with 10–100 ms response time	-	INT	SUM
	range3	100–500 ms	Number of requests with 100–500 ms response time	-	INT	SUM
	range4	500–1000 ms	Number of requests with 500–1000 ms response time	-	INT	SUM
	range5	1–10s	Number of requests with 1–10s response time	-	INT	SUM
	range6	> 10s	Number of requests with response time longer than 10s	-	INT	SUM
WebSocket summary (**total**: summary statistics)	errorCount	Errors	Total number of errors	-	INT	SUM
	invokeCount	Calls	Total number of calls	-	INT	SUM
	createSessionCount	Created Connections	Number of connections that have been created	-	INT	SUM
	closeSessionCount	Closed Connections	Number of closed connections	-	INT	SUM
	traffic	Traffic	Traffic	-	INT	SUM
	totalTime	Total RT	Total response time	-	INT	SUM

1.8 Privacy and Sensitive Information Protection Statement

All O&M data will be displayed on the APM console. Therefore, you are not advised to upload your privacy or sensitive data to APM. If you need to upload such data, encrypt them.

1.9 Data Collection

After you enable data collection, APM collects application performance metrics and tracing data. Your personal privacy data will not be collected. The collected data will be used only for application performance analysis and fault diagnosis, and will not be used for commercial purposes.

Data Type	Collected Data	Transmission Mode	Storage Mode	Function	Storage Period
Performance metric data	JVM data, exceptions, databases, SQL statements, and middleware call data	WebSocket Secure (WSS)	Tenant-based isolated storage on the server	Metric query and display at the frontend	7 days for the basic edition and 30 days for the enterprise edition. The data will be permanently deleted upon expiration.
Tracing data	Trace event data, including middleware invocation data	WSS	Tenant-based isolated storage on the server	Query and display at the tracing frontend	7 days for the basic edition and 30 days for the enterprise edition. The data will be permanently deleted upon expiration.

Resource information	Service type, service name, creation time, deletion time, node address, and service release API	WSS	Tenant-based isolated storage on the server	Query and display at the resource library frontend	7 days for the basic edition and 30 days for the enterprise edition. The data will be permanently deleted upon expiration.
Resource attributes	System type, system startup event, number of CPUs, service executor, service process ID, service pod ID, CPU label, system version, web framework, JVM version, time zone, system name, collector version, and LastMail URL	WSS	Tenant-based isolated storage on the server	Query and display at the resource library frontend	7 days for the basic edition and 30 days for the enterprise edition. The data will be permanently deleted upon expiration.

Table 1-76 Restrictions on collection items

Collection Item	Maximum Value
Monitoring item rows	500
SQL length	2000 characters
SQL result bodies	100
SQL result body content	999 characters
Redis body length	100 characters
Mongo clusters	10
Mongo command length	2000 characters
HBase command length	500 characters
ES RestClients	10

Collection Item	Maximum Value
Cassandra CQL length	2000 characters
Cassandra sessions	10
Kafka MBean object names	100
Cache IP addresses corresponding to Kafka client IDs	100
RabbitMQ connection addresses	20
Cache connections for each RabbitMQ address	100
RabbitMQ consumers	500
Cache channels for each RabbitMQ consumer	100
RabbitMQ messages without ACK in each channel	3000
Manual ACK consumers in RabbitMQ cache	20
RocketMQ PIDs	20
RocketMQ client IDs	20
Jetcd tag length	500 characters
HttpClient connections	10
Report time of connection pool trace	1 ms
Dubbo invocation length	500 characters
Dubbo attachment length	500 characters
URL body length	9999 characters
Application code body length	0 characters
Java method body length	8192 characters

1.10 Usage Restrictions

1.10.1 Java

Supported Java Components and Frameworks

Currently, APM can connect to Java applications. APM supports multiple mainstream Java frameworks, web servers, communications protocols, and databases.

Table 1-77 Java components and frameworks supported by self-developed Agents

Agent Type	Component	JDK 1.8	JDK 17
Self-developed	Dubbo	2.6.x	2.6.x
Self-developed	Jedis	2.x.x-3.x.x	2.x.x-3.x.x
Self-developed	Lettuce	5.x.x	5.x.x
Self-developed	ServiceComb	2.x.x	-
Self-developed	Log4j	1.x.x	1.x.x
Self-developed	Log4j2	2.x.x	2.x.x
Self-developed	HttpClient	4.x.x	4.x.x–5.3.x (5.x.x applies only to synchronization scenarios)
Self-developed	JDK HttpClient	1.6–1.8	17
Self-developed	MariaDB	2.x.x	2.x.x
Self-developed	MySQL	5.x.x–8.x.x	5.x.x–8.x.x
Self-developed	OkHttpClient	3.x.x	3.x.x
Self-developed	Tomcat	6.x.x–9.x.x	9.x.x
Self-developed	Jetty	8.x.x–9.x.x	9.x.x
Self-developed	gRPC	1.x.x	1.x.x
Self-developed	Reactor Netty	1.x.x	-
Self-developed	HBase	2.x.x	2.x.x
Self-developed	MongoDB	3.x.x–4.x.x	3.x.x–4.x.x
Self-developed	c3p0	0.9.x	0.9.x
Self-developed	Cassandra3	3.x.x	3.x.x
Self-developed	ClickHouse	0.2.x	0.2.x
Self-developed	DBCP	2.x.x	2.x.x
Self-developed	Druid	1.x.x	1.x.x

Agent Type	Component	JDK 1.8	JDK 17
Self-developed	HttpAsyncClient	4.x.x	4.x.x
Self-developed	Jetty Client	9.x.x	9.x.x
Self-developed	MariaDB3	3.x.x	3.x.x
Self-developed	MyBatis	3.x.x	3.x.x
Self-developed	Netty	4.x.x	4.x.x
Self-developed	PostgreSQL	42.x.x	42.x.x
Self-developed	RabbitMQ	5.x.x	5.x.x
Self-developed	Undertow	2.x.x	2.x.x
Self-developed	WebSocket	9.x.x	9.x.x
Self-developed	Elasticsearch	7.x.x	7.x.x
Self-developed	Oracle	10.x.x	10.x.x
Self-developed	RocketMQ	4.x.x	4.x.x
Self-developed	Kafka	2.x.x	2.x.x

2 Getting Started

2.1 Enabling APM 2.0

1. Enable APM 2.0.
   a. Log in to the APM console.

      NOTE

      If you log in to the APM console for the first time, you can choose the free or enterprise edition. For details, see **Edition Differences**.

   b. Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.
2. Obtain the access key that is automatically created.

   APM 2.0 uses AK/SK for signature verification. Only authorized accounts can report data.

   a. Log in to the APM 2.0 console.
   b. In the navigation pane, choose **System Management** > **Access Keys**.
   c. On the **Access Keys** page, view the access key that has been automatically created.

2.2 Monitoring Java Applications

2.2.1 Connecting Agents

Prerequisites

The network between your host and APM is normal.

You can run the **Telnet** command to check the network.

NOTICE

Java supports enhanced Agents.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Applications**.

Step 4 On the displayed page, click **Connect Application**.

Step 5 Select a region and application.

Step 6 Select **Java** for **Backend Language**.

Step 7 Select **Enhanced Agent** for **Code Source**.

Step 8 Select an access mode based on the application type and access data by following the instructions.

Table 2-1 Parameter description

Parameter	Description	Mandatory
pwd	Path where the **apm-javaagent.jar** package is located.	Yes
appName	Component name, which must start with a letter. A component can contain multiple environments. The names of components under an application must be unique. If there are duplicate names, use **instanceName** to distinguish them.	Yes
env	Name of an environment where an application is deployed. A program can be deployed in different environments (such as the test or live network environment). Each environment is deployed in one region and has a unique region attribute. If this parameter is blank, the default environment will be used.	No
envTag	Environment tag for filtering environments. Different environments may have the same tag. This parameter can be left blank.	No
business	Name of an application that already exists (a global concept). If this parameter is left blank, the automatically created application will be used.	No

Parameter	Description	Mandatory
subBusiness	Name of a sub-application (a global concept), which is similar to a folder. If it is left blank, resources will be mounted to the root application. There can be up to three layers of sub-applications. For example, for **a/b/c**, **a**, **b**, and **c** respectively represents a layer.	No
instanceName	Name of an instance, which is left blank by default. If an application has multiple instances deployed on a host, use this parameter to distinguish them. Generally, Java instances deployed on a host belong to different applications. An application rarely has identical instances.	No

----End

2.2.2 Manually Installing Agents for Java Applications

Prerequisites

- The network between your host and APM is normal.

  You can run the **Telnet** command to check the network.

  If the network is not connected, use a proxy instead.
- The AK/SK required for accessing JavaAgents have been obtained. To obtain them, log in to the APM console and choose **System Management** > **Access Keys** in the navigation pane.

Procedure

Step 1 Download **apm-javaagent** to any directory of your host. For the download address, see **JavaAgent Download Addresses**.

Example command:

curl -O https://xxx/apm-javaagent-x.x.x.tar

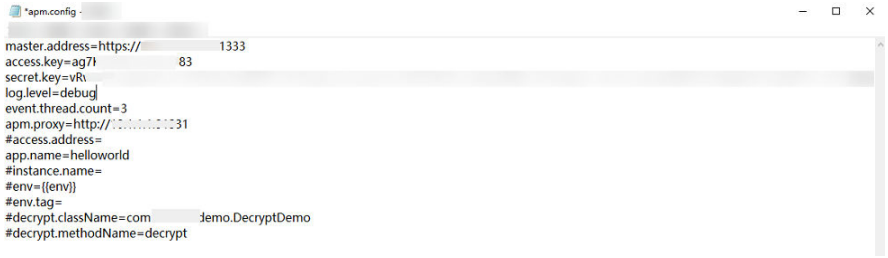
Step 2 Run the **tar** command to decompress the JavaAgent package.

Example command:

tar -xvf apm-javaagent-x.x.x.tar

Step 3 Modify the **apm.config** file in the JavaAgent package. Configure **master.address** by referring to **Access Addresses**, and add the AK/SK to the configuration file, as shown in the following figure.

Figure 2-1 Adding the AK/SK



Step 4 Modify the startup script of the Java process.

Add the path of the **apm-javaagent.jar** package and the component name of the Java process to the end of the Java command in the service startup script.

Example of adding **-javaagent** parameters:

java **-javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName={appName}**

If your enterprise has a large number of services, you can add more complex configurations. For example:

java **-javaagent:/xxx/apm-javaagent/apm-javaagent.jar=appName=myApp,env=myEnv,envTag=myTag,business=myBusiness,subBusiness=mySub**

NOTE

- The preceding parameters are built-in CMDB information of APM. For details, see **CMDB Management**.
- Due to historical reasons, the metadata of APM startup parameters conflicts with some CMDB concepts. The following shows the details.

  Generally, the startup parameter is set to **-javaagent:D:\javaagent-package\apm-javaagent\apm-javaagent.jar=appName=xxx,env=yyy,business=zzz,subBusiness=sss,envTag=xxx**. **appName** indicates a component, **business** indicates an application, **subBusiness** indicates a sub-application, and **envTag** indicates an environment tag.

  If **business** is not set on the web page, the system reports an error when the JavaAgent is started. If other parameters (**subBusiness**, **appName**, **env**, and **envTag**) are not set, the system automatically creates them when the JavaAgent is started.

  Component names are unique under an application.

Step 5 Redeploy the application.

----End

2.2.3 Installing Agents for the Java Applications Deployed in CCE Containers

NOTE

- You are advised to install self-developed Agents for the Java applications deployed in CCE containers.
- Java supports enhanced Agents.

Prerequisites

- The network between your host and APM is normal.

  You can run the **Telnet** command to check the network.

- For details, see **Regions and Endpoints**.
- The AK/SK required for accessing JavaAgents have been obtained. To obtain them, log in to the APM console and choose **System Management** > **Access Keys** in the navigation pane.

Usage Instruction

APM only supports Java applications deployed on CCE. **Table 2-2** describes the parameters.

Table 2-2 Parameters for configuring performance management

Name	Description
Probe	Select a target probe. Options: **Disable/APM 2.0**.
Probe Version	Version of the probe.
Probe Upgrade Policy	Policy for the probe upgrade. The default value is **Auto upgrade upon restart**. • **Automatic upgrade upon restart**: The system downloads the probe image each time the pod is restarted. • **Manual upgrade**: If a local image is available, it will be used. If no local image is available, the system downloads the probe image.
APM Environment	Enter an APM environment name. This parameter is optional.
APM App	Select an existing APM application.
Sub-app	Enter an APM sub-application. This parameter is optional.
Access Key	The system automatically obtains the APM key. For details, see **Prerequisites**.

Procedure

Step 1 Log in to the CCE console. In the navigation pane, choose **Workloads** > **Deployments** or **StatefulSets**, and click **Create Deployment** or **Create StatefulSet**.

Step 2 In the **APM Settings** area on the **Configure Advanced Settings** page, select **Java probe**. The APM service will be enabled and a probe will be installed on the node.

NOTE

Probes provide traces, topologies, SQL analysis, and stack tracing for Java workloads. A small number of resources will be consumed when you run probes.

Step 3 Set probe-related parameters.

- **Monitoring Group**: Enter a monitoring group name, for example, **testapp**. Select a group from the drop-down list if there are any.
- **Probe Version**: Select a probe version.
- **Probe Upgrade Policy**: By default, **Automatic upgrade upon restart** is selected.
  - **Automatic upgrade upon restart**: The system downloads the probe image each time the pod is restarted.
  - **Manual upgrade**: If a local image is available, it will be used. If no local image is available, the system downloads the probe image.

Step 4 After the application is started, wait for about 3 minutes. Then, the application data is displayed on the APM console. You can log in to the APM console and optimize the application performance through topology and tracing.

----End

2.2.4 Installing Agents on Applications Deployed Using CodeArts Deploy

Prerequisite

The network between your host and APM is normal.

You can run the **Telnet** command to check the network.

NOTICE

Java supports enhanced Agents.

Procedure

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Applications**.

Step 4 On the displayed page, click **Connect Application**.

Step 5 Select a region and application.

Step 6 Select **Java** for **Backend Language**.

Step 7 Select **Enhanced Agent** for **Code Source**.

Step 8 Select an access mode based on the application type and access data by following the instructions.

Table 2-3 Parameter description

Parameter	Description	Mandatory
pwd	Path where the **apm-javaagent.jar** package is located.	Yes
appName	Component name, which must start with a letter. A component can contain multiple environments. The names of components under an application must be unique. If there are duplicate names, use **instanceName** to distinguish them.	Yes
env	Name of an environment where an application is deployed. A program can be deployed in different environments (such as the test or live network environment). Each environment is deployed in one region and has a unique region attribute. If this parameter is blank, the default environment will be used.	No
envTag	Environment tag for filtering environments. Different environments may have the same tag. This parameter can be left blank.	No
business	Name of an application that already exists (a global concept). If this parameter is left blank, the automatically created application will be used.	No
subBusiness	Name of a sub-application (a global concept), which is similar to a folder. If it is left blank, resources will be mounted to the root application. There can be up to three layers of sub-applications. For example, for **a/b/c**, **a**, **b**, and **c** respectively represents a layer.	No
instanceName	Name of an instance, which is left blank by default. If an application has multiple instances deployed on a host, use this parameter to distinguish them. Generally, Java instances deployed on a host belong to different applications. An application rarely has identical instances.	No

Step 9 Access the CodeArts Deploy deployment task, edit the deployment action, add the step of running the **shell** command, and then add the **copied command**.

Step 10 Modify the deployment procedure, copy the startup parameter, and add the parameter to the Java command of the service startup script.

Step 11 Redeploy the application.

----End

2.3 JavaAgent Download Addresses

Region	Latest Version	Earlier Versions
AP-Kuala Lumpur	2.2.5 sha256:fc547d5c9c62a80f079c5e67b2f999f0df2193f6f1a2e959bf5506b824ab7853	-

2.4 Access Addresses

Table 2-4 Access addresses of Enhanced Agents

Region	Access Address
AP-Kuala Lumpur	https://100.125.14.18:41333

3 User Guide

3.1 Before You Start

This document describes how to use Application Performance Management (APM).

Application List	The **Applications** page displays information such as components, environments, Agent status, and supported operations.
CMDB Management	APM has built-in CMDB for managing the application structure and related configurations.
Application Metric Monitoring	APM can manage tags and monitor the metric data of JVM, GC, service calls, exceptions, external calls, database access, and middleware, helping you comprehensively monitor application running. Application metrics can be reported to the AOM console through Prometheus instances.
Tracing	Information such as the call status, duration, and API is displayed, helping you further locate fault causes.
Application Topology	The call and dependency relationships between applications are displayed, and abnormal instances can be automatically discovered. There are two types of application topologies: • Single-component topology: topology of a single component under an environment. You can also view the call relationships of direct and indirect upstream and downstream components. • Global application topology: topology of some or all components under an application.

URL Tracing	Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and then detect problems more precisely.
Resource Tag Management	You can tag resources under your account for classification.
Alarm Management	When an application connected to APM meets a preset alarm condition, an alarm is triggered and reported in a timely manner. In this way, you can quickly learn about service exceptions and rectify faults to prevent loss.
Agent Management	Agent Management allows you to check the deployment and running statuses of the Agents that are connected to APM, and to stop, start, or delete them.
Configuration Management	Configuration Management manages and displays the configurations supported by APM in a centralized manner. It consists of two parts: • Collection Center: displays collectors in a centralized manner. You can view and manage various collectors, metrics, and collection parameters supported by APM. • Data Masking: You can set policies to mask the data reported using APM 2.0 APIs.
System Management	System Management manages and displays system configurations in a centralized manner, including: • Access Keys: long-term identity credentials. They ensure that the requests are secret, complete, and correct. • General Configuration: You can determine whether to collect data through bytecode instrumentation, specify the slow request threshold and maximum number of rows to collect, and set web monitoring aggregation. • Agent Count: APM counts the number of Agents used by tenants.
Permissions Management	Enterprise Project Management Service (EPS) is used to control user access to APM resources.
Learn more	**Permissions Management** Create a user and grant permissions. *Getting Started* Learn how to connect applications to APM in different scenarios.

3.2 Application List

Application List

The **Applications** page displays information such as components, environments, Agent status, and supported operations.

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane on the left, choose **Application Monitoring** > **Applications**.

Component|Environment: name of a component or environment. You can click the text in blue to go to the corresponding to component or environment page.

Agent Status: number of Agents in different statuses.

The following table describes the Agent statuses.

Status	Description
Enabled	The Agent is running properly.
Offline	• The Agent is offline due to network problems. Check and restore the network. • The Agent is offline if your process does not exist. • The Agent is disabled and offline if the trial period expires.
Disabled	The Agent is manually or globally disabled. Contact technical support.

----End

More Operations

Perform the operations listed in **Table 3-1** if needed.

Table 3-1 Related operations

Operation	Description
Selecting an application	Select an application from the **Application** drop-down list on the left of the page.
Viewing the topology of an environment	Click **Topology** in the **Operation** column of an environment.

Operation	Description
Setting a component or environment	Click **Configure** in the **Operation** column. On the displayed **Instance** tab page, set the component or environment as required.
Deleting an environment	Click **Delete** in the **Operation** column of an environment.
Searching for a component or environment	Enter a component or environment keyword or name on the right.

3.3 CMDB Management

3.3.1 Introduction

APM has a built-in CMDB for managing application structure information and related configurations. It involves the following concepts:

- **Application** (global concept): a logical unit. An application can be an independent functional module. The same application information can be viewed in all regions.
- **Sub-application** (global concept): similar to a folder. There can be up to three layers of sub-applications under an application.
- **Component** (global concept): a program or microservice. It is generally used together with environments. It may contain one or more environments. For example, an order app can be deployed in the function test environment, pressure test environment, pre-release environment, or live network environment.
- **Environment**: Components or programs with different configurations are deployed in different environments. Each environment has its own region attribute. You can filter environments by region. You can also add one or more tags to an environment and filter environments by tag.
- **Instance**: a process in an environment. It is named in the format of "host name+IP address+instance name". An environment is usually deployed on different hosts or containers. If an environment is deployed on one host, differentiation by instance is supported.
- **Environment tag**: an attribute for filtering environments. Different environments may have the same tag. Tags carry public configuration capabilities. For example, the configuration set on a tag can be shared by the environments with the same tag. Tags defined for environments of one application cannot be applied to other applications.

The CMDB structure tree can be hidden.

Step 1 Click **Hide** to hide the CMDB structure tree.

Step 2 Go to the path above in the upper part of the page and select your target node.

Step 3 Click **Expand** to display the CMDB structure tree.

----End

3.3.2 Creating an Application

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 Click ⊕ on the right of **Application Metric** to create an application.

Step 5 In the displayed dialog box, set application parameters.

Table 3-2 Parameters for creating an application

Parameter	Description
Application Name	Name of an application, which cannot be empty. Enter 1 to 128 characters and start with a letter. Only digits, letters, underscores (_), and hyphens (-) are allowed.
Display Name	Application alias. The alias takes precedence over the application name to display. Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Enterprise Project	Select an enterprise project from the drop-down list. This parameter is displayed only when you use the enterprise edition.
Description	Description of the application. Enter up to 1000 characters.

Step 6 Click **Confirm**.

NOTE

After an application is created, connect it to APM for monitoring.

----End

3.3.3 Creating a Sub-application

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 Click next to your target application in the tree.

Step 5 In the displayed dialog box, set sub-application parameters.

Table 3-3 Parameters for creating a sub-application

Parameter	Description
Sub-application Name	Name of a sub-application, which cannot be empty. Enter 1 to 128 characters and start with a letter. Only digits, letters, underscores (_), and hyphens (-) are allowed.
Display Name	Display name of a sub-application, which cannot be empty. Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Description	Description of the sub-application. Enter up to 1000 characters.

Step 6 Click **Yes**.

NOTE

A maximum of three layers of sub-applications can be created.

----End

3.3.4 Configuring an Application and Sub-application

Step 1 Log in to the management console.

Step 2 Click on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 Click next to the application or sub-application name in the tree.

Step 5 Configure the application and sub-application according to **Table 3-4**.

Table 3-4 Parameters for configuring the application and sub-application

Operation	Description
Modify	Click **Modify**. In the displayed dialog box, modify the information about the application or sub-application.
Set as Default	If you select **Set as Default** for an application, it will become the default application. When you log in to the system, the default application will be selected. This option is not available for sub-applications.

Operation	Description
Delete	Click **Delete**.

Step 6 Click **Yes**.

----End

3.4 Application Metric Monitoring

3.4.1 Overview

APM Agents periodically collect performance metric data to measure the overall health status of applications. They can collect the metric data of JVM, GC, service calls, exceptions, external calls, database access, and middleware, helping you comprehensively monitor application running.

APM has strict definitions on metric data collection. Each type of data to be collected corresponds to a collector. For example, for JVM data of Java applications, a JVM collector is provided. A collector collects data of multiple metric sets. For details about collectors and metric sets, see **Collection Center**.

After collectors are deployed in the environment, monitoring items are generated. During data collection, the monitoring items determine data structures and collection behaviors.

- Collection period: A monitoring item has the same period attribute as a data collector. The default data collection period is 1 minute and cannot be changed.
- Monitoring item status: A monitoring item is enabled by default. You can disable it so that an Agent does not intercept or report the metric data. For details, see **Enabling or Disabling a Monitoring Item**.
- Collection status: Each collection instance or monitoring item has a collection status. If a collection error occurs, you can view it on the **Collection Status** tab page. A common error is that there are too many primary keys. As a result, data aggregation on the client is abnormal.

Monitoring Item Types

Agents automatically discover collection plug-ins and instantiate collectors to form monitoring items. Monitoring items are instantiated in an environment.

There are many types of collectors, which are hard to distinguish. The system backend groups collectors for easy data query.

NOTE

The **Metrics** page displays only the involved monitoring item metrics of connected applications.

Based on collector functions, monitoring items can be classified into:

- **Topology**: Displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also check the trend.
- **URL**: Monitors the external services that call the current application.
- **JVM**: Monitors basic system performance metrics.
- **Exception**: Monitors application exceptions.
- **Call**: Monitors the external services called by the current application.
- **SQL**: Monitors database access.
- **Web Container**: Monitors web containers such as Tomcat. Generally, the total number of threads, number of busy threads, and number of connections are collected to measure the overall system capacity.

Monitoring Item Configuration

Collectors corresponding to monitoring items define collection parameters. You can modify collection parameters on the page as required. These parameters will be delivered to Agents with heartbeat parameters to change collection behaviors. By default, Redis instruction content is not collected for security purposes. If necessary, modify collection parameters to collect specific instruction data. Collection parameters can also be defined on environment tags. Collectors automatically inherit collection parameter attributes of corresponding environment tags. In this way, configuration is automated.

Monitoring Item Views

On the metric monitoring details page, a monitoring item corresponds to one or more tab views, and each view corresponds to a metric set. APM provides summary tables, trend graphs, latest data tables, and original tables. For details, see **Monitoring Item Views**.

3.4.2 Application Monitoring Details

3.4.2.1 Topology

The topology displays the call relationships between services within a period. The statistics can be collected from the caller or the callee. You can also view the trend. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults. The application relationships, call data (service and instance metrics), and health status are clearly displayed.

Viewing the Topology

Step 1 Log in to the management console.

Step 2 Click  on the left and choose Management & Deployment > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment.

Step 5 Switch to the **Topology** tab page. The call trend of the selected instance is displayed.

Step 6 Click next to **Display only calls between components**.

When the button turns blue, only the calls between components are displayed.

Step 7 Click **Show All** to display all call relationships of the selected instance in a specified time range.

Step 8 Click **Reset Layout** to restore to the initial topology.

Step 9 Select the refresh mode and time. Default: **Manual Refresh**. In addition, **Automatic refresh in 1 minute**, **Automatic refresh in 5 minutes**, and **Automatic refresh in 15 minutes** are supported.

Step 10 Select a time dimension. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

----End

3.4.2.2 URL

This function monitors the calls of the current application by external services. It includes URL, Dubbo server, CSE server, CSEProvider cluster, and FunctionGraph monitoring. This type of monitoring item demonstrates the actual external status of the entire service. For example, if the average response time of a URL is long, it means that external users take a long time to query the corresponding data.

This section focuses on URL monitoring.

Going to the URL Page

Step 1 Log in to the management console.

Step 2 Click on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment. On the **URL** tab page that is displayed, check URL monitoring information of all instances.

Step 5 On the displayed **URL** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 6 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-1 Selecting a time range

Step 7 Click in the upper right corner of the list and select the metric data you want to view.

----End

Viewing URL Monitoring Data

URL summary

For common URL calls, the system collects the metrics of each URL. For details about the metrics, see **Table 3-5**.

Table 3-5 Parameters of the URL summary

Metric Set	Metric	Description
URL summary	url	URL.
	method	Request HTTP method.
	Calls	Number of times that the URL is called.
	Avg RT (ms)	Average response time of the URL in a collection period.
	Errors	Number of call errors of the URL.
	Max Concurrency	Maximum concurrency of the URL.
	Max RT (ms)	Maximum response time of the URL in a collection period.
	Apdex	Application performance index (Apdex), which indicates users' satisfaction. The value ranges from 0 to 1. The closer the value is to 1, the higher the satisfaction is.
	Exceptions	Number of exceptions of the URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.

Metric Set	Metric	Description
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

- URL invocation is the starting point of tracing. When you click a URL, the tracing page is displayed, showing the URL invocation condition in a certain period (default: 20 minutes).
- You can add a URL for tracing by referring to **Configuring URL Tracing**.
- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Status code summary

APM supports status code-based summary. The system collects the metrics of each URL. For details about the metrics, see **Table 3-6**.

Table 3-6 Parameters of status code summary

Metric Set	Metric	Description
Status code summary	code	Status code.
	Count	Number of times that the status code occurred.
	Latest URL	Sample URL which returns the status code in a collection period.

- Click a status code in the **code** column. The tracing page is displayed, showing the invocation condition of the status code of the selected instance in the environment in last 20 minutes (default).
- Click a number in the **Count** column to view the trend of the status code in a specified period.
- Click the latest URL to view the invocation details of the corresponding status code.

Cluster summary

APM can summarize metrics by cluster. For details about the metrics, see **Table 3-7**.

Table 3-7 Parameters of the cluster summary

Metric Set	Parameter	Description
Cluster summary	Cluster ID	Cluster ID of the caller.
	Calls	Number of times the cluster is called.
	Avg RT (ms)	Average response time in a collection period.
	Errors	Number of times that the cluster fails to be called.
	Max Concurrency	Maximum concurrency of the cluster.
	Max RT (ms)	Slowest call time in a collection period.

Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Overview

View the metric trend of the selected instance on the **Overview** tab page. For details about the metrics, see **Table 3-8**.

Table 3-8 Overview metrics

Metric Set	Metric	Description
Overview	Total Requests	Total number of URL requests.
	Avg RT (ms)	Average response time of the URL.
	Errors	Total number of URL errors.
	Apdex	Users' satisfaction with the URL.

3.4.2.3 JVM

This function monitors JVMInfo, JVMMonitor, GC, thread, and JavaMethod.

Going to the JVM Page

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment.

Step 5 Click the **JVM** tab. By default, the JVMMonitor information of all instances is displayed.

Step 6 On the displayed **JVM** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-2 Selecting a time range

----End

Viewing JVM Information

On the **JVM** tab page, view the JVMInfo metrics of the corresponding instance. For details about the metrics, see **Table 3-9**.

Table 3-9 JVMInfo metrics

Metric Set	Metric	Description
JVMInfo	JavaAgent Version	Java Agent version.
	Started	JVM startup time.
	Startup Parameter	JVM startup parameter.
	Java Class Library Path	Java class library path.
	Java Version	Java version.
	Java Specification Version	Java specification version.
	OS	OS name.
	OS Version	OS version.

Metric Set	Metric	Description
	arch	CPU architecture.
	Processors	Number of processors.
	SDK Version	SDK version.

Viewing JVM Monitoring Data

APM monitors JVM metrics. For details about the metrics, see **Table 3-10**. JVM monitoring metrics are displayed in graphs, so that you can view and analyze JVM monitoring data more easily.

Table 3-10 JVM monitoring metrics

Metric Set	Metric	Description
Thread	Current Threads	Number of current threads.
	Deadlock Threads	Number of deadlock threads.
	Daemon Threads	Number of daemon threads.
	Started Threads	Number of started threads.
	Peak Threads	Peak number of threads.
Thread Status	Waiting Threads	Number of waiting threads.
	Terminated Threads	Number of threads in the terminated state.
	Runnable Threads	Number of runnable threads.
	Blocked Threads	Number of blocked threads.
	New Threads	Number of new threads.
	Timed Waiting Threads	Number of threads that timed out.
Memory	Used Non-Heap Memory	Size of the used non-heap memory.
	Used Heap Memory	Size of the used heap memory.
	Used Direct Memory	Size of the used direct memory.
Class loading	Current Classes	Number of current classes.
	Total Loaded Classes	Total number of loaded classes.
	Unloaded Classes	Number of unloaded classes.

Metric Set	Metric	Description
Memory pool	committed(M)	Size of available memory.
	init(M)	Size of the initialized memory.
	max(M)	Size of the maximum memory.
	name	Memory pool name.
	used(M)	Size of the used memory.
CPU	CPU Usage	CPU usage of the Java process.

Viewing GC Information

APM monitors GC metrics. For details about the metrics, see **Table 3-10**.

Table 3-11 GC metrics

Metric Set	Metric	Description
GC statistics	Full GC (times)	Number of full GC times in a collection period.
	Full GC Duration (ms)	Full GC duration in a collection period.
	Young GC (times)	Number of young GC times in a collection period.
	Young GC Duration (ms)	Young GC duration in a collection period.
GC Details	GC Type	GC type, which can be **major** or **minor**.
	GC Cause	GC cause.
	Count	Number of times that GC has occurred.
	Total GC Duration (ms)	GC duration.
	Max GC Duration (ms)	Time consumed by the slowest GC.
	GC Recycler	GC recycler name.
	Slowest GC Details	Details about the slowest GC.

- Click the digits in blue (such as those in the **Count**, **Total GC Duration (ms)**, or **Max GC Duration (ms)** column) to view the corresponding GC trend graph in a certain period (default: 20 minutes).

- On the GC details area, you can view the GC type, GC cause, count, total GC duration (ms), maximum GC duration (ms), GC recycler, and slowest GC details (details and history).

Viewing Threads

You can view the thread details of the corresponding instance on APM. For details, see **Table 3-12**.

Table 3-12 Thread metrics

Metric Set	Metric	Description
Thread details	Thread Name	Thread name.
	Threads	Number of threads.
	CPU Time (ms)	Thread CPU time.
	Memory (MB)	Memory (MB).
	Thread Stack	Thread stack.

- Click a number in the **Threads** column to view the trend of the thread in a specified period.
- Click **Detail** in the **Thread Stack** column to view the thread details.
- Click **History** in the **Thread Stack** column to view the historical thread stack list.

Viewing Java Methods

1. By default, APM does not monitor Java methods. To monitor them, **configure the JavaMethod monitoring item** first.
2. After the configuration is complete, the system monitors the methods and classes of JavaMethod.
3. On the **JVM** page, select a target instance and **JavaMethod** to view details. For details, see **Table 3-13**.

Table 3-13 JavaMethod metrics

Metric Set	Metric	Description
JavaMethod	Class	Class of a Java method.
	Method	Method.
	Calls	Number of times that the method is called.
	Avg RT (ms)	Average response time.

Metric Set	Metric	Description
	Errors	Number of times that the method fails to be called.
	Max Concurrency	Maximum concurrency of the method.
	Max RT (ms)	Maximum response time of the method.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

- Click a number (such as those in the **Calls** or **Errors** column) to view the trend of the thread in a specified period.

3.4.2.4 Exception

This function monitors application exception logs. Take the monitoring of Java exception logs as an example. Once you use the log system to print logs, they will be collected by APM. The exception collection type varies according to the collector type.

Viewing Exception Logs

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Exception** tab. By default, exception logs of all instances are displayed. For details about the metrics, see **Table 3-14**.

Table 3-14 Exception and log parameters

Metric Set	Parameter	Description
Exception	Class	Exception class
	Exception Type	Exception type
	Log Type	Exception log type
	Total Exceptions	Number of times that an exception has occurred
	Message	Message returned when the exception occurred
	Error Stack	Error stack
Log Version	Log Type	Log type
	Version	Log version

- Click a number in blue in the **Total Exceptions** column to view the trend of the total exceptions in a specified period.
- Click the blue text in the **Message** column to view the message time and content.
- Click **Detail** in the **Error Stack** column to view exception details.
- Click **History** in the **Error Stack** column to view the historical error stack list.
- Click the blue text in the **Version** column to view details.

Step 6 On the **Exception** tab page, select a target instance and then select **Exception** to view the exception monitoring data.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-3 Selecting a time range

Step 8 Click in the upper right corner of the list and select the metric data you want to view.

----End

3.4.2.5 Call

This function monitors the calls of external services by the current application. It covers CSEConsumer, ApacheHttpClient, ApacheHttpAsyncClient, DubboConsumer, and HttpClient monitoring.

This section focuses on HttpClient monitoring.

Going to the Call Page

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment.

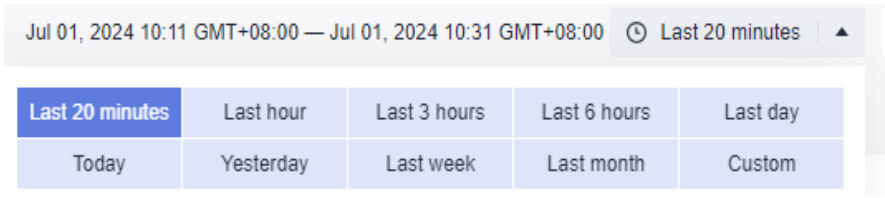
Step 5 Click the **Call** tab. By default, the HttpClient monitoring information of all instances is displayed.

Step 6 On the displayed **Call** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-4 Selecting a time range



Step 8 Click in the upper right corner of the list and select the metric data you want to view.

----End

Viewing HttpClient Monitoring Data

URL summary

The HttpClient monitoring system collects the metrics of each URL. For details about the metrics, see **Table 3-15**. Click in the upper right corner of the list and select the metric data you want to view.

Table 3-15 Parameters of URL summary under HttpClient monitoring

Metric Set	Metric	Description
URL summary	url	Called URL.
	method	HTTP method of the URL.
	Calls	Number of times that the URL is called.
	Avg RT (ms)	Average response time of the called URL.
	Errors	Number of call errors of the URL.
	Max Concurrency	Maximum concurrency of the URL.
	Max RT (ms)	Maximum response time of the called URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
	Error Trace	ID of the trace that encounters an error in a collection period.
	Slowest Trace	ID of the slowest trace in a collection period.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click the text in blue (such as those in the **Slowest Trace** or **Error Trace** column) to view more details.

Cluster summary

APM can summarize external call metrics by cluster. For details about the metrics, see **Table 3-16**.

Table 3-16 Parameters of cluster summary under HttpClient monitoring

Metric Set	Metric	Description
Cluster summary	envId	Cluster ID of the called party.
	hostUri	Host URL
	Calls	Number of times that the cluster URL is called.
	Avg RT (ms)	Average response time for calling the cluster URL.
	Errors	Number of call errors of the URL.
	Max RT (ms)	Maximum response time for calling the cluster URL.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Status code summary

APM can summarize external call metrics by status code. For details about the metrics, see **Table 3-17**.

Table 3-17 Parameters of status code summary under HttpClient monitoring

Metric Set	Metric	Description
Status code summary	code	Status code.
	Count	Number of times that the status code occurred.
	Latest URL	URL that returns the status code.

- Click a status code in the **code** column. The tracing page is displayed, showing the invocation condition of the status code of the selected instance in the environment in last 20 minutes (default).
- Click a number in the **Count** column to view the trend of the status code in a specified period.
- Click the latest URL to view the invocation details of the corresponding status code.

Exception

On the **Exception** tab page, view the exception statistics about HttpClient calls. For details about the metrics, see **Table 3-18**.

Table 3-18 Parameters of HttpClient monitoring exceptions

Metric Set	Metric	Description
Exception	causeType	Exception class.
	exceptionType	Exception type.
	Count	Number of times the exception occurred.
	Error Message	Message returned when the exception occurred.
	Error Stack	Exception stack information.

- Click a number in blue in the **Count** column to view the trend of the thread in a specified period.
- Click the text in blue in the **Error Message** column to view message details.
- Click **Detail** in the **Error Stack** column to view exception details.
- Click **History** in the **Error Stack** column to view the historical error stack list.

Overview

On the **Overview** tab page, view the metrics of the selected instance. For details about the metrics, see **Table 3-19**.

Table 3-19 Overview parameters of HttpClient monitoring

Metric Set	Metric	Description
Overview	Calls	Total number of calls.
	Avg RT (ms)	Average response time
	Errors	Total number of errors.

3.4.2.6 SQL

This function monitors database access. The databases that can be monitored include the C3P0, Cassandra, ClickHouse, DBCP, Druid, EsRestClient, GaussDB, Hikari, Jetcd, ObsClient, MySQL, PostgreSQL, Oracle, HBase, and MongoDB. APM collects details about executed statements to help you locate performance problems in code.

This section focuses on MySQL database monitoring.

Going to the SQL Page

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **SQL** tab. By default, the MySQL database information of all instances is displayed.

Step 6 On the displayed **SQL** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-5 Selecting a time range

Step 8 Click ⚙ in the upper right corner of the list and select the metric data you want to view.

----End

Viewing MySQL Database Monitoring Data

SQL summary

APM can monitor MySQL databases by SQL. For details about the metrics, see Table 3-20. Click ⚙ in the upper right corner of the list and select the metric data you want to view.

Table 3-20 SQL summary parameters

Metric Set	Metric	Description
SQL monitoring	sql	Unique ID of the SQL statement, which is used for alarm configuration.
	SQL Statement	SQL statement.
	Calls	Number of times that the SQL statement is called.
	Avg RT (ms)	Average response time (ms).
	Errors	Number of errors that the SQL statement encounters.
	Rows Read	Number of read rows of the SQL statement.
	Rows Updated	Number of updated rows of the SQL statement.
	Max Concurrency	Maximum concurrency of the SQL statement.
	Max RT (ms)	Maximum response time of the SQL statement.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–200 ms	Number of requests with 100 ms–200 ms response time.
	200 ms–1s	Number of requests with 200 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
	Slowest Trace	ID of the slowest trace in a collection period.
	Error Trace	ID of the trace that encounters an error in a collection period.

- Click an SQL statement to view details.
- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

- Click a slow or an error trace to view its details.

Database summary

APM can summarize MySQL database metrics by database. For details about the metrics, see **Table 3-21**.

Table 3-21 Database summary parameters

Metric Set	Metric	Description
Database connections	db	Database name.
	Connections Created	Number of connections created by the database.
	Connections Destroyed	Number of the database's connections that have been destroyed.
	Avg RT (ms)	Average response time (ms).
	Calls	Number of times that the database is called.
	Errors	Number of errors that the database encounters.
	Rows Read	Number of rows read from the database.
	Rows Updated	Number of rows updated in the database.
	Max RT (ms)	Maximum response time of the database.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–200 ms	Number of requests with 100 ms–200 ms response time.
	200 ms–1s	Number of requests with 200 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.

Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.

Exception

On the **Exception** tab page, view exception statistics about SQL calls. For details about the metrics, see **Table 3-22**.

Table 3-22 Exception parameters

Metric Set	Metric	Description
Exception	causeType	Exception class.
	exceptionType	Exception type.
	Count	Number of exceptions.
	SQL	SQL statement that encounters an exception.
	Error Stack	Exception stack information.
	Message	Error message.

Overview

On the **Overview** tab page, view the call trend of the selected instance. For details about the metrics, see **Table 3-23**.

Table 3-23 Overview parameters

Metric Set	Metric	Description
Overview	Calls	Total number of calls.
	Rows Read	Total number of read rows.
	Avg RT (ms)	Average response time (ms).
	Errors	Total number of errors.
	Rows Updated	Number of rows updated in the database.

Info

On the **Info** tab page, view the driver version information. Click the text in blue to view more details.

Viewing Druid Connection Pool Monitoring Data

The Druid connection pool monitoring system collects data sources, connection details, additional configurations, and exception information. You can click  in the upper right corner of the list to customize the columns you want to view. For details about the metrics, see **Table 3-24**.

Table 3-24 Druid connection pool parameters

Metric Set	Metric	Description
Data source	Connection Address	Connection address.
	Driver	Driver name.
	Initialized Connections	Number of initialized connections.
	Min Idle Connections in Pool	Minimum of idle connections in a pool.
	Max Idle Connections in Pool	Maximum number of idle connections in a pool.
	Max Connections in Pool	Maximum number of connections in a pool.
	Idle Connections	Number of idle connections.
	Max Idle Connections	Maximum number of idle connections.
	Active Connections	Number of active connections.
	Max Active Connections	Maximum number of active connections.
	Waiting Threads	Number of waiting threads.
	Max Waiting Threads	Maximum number of waiting threads.
	Upper Limit for Waiting Threads	Upper limit for waiting threads.
	Total Connections	Total number of connections.
Connection details	Connection Address	Connection address.
	Calls	Number of calls.
	Total RT (ms)	Total response time (ms).
	Avg RT (ms)	Average response time (ms).
	Errors	Number of errors.
	Max Concurrency	Maximum number of concurrent connections.
	Max RT (ms)	Maximum response time.
	0 ms–10 ms	Number of requests with 0 ms–10 ms response time.
	10 ms–100 ms	Number of requests with 10 ms–100 ms response time.
	100 ms–500 ms	Number of requests with 100 ms–500 ms response time.

Metric Set	Metric	Description
	500 ms–1s	Number of requests with 500 ms–1s response time.
	1s–10s	Number of requests with 1s–10s response time.
	10s–n	Number of requests with response time longer than 10s.
Additional configuration	Connection Address	Connection address.
	Max Wait (ms)	Maximum waiting time.
	Test on Borrow	Whether to verify the validity of a connection before obtaining it from the connection pool.
	Test on Return	Whether to verify the validity of a connection when it is returned.
	Test While Idle	Whether to verify the validity of an idle connection when an application applies for it from the pool
	Remove Abandoned	Whether to automatically reclaim timeout connections.
	Remove Abandoned TimeoutMillis (ms)	If a connection in the pool is not returned within the specified duration, the connection will be reclaimed.
	Remove Abandoned Count	Number of timeout connection reclaims.
	Min Evictable Idle TimeMillis (ms)	Minimum idle time of a connection in the pool.
	Time Between EvictionRunsMillis (ms)	Interval for checking the validity of idle connections.
Exception	causeType	Exception class.
	Exception Type	Exception type.
	Count	Number of times the exception occurred.
	Error Message	Message returned when the exception occurred.
	Error Stack	Exception stack information.
Version	Driver Version	Driver version.

- Click a number in blue (such as those in the **Calls** or **Avg RT (ms)** column) to view more details.
- Click the text in blue (such as those in the **Driver** or **Driver Version** column) to view more details.

3.4.2.7 Web Container

This function monitors web containers, including Tomcat. This section focuses on Tomcat monitoring.

Going to the Web Container Page

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Web Container** tab. By default, the Tomcat monitoring information of all instances is displayed. For details about the metrics, see **Table 3-25**.

Table 3-25 Tomcat monitoring parameters

Metric Set	Metric	Description
Tomcat port monitoring	name	Port name.
	Current Threads	Number of current threads on the port.
	Busy Threads	Number of busy threads on the port at the time of collection.
	Peak Busy Threads	Maximum number of busy threads on the port in a collection period.
	Max Threads	Maximum number of threads on the port.
	Max Connections	Maximum number of connections on the port.
	Current Connections	Number of current connections of the port at the time of collection.
	Peak Connections	Maximum number of connections on the port in a collection period.
Version	Version	Tomcat version.

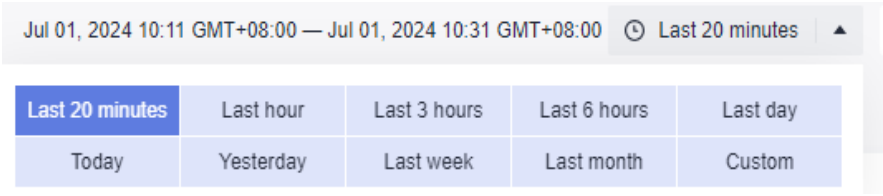
- Click a number in blue (such as those in the **Current Threads**, **Busy Threads**, or **Peak Busy Threads** column) to view the trend graph of the target web container in the specified period.
- Click a version in the **Version** column to view details.

Step 6 On the displayed **Web Container** tab page, select a target instance and monitoring item to view the monitoring data in different metric sets.

Step 7 Select a time range. Default: **Last 20 minutes**.

Options: **Last 20 minutes**, **Last hour**, **Last 3 hours**, **Last 6 hours**, **Last day**, **Today**, **Yesterday**, **Last week**, **Last month**, or **Custom**.

Figure 3-6 Selecting a time range



Step 8 Click in the upper right corner of the list and select the metric data you want to view.

----End

3.4.3 Application Monitoring Configuration

3.4.3.1 Configuration Details

You can define collection parameters for some collectors corresponding to monitoring items.

NOTE

On the **Monitoring Item** tab page, only monitoring items related to the connected application are displayed.

Configuring a Monitoring Item

Step 1 Log in to the management console.

Step 2 Click on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Monitoring Item** tab.

Step 6 Locate the row that contains the target monitoring item and click **Modify** in the **Operation** column.

Step 7 On the displayed page, edit the monitoring configuration. For details, see the corresponding section.

Step 8 Click **Yes**.

----End

Enabling or Disabling a Monitoring Item

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Monitoring Item** tab.

Step 6 Locate the row that contains the target monitoring item and enable or disable it.

----End

3.4.3.2 Configuring the MySQL Monitoring Item

On the **Modify MySQL Monitoring Configuration** page, set the following parameters:

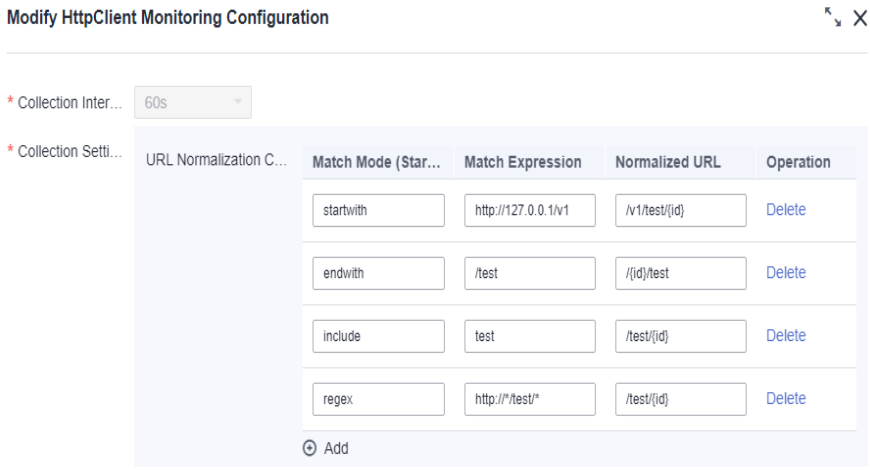
- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Collect Original SQL**: This function is disabled by default. In that case, only SQL statements without values are collected, for example, **select name from user where id=?**. When this function is enabled, SQL statements with values are collected, for example, **select name from user where id=1**.
- **shardTableName**: specified when you need to aggregate multiple tables into one table. For example, there are two tables: **UserTable_1** and **UserTable_2**. By default, two SQL statements (**select name from UserTable_1** and **select name from UserTable_2**) are displayed on the SQL monitoring page. If you set **shardTableName** to **UserTable**, tables starting with **UserTable** are aggregated into the same table. Only one SQL statement (**select name from UserTable**) is displayed on the SQL monitoring page.

3.4.3.3 Configuring the HttpClient Monitoring Item

On the **Modify HttpClient Monitoring Configuration** page, set the following URL normalization parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- URL normalization is used to aggregate URLs that meet the conditions you set. For example, **http://localhost/rest/v1/test/123** and **http://localhost/rest/v1/test/234** can be aggregated into **http://localhost/rest/v1/test/*{id}***.

Figure 3-7 Configuring the HttpClient monitoring item



Normalization Methods

There are four normalization methods: **Startwith**, **Endwith**, **Include**, and **Regex**.

- **Startwith**: URLs starting with a certain expression are counted as normalized URLs. For example, URLs starting with **http://127.0.0.1/v1** are aggregated into **/v1/test/***{id}*, as shown in **Figure 3-7**.
- **Endwith**: URLs ending with a certain expression are counted as normalized URLs. For example, URLs ending with **/test** are aggregated into **/***{id}***/test**, as shown in **Figure 3-7**.
- **Include**: URLs containing a certain expression are counted as normalized URLs. For example, URLs containing **test** are aggregated into **/test/***{id}*, as shown in **Figure 3-7**.
- **Regex**: URLs that meet the wildcard expression are counted as normalized URLs. For details about the wildcard rules, see **Table 3-26**.

Table 3-26 Wildcard description

Wildcard	Description
?	Matches any character.
*	Matches zero, one, or more characters.
**	Matches zero, one, or more directories.

Usage Example

The following is an example:

URL Path	Description
/app/p?ttern	Matches files such as **/app/pattern** and **/app/pAttern**, excluding **/app/pttern**.

URL Path	Description
/app/*.x	Matches all **.x** files in the **app** directory.
/**/example	Matches **/app/example**, **/app/foo/example**, and **/example**.
/app/**/dir/file.*	Matches **/app/dir/file.jsp**, **/app/foo/dir/file.htm**, **/app/foo/bar/dir/file.pdf**, and **/app/dir/file.c**.
/**/*.jsp	Matches all **.jsp** files.

3.4.3.4 Configuring the URL Monitoring Item

On the **Modify URL Monitoring Configuration** page, set the following parameters:

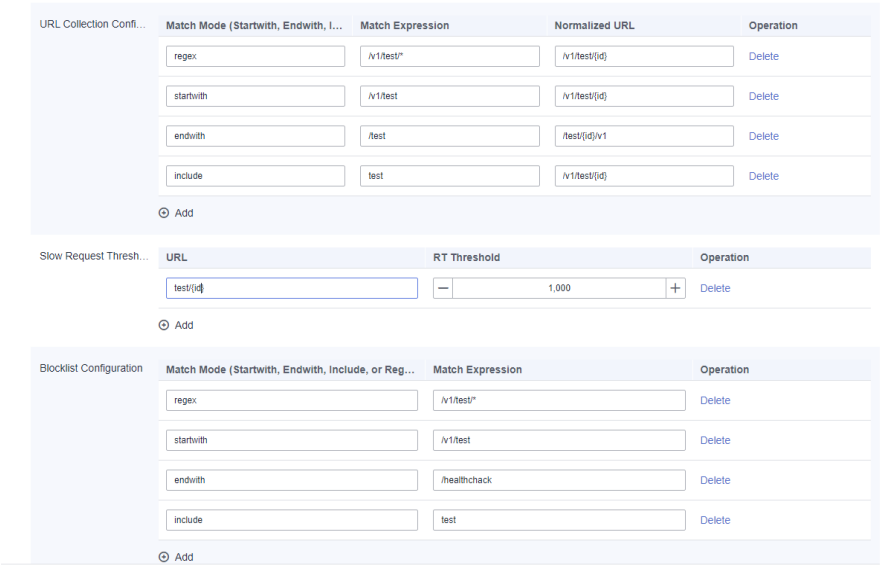
CAUTION

For security purposes, do not contain sensitive data in headers, URL parameters, cookies, or other parameters.

Parameter	Description	Example
Collection Interval	The default value is **60s** and cannot be changed.	60s
Key for Header Value Interception	Key specified for collecting values in headers. The collected information can be seen in the trace parameters.	Host
Key for Parameter Value Interception	Key specified for collecting values in URLs. The collected information can be seen in the trace parameters. Take **http://127.0.0.1/test?param=123** as an example. If the key is set to **param**, value **123** can be seen in the trace parameters.	param
Key for Cookie Value Interception	Key specified for collecting values in cookies. The collected information can be seen in the trace parameters.	testKey

Parameter	Description	Example
URL Collection Configuration	URLs that meet the conditions you set are aggregated. For example, **/rest/v1/test/123** and **/rest/v1/test/234** can be aggregated into **/rest/v1/test/***{id}*. The configuration method is the same as that described in **HttpClient URL Normalization**.	**Figure 3-8** -
Blocklist Configuration	Data of URLs that meet the conditions you set will not be collected. The configuration method is the same as that described in **HttpClient URL Normalization**.	**Figure 3-8** -
Service Code Length	Maximum length of the response body to be parsed to prevent the performance from being affected. Content that beyond this limit will not be parsed, but corresponding service status codes are regarded as normal by default.	-
Key for Service Code Interception	Key specified for collecting service status codes. If the custom API returned content is **{"errorCode":500,"errorMsg":"error msg"}**, set this parameter to **errorCode**.	errorCode
Normal Service Code	If this status code is returned, traces are regarded as normal. If other codes are returned, traces are regarded as abnormal.	-
Slow Request Threshold	Global response time threshold. The default value is **800**. Requests with the response time longer than 800 ms are regarded as slow requests. The sampling ratio of slow requests will be increased.	-
URL Configuration	Response time threshold separately set for a URL. If the response time of this URL exceeds the threshold, the sampling rate of this URL will be increased. If this parameter is not set, the global slow request threshold is used by default.	**Figure 3-8** -
Error Code	Options: **400 or greater** and **500 or greater** (default). By default, if status code 500 or greater occurs, the system regards that there is an error.	-

Figure 3-8 Example

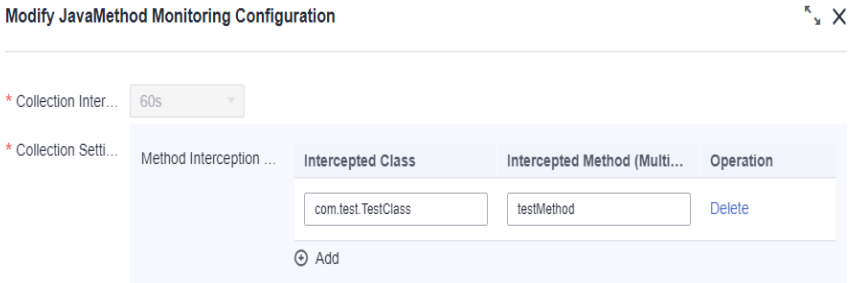


3.4.3.5 Configuring the JavaMethod Monitoring Item

On the **Modify JavaMethod Monitoring Configuration** page, set method interception parameters.

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Method Interception Configuration**: is used to collect specified service methods. The method data is displayed on the JavaMethod metric page and in traces.
- **Intercepted Class**: name of the fully-qualified class to be collected. Both the package name and class name need to be specified.
- **Intercepted Method**: name of the method to be collected. If multiple methods exist, separate them by commas (,), for example, **testMethod1,testMethod2**.

Figure 3-9 Configuring the JavaMethod monitoring item



3.4.3.6 Configuring the Druid Monitoring Item

On the **Modify Druid Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.

- **TraceReportTimeSpanThreshold(ms)**: threshold for reporting getConnection method traces. If the threshold is not exceeded, such traces will not be reported. The default value is **1**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Get pool info when calling getConnection**: specifies whether to obtain the pool information when calling the **getConnection** method. The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.

3.4.3.7 Configuring the ApacheHttpAsyncClient Monitoring Item

On the **Modify ApacheHttpAsyncClient Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.

3.4.3.8 Configuring the Redis Monitoring Item

On the **Modify Redis Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Parameter Parsing**: The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Parameter Length**: The default value is **1000**. If you select **Use default value**, the value of the inherited tag is preferentially used.
- **Distinguish Redis Ports**: The default value is **No**. If you select **Use default value**, the value of the inherited tag is preferentially used.

3.4.3.9 Configuring the Jedis Monitoring Item

On the **Modify Jedis Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.10 Configuring the HBase Monitoring Item

On the **Modify HBase Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.11 Configuring the ApacheHttpClient Monitoring Item

On the **Modify ApacheHttpClient Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.12 Configuring the Tomcat Monitoring Item

On the **Modify Tomcat Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.13 Configuring the EsRestClient Monitoring Item

On the **Modify EsRestClient Monitoring Configuration** page, set the following parameter:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Index Normalization Configuration**: The system matches indexes based on a regular expression and then normalizes them.

3.4.3.14 Configuring the WebSocket Monitoring Item

On the **Modify WebSocket Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.15 Configuring the KafkaProducer Monitoring Item

On the **Modify KafkaProducer Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.16 Configuring the Hikari Monitoring Item

On the **Modify Hikari Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **TraceReportTimeSpanThreshold(ms)**: The default value is **1**. If **Use default value** is selected, the value of the inherited tag is preferentially used.
- **Get pool info when calling getConnection**: The default value is **No**. If **Use default value** is selected, the value of the inherited tag is preferentially used.

3.4.3.17 Configuring the Exception Monitoring Item

On the **Modify Exception Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Determine Trace Exception upon Log Error Detection**: The default value is **Yes**. If **Use default value** is selected, the value of the inherited tag is preferentially used.

3.4.3.18 Configuring the Thread Monitoring Item

On the **Modify Thread Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.

3.4.3.19 Configuring the GC Monitoring Item

On the **Modify GC Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.20 Configuring the JVMInfo Monitoring Item

On the **Modify JVMInfo Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.3.21 Configuring the JVMMonitor Monitoring Item

On the **Modify JVMMonitor Monitoring Configuration** page, set the following parameters:

- **Collection Interval**: The default value is **60s** and cannot be changed.
- **Call Chain Stack Collection Threshold**: When the request latency exceeds the threshold, the stack is automatically printed. The default value is **0** and the maximum value is **10000**.

3.4.3.22 Configuring ProbeInfo Monitoring Item

On the **Modify ProbeInfo Monitoring Configuration** page, set the following parameter:

Collection Interval: The default value is **60s** and cannot be changed.

3.4.4 Monitoring Item Views

APM supports summary tables, trend graphs, latest data tables, and original data tables.

- Summary table: records the summary calculation results based on the primary key metric within a period. You can click a number or character string in the summary table to view the trend graph of the primary key metric.
- Trend graph: displays the trend of a primary key metric in a period. A trend graph may have breakpoints, indicating that no data is collected in this period. There are multiple reasons why data is not collected. For example, collectors do not collect the metrics with zero calls or the data may be lost.
- Original data table: For character strings, no trend graphs can be generated. Therefore, original data tables are used. Each row indicates the mapping between a time and a value.
- Latest data table: displays the latest data. You can click a data record to view its trend graph.

NOTE

The view of each monitoring item is configured in the background and has not been opened. You can check views together with corresponding background metric sets. For details, see **Metric Sets**.

3.5 Tracing

When the calls between enterprise microservices are complex, APM Agents sample some requests, and intercept corresponding requests and subsequent call information. For example, in the scenario where service A calls service B and then calls service C, after service A receives a request, APM determines whether to trace the request based on the intelligent sampling algorithm.

Intelligent Sampling Algorithm

APM uses the intelligent sampling algorithm to determine whether to trace requests.

- If a request needs to be traced, a trace ID is generated and details (events) about some important methods (generally the tree structure with the parent-child relationship) under service A are intercepted. At the same time, the trace ID is transparently transmitted to service B. The important methods under service B are also intercepted. The trace ID is also transparently transmitted to service C. Some methods under service C are intercepted in a similar way as those under services B and A. Each node respectively reports event information and an association relationship can be formed based on the trace ID. In this way, you can view the call details of the entire request based on the trace ID.
- If a request does not need to be traced, no trace ID is generated. Service B does not receive the trace ID and uses the same algorithm as service A to determine whether to perform tracing.
- After data is reported, APM stores not only all event details, but also the root event (called span) information of each service for subsequent trace search. Generally, you search for the span information and then obtain the overall trace details based on the trace ID in the span information.
- By default, the intelligent sampling policy is used. There are three types of URLs: error URLs, slow URLs (use the default 800 ms or customize a threshold), and normal URLs. The sampling ratio of each type of URL is calculated separately. For APM, statistics are collected and reported every minute. In the first collection period, all URLs are regarded as normal for sampling. In the second collection period, URLs are classified into error, slow, and normal URLs based on the statistics collected in the previous period.
  - Sampling rate of error URLs: If the CPU usage is less than 30%, 100 records are collected per minute. If the CPU usage is greater than or equal to 30% but less than 60%, 50 records are collected per minute. If the CPU usage is greater than or equal to 60%, 10 records are collected per minute. At least two records are collected for each URL.
  - Sampling rate of slow URLs: If the CPU usage is less than 30%, 100 records are collected per minute. If the CPU usage is greater than or equal to 30% but less than 60%, 50 records are collected per minute. If the CPU usage is greater than or equal to 60%, 10 records are collected per minute. At least two records are collected for each URL.
  - Sampling rate of normal URLs: If the CPU usage is less than 30%, 20 records are collected per minute. If the CPU usage is greater than or equal to 30% but less than 60%, 10 records are collected per minute. If

the CPU usage is greater than or equal to 60%, 5 records are collected per minute. At least one record is collected for each URL.

The advantage of the preceding algorithm is that once the trace information is generated, the link is complete, helping you make correct decisions. If a large number of URLs are called, abnormal requests may fail to be collected. In this case, you can collect metrics to locate system exceptions.

Trace Search

This function is used to search for span information, that is, the root event of a node. A trace can be found in multiple environments. For example, in the scenario where service A calls service B and then calls service C, the same trace may be found from services A, B, and C.

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Tracing**.

Step 4 Specify the following search criteria or add custom criteria to query traces.

Table 3-27 Search criteria of traces

Search Criterion	Description	Mandatory
Application	Application to which the trace belongs.	Yes
Region	Region where the trace is located.	Yes
Component	Component to which the trace belongs.	No
Environment	Environment to which the trace belongs.	No
Instance	Instance to which the trace belongs.	No
URL	Trace URL, which can be a REST URL or real URL. A REST URL contains a variable name, for example, **/apm/get/{*id*}**. A real URL indicates an actual URL.	No
Exact Search	Whether to perform exact match on URLs. If this option is selected, exact match is performed. If this option is not selected, fuzzy match is performed.	No
Call Method	HTTP method of the trace.	No
Status Code	HTTP status code returned by the trace.	No

Search Criterion	Description	Mandatory
Response Time	Response time range of the trace. You can specify the minimum and maximum response time to search for the trace or leave them empty.	No
Exception or Not	Whether to filter the traces that are regarded as exceptions.	No
Trace ID	If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No
Custom Parameter	• Search for traces by call parameter. The format is **key=value**. Example: **exceptionMsg=failed**. • Configure required parameters before you search for traces by custom parameter. For example, if you have configured **Key for Header Value Interception**, **Key for Parameter Value Interception**, and **Key for Cookie Value Interception** for URL monitoring, you can set **key=value** to search, for example, **httpMethod=POST**. For details about how to configure URL monitoring, see **Configuring the URL Monitoring Item**.	No
Global Trace ID	Global ID of a trace. If you specify this parameter, other search criteria become invalid and the search will be performed based on the trace ID you specify.	No
Application Code	If you have configured **Service Code Length**, **Key for Service Code Interception**, and **Normal Service Code** for URL monitoring, responses' application codes will be collected. You can search information based on application codes. Generally, the value of **Application Code** is the same as the value of **Normal Service Code**. For details about how to configure URL monitoring, see **Configuring the URL Monitoring Item**.	No

----End

Viewing Trace Details

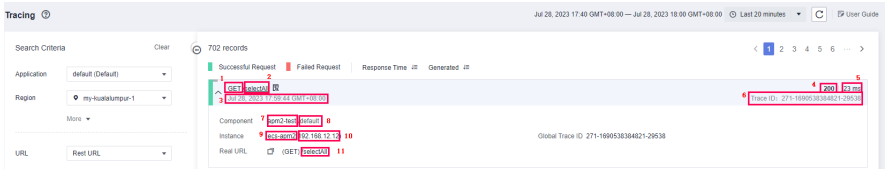
Viewing Basic Information About the Trace Filtered Based on the Search Criteria

Only the traces of successful or failed requests can be displayed. The following uses **Successful Request** as an example.

- Click the green button next to **Successful Request**, only the traces of successful requests are displayed. The red button next to **Failed Request** becomes dimmed.
- If you click the green button again, both the traces of successful and failed requests are displayed. The red button is no longer dimmed.
- The green and red buttons cannot be dimmed at the same time.

In the displayed trace list, click > next to the target trace to view its basic information, as shown in the following figure.

Figure 3-10 Basic information about a trace



Parameter description:

1. HTTP method of the trace.
2. REST URL of the trace. A REST URL contains a variable name, for example, **/apm/get/{*id*}**. You can click the URL to go to the trace details page.
3. Start time of the trace.
4. HTTP status code returned by the trace.
5. Response time of the trace.
6. Trace ID.
7. Component to which the trace belongs.
8. Environment to which the trace belongs.
9. Host of the instance to which the trace belongs.
10. IP address of the instance to which the trace belongs.
11. Actual URL of the trace.

You can also click a specific URL on the monitoring item view page, for example, the table view of the URL monitoring item. In this way, you can quickly search for required trace information based on preset search criteria.

Viewing the Complete Information About the Trace, Including Local Method Stacks and Remote Call Relationships

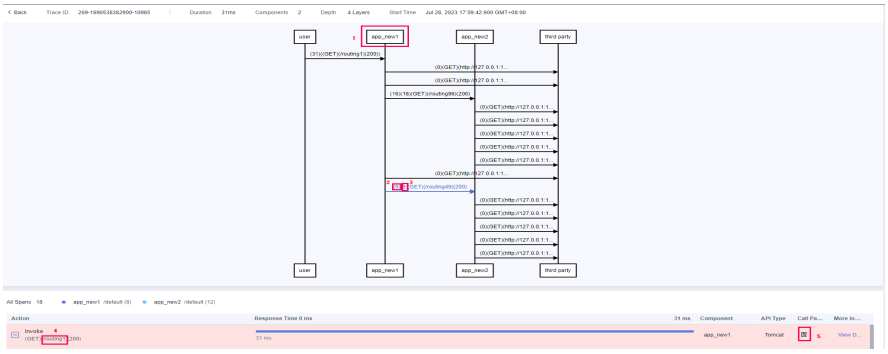
Click the name of a trace to view its details, as shown in the following figure.

- The upper part is the sequence diagram of the trace, which shows complete call relationships between components. This diagram contains the information

about the client and server corresponding to each call. The lower the line is, the later a call occurs.

- The lower part lists the method stack details of the trace. Each line indicates a method call. You can view the detailed method call relationships of the trace. By default, only component methods supported by JavaAgents are displayed. To display application methods, configure the application methods to be intercepted during JavaMethod configuration.

Figure 3-11 Call relationship



Parameter description:

1. Component and environment to which the called API belongs
2. Response time (unit: ms) of the client. You can hover the mouse pointer over this digit to view more details.
3. Response time (unit: ms) of the server.
4. Key parameter of the method in the trace method stack. For example, for a Tomcat entry method, a real URL is displayed. For a MySQL call method, an executed SQL statement is displayed.
5. Extended data of the trace method. Generally, parameters related to the method are displayed.

3.6 Application Topology

On the tracing page, you can view the topology of a single call, as well as the overall topology between different services based on collected metric data. There are two types of application topologies:

- Single-component topology: topology of a single component under an environment. You can also view the call relationships of direct and indirect upstream and downstream components.
- Global application topology: topology of some or all components under an application.

Each line in the topology indicates the call relationship between services within a period. The statistics can be collected from the calling or called party. You can click a line to view the call trend on the right. The topology can also display the call relationships between middleware. On the topology, you can view the call relationships between services and check whether the calls between services are normal to quickly locate faults.

Viewing the Topology of a Component

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click  next to the target environment.

Step 5 Click the **Topology** tab to view the call and dependency relationships of the component.

Click a line between components. The detailed data is displayed on the right.

Enable **Display only calls between components** to shield the calls of external components, or click **Show All** to display the calls between all components except the central node.

----End

Viewing the Global Topology

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click an application. The application details page is displayed.

Step 3 Click the **Global Topology** tab to view the call and dependency relationships of all components under the application.

Click a line between components. The detailed data is displayed on the right.

Use tags to filter calls or enable **Display only calls between components** to shield the calls of external components.

----End

3.7 URL Tracing

You can view the topology of a single call, as well as the overall topology between different services. In some scenarios, the call relationships of an important business need to be traced. This process is called URL tracing. For example, to trace the API for creating online shopping orders. In APM, URL tracing consumes a large number of resources. Therefore, an entry URL will not be added for tracing by default. However, you can set that if necessary. APM has a limit on the total number of URLs added for tracing. It focuses on tracing the downstream calls for the APIs that are added for tracing. Through URL tracing, you can monitor the call relationships between important APIs and downstream services, and then detect problems more precisely.

Configuring URL Tracing

Step 1 Log in to the management console.

Step 2 Click on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click the environment that needs URL tracing. The environment details page is displayed. By default, the **URL** tab page is displayed.

Step 5 Move the mouse pointer to the target URL, click , and add it for URL tracing.

----End

Checking the URL Tracing View

- On the **URL** tab page:

  For the URL added for tracing, click next to it to view its topology.
- On the **URL Tracing Views** tab page:

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click an application. The application details page is displayed.

Step 3 Click the **URL Tracing Views** tab to check all URL tracing views of the application.

Step 4 Filter transaction views by region and environment.

Step 5 Click **View** in the **Operation** column of the row that contains the URL you want to view.

----End

Viewing the URL Tracing Configuration

The URL which has been added for tracing will be displayed in the URL tracing configuration list.

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click an application. The application details page is displayed.

Step 3 Click the **URL Tracing Configuration** tab to check all URL tracing configurations of the application.

Step 4 To delete a URL tracing configuration, click **Delete** in the **Operation** column.

----End

Viewing Transactions

Transaction URLs are displayed in a list. By default, the system displays the invocation of all entries.

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click an application. The application details page is displayed.

Step 3 Click the **Transactions** tab to view all transactions of the application.

Step 4 Click **View the call chain** in the **Operation** column of the target transaction. For operations related to call chains, see **Tracing**.

----End

3.8 Resource Tag Management

You can tag resources under your account for classification. This section describes how to use tags to query resources and how to modify and delete tags.

NOTE

Resource tag management is related to **Tag Management Service**.

Checking Resource Tags

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 5 Checking the tag list of the current application, as shown in the following figure.

----End

Adding Resource Tags

To add a tag with the same key to all resources in the search result list, click **Add Tag**.

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree on the left, click the application to which you want to add a tag and choose **Resource Tag Management** > **Add Tag**.

Step 3 Set tag parameters.

Table 3-28 Tag parameters

Parameter	Description
Tag Key	• The tag key cannot be empty or start or end with a space. • Enter 1 to 128 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. • Each tag value must be unique.
Tag Value	• Enter up to 255 characters. Only letters, digits, spaces, and special characters (_.:=+-@) are allowed. • The resource tag value can be empty, but the predefined tag value cannot be empty.

NOTE

1. Each application supports up to 20 tags.
2. It is recommended that you use the TMS predefined tag function to add the same tag to different cloud resources. For details, see **Creating Predefined Tags**.

Step 4 Click **OK**.

----End

Editing Resource Tags

When you modify a tag, the modification applies only to the cloud resources that contain this tag. To modify a tag, perform the following steps:

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 3 Click **Edit** in the **Operation** column to modify the tag content, as shown in the following figure.

Step 4 Click **OK**.

----End

Deleting Resource Tags

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree on the left, click a target application and click the **Resource Tag Management** tab.

Step 3 Click **Remove** in the **Operation** column to delete the target tag, as shown in the following figure.

Step 4 Click **Yes**.

----End

3.9 Managing Tags

You can add tags for different environments and applications for easy management.

Tag management covers tags and global tags.

A tag is used to set a collector corresponding to the monitoring item under one or more environments of an application.

A global tag is used to set a collector corresponding to the monitoring item under all environments of an application.

NOTE

Priority: Global tag collector configuration > Tag collector configuration > Collector configuration of a monitoring item under an environment

Adding a Tag

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the navigation tree, select a target application.

Step 5 Click the **Tags** tab.

Step 6 Click **Add Tag**.

Step 7 On the page that is displayed, set **Tag** and **Description**, and select the environment to be associated.

Table 3-29 Tag parameters

Parameter	Description
Tag	Enter 1 to 128 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Description	Enter up to 1000 characters. Only digits, letters, underscores (_), hyphens (-), brackets, and periods (.) are allowed.
Bind Environment	• Search by component, environment, or application name is supported. • You can select one or more environments.

Step 8 Click **Yes**.

----End

Modifying a Tag

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Locate the row that contains the tag to be modified and click **Collector Configuration** in the **Operation** column. In the dialog box that is displayed, select your desired collector from the drop-down list and click **Yes**.

Locate the row that contains the tag to be modified and click **Change Environment** in the **Operation** column. In the dialog box that is displayed, select your desired environment and click **Yes**.

Locate the row that contains the tag to be modified and click **Modify Tag** in the **Operation** column. In the dialog box that is displayed, modify the tag and description.

----End

Deleting a Tag

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Locate the row that contains the target tag and click **Delete** in the **Operation** column. Alternatively, select the tags to delete and click **Delete** above the tag list.

Step 5 In the dialog box that is displayed, click **Yes**.

----End

Global Tag Collector Configuration

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the navigation tree, select a target application.

Step 3 Click the **Tags** tab.

Step 4 Click **Global tag collector configuration**.

Step 5 Select a collector from the drop-down list and click **Yes**. For details about how to configure monitoring items, see **Application Monitoring Configuration**.

----End

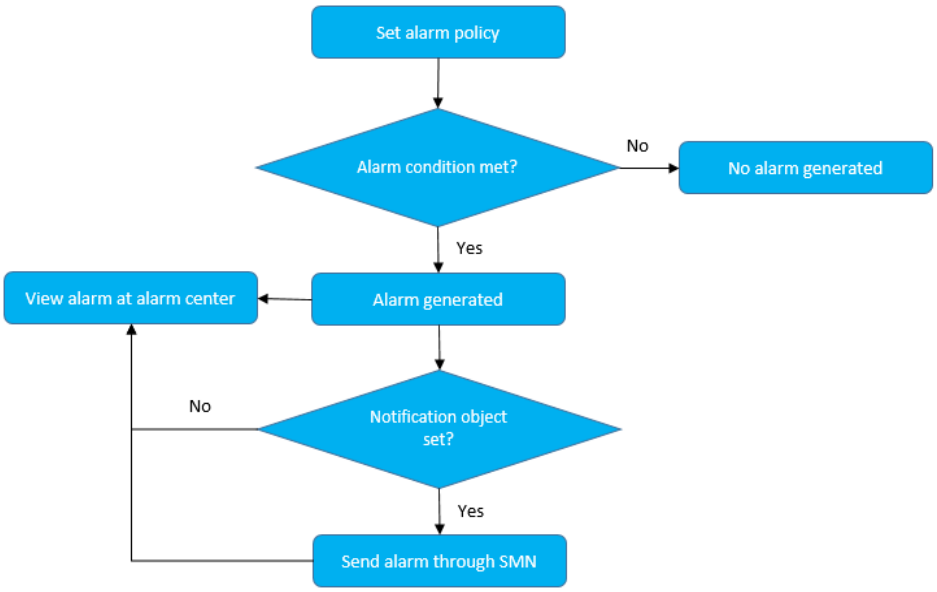
3.10 Alarm Management

3.10.1 Alarm List

Alarms are reported by services connected to APM Agents when specified conditions are met. You can learn about service exceptions in a timely manner and quickly rectify faults to prevent service loss.

Alarm process

Figure 3-12 Alarm process



Viewing Alarms

Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Alarm Center** > **Alarm List**.

Step 4 View alarms on the **Alarm List** page.

1. Select an application from the application drop-down list to view its alarms.
2. In the search text box, set search criteria, and click 🔍 to view the alarms that meet the criteria.
3. Click ▽ next to **Alarm Status** to filter alarms by alarm status.
4. When necessary, enable **Scheduled Refresh**. In that case, the alarm list is refreshed every 5s.
5. Click ⚙ in the **Operation** column to view the alarm details and notification.

----End

3.10.2 Alarm Policies

3.10.2.1 Configuring an Alarm Template

APM allows you to configure alarm templates. You can create multiple alarm policies under a template and bind them to nodes.

Procedure

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click an application. The metric details page of the application is displayed.

Step 5 Click the **Alarm Templates** tab.

Step 6 Click **Add Template** to add an alarm template as prompted.

1. Enter basic information and then click **Next**.

   Template Name: Enter up to 64 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

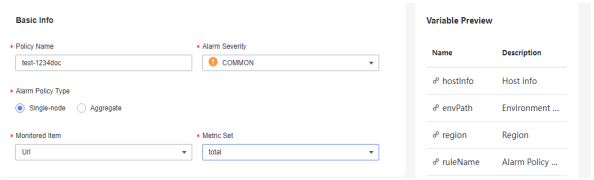
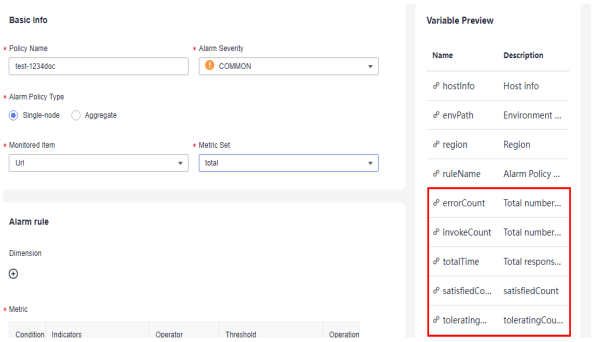
   Remarks: Enter up to 512 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed.

2. Click **Add Alarm Policy** to add an alarm policy.

   a. Basic information

   Table 3-30 Basic information about an alarm policy

Parameter	Description
Policy Name	Custom name, which cannot be left blank. Only letters, digits, underscores (_), and hyphens (-) are allowed. Enter up to 512 characters.
Alarm Severity	Severity of an alarm. Options: **COMMON** and **CRITICAL**.
Alarm Policy Type	Options: **Single-node** and **Aggregate**. **Single-node** indicates single-instance metric alarms, and **Aggregate** indicates aggregated metric alarms of all instances under a component.

Parameter	Description
Monitoring Item	Select a target monitoring item. The information about the selected item is displayed on the right. 
Metric Set	Select a target metric set. The information about the selected metric set is displayed on the right. 

b. Alarm rule

Figure 3-13 Alarm rule

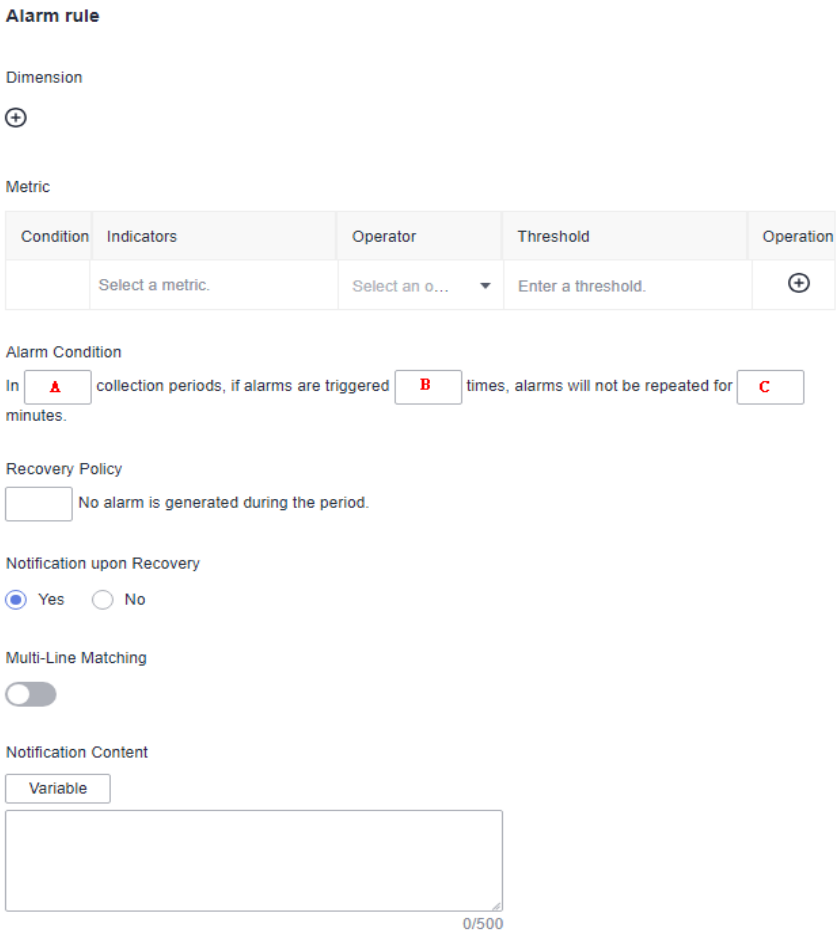


Table 3-31 Alarm rule parameters

Parameter	Description
Dimension	(Optional) A category of metrics.
Metric	Metric for which you want to define one or more alarm rules. **Metric**: a metric in the metric set. For example, if **Monitoring Item** is set to **Url** and **Metric Set** is set to **total**, you can select the **errorCount** metric. **Operator**: operation to be performed. **Threshold**: threshold of the metric.

Parameter	Description
Alarm Condition	Condition for triggering an alarm. A: the number of collection periods. Range: 1–10. B: the number of times the alarm is triggered. Range: 1–10. This value cannot be greater than that of A. C: period (in minutes) during which identical alarms will not be sent. This period cannot be shorter than 10 minutes.
Recovery Policy	Condition for clearing an alarm. For example, if this parameter is set to **3**, the alarm status will change to "Cleared" if no alarm is generated within 3 minutes.
Notification upon Recovery	Whether to notify recipients of alarm clearance.
Multi-Line Matching	(Optional) Whether to define data in the alarm notification content line by line.
Notification Content	Alarm details, which contain up to 500 characters. ■ If **Multi-Line Matching** is enabled, the alarm notification content supports both **Variable** and **Loop**. If **Multi-Line Matching** is disabled, only **Variable** can be selected. ■ Alarm notification content. You can customize the content or select metrics as required. ■ Alarm details, which contain up to 500 characters. ■ Select required metrics. Specifically, on the right of the page, click next to the target metric. The metric will then be displayed in the notification content. 

c. Notification object

Table 3-32 Alarm notification parameters

Parameter	Description
Use default settings	– If this option is enabled, alarms will be sent to one or more default notification objects. For details about how to set notification objects, see **Alarm Notification**. – After this option is enabled, you can no longer select notification objects from the **Notification Object** drop-down list. – If all the values in the **Default** column of the **Notifications** page are **No**, no default notification objects have been set. In this case, the **Use default settings** option on the alarm policy creation page is dimmed and cannot be enabled.
Notification Object	Select a notification object from the drop-down list. Alarms will only be sent to the selected notification objects.

3. Click **Yes**.
4. Click **Next**. The **Bind Node** page is displayed.
5. Click **Bind Node** to bind nodes by environment, environment tag, or region.

Table 3-33 Node parameters

Parameter	Description
All	All nodes (including those added subsequently) in all regions will be bound.
Environment	All nodes in the selected environment will be bound.
Environment Tag	All nodes with the same tag will be bound.
Region	All nodes in the selected region will be bound.

Step 7 Click **Yes**. The information about the bound nodes is displayed in the lower part of the page, including the environment, tag, and region names.

NOTE

If there are multiple types of bound nodes, only information about the nodes with the same environment, tag, and region is displayed.

Step 8 Click **Complete**.

----End

More Operations

After the alarm template is created, perform the operations listed in **Table 3-34** if needed.

Table 3-34 Related operations

Operation	Description
Copying a template	Click **Copy** in the **Operation** column in the row that contains the template you want to copy.
Modifying a template	Click **Modify** in the **Operation** column in the row that contains the template you want to modify.
Deleting a template	Click **Delete** in the **Operation** column in the row that contains the template you want to delete.
Starting and stopping a template	Turn on or off the button ( ) in the **Operation** column in the row that contains the template you want to start or stop.

3.10.2.2 Creating a Custom Alarm Policy

You can create a custom alarm policy for a single component.

Procedure

Step 1 Log in to the management console.

Step 2 Click on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click next to the target environment. The instance monitoring page is displayed.

Step 5 Click the **Alarm Policy** tab.

Step 6 Click **Add Custom Alarm Policy** and set the alarm condition in the same way as that when you create an alarm template.

----End

Create an Alarm Policy Based on a Template

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 In the tree on the left, click next to the target environment. The instance monitoring page is displayed.

Step 3 Click the **Alarm Policy** tab.

Step 4 In the template list, click **Copy** in the **Operation** column in the row that contains the template you want to copy.

----End

More Operations

After the alarm policy is created, perform the operations listed in **Table 3-35** if needed.

Table 3-35 Related operations

Operation	Description
Starting or stopping a policy	In the custom alarm policy list, start ( in the **Operation** column) or stop the target policy.
Modifying a policy	Click **Edit** in the **Operation** column in the row that contains the policy you want to modify.
Deleting a policy	Click **Delete** in the **Operation** column in the row that contains the policy you want to delete.

3.10.2.3 Recommended Alarm Templates

APM provides recommended alarm templates.

Using Recommended Alarm Templates

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click an application. The metric details page of the application is displayed.

Step 5 Choose **Alarm Templates** > **Recommendation Template** to view the configured alarm templates.

Step 6 Click **View Details** in the **Operation** column in the row that contains the target alarm template.

Step 7 Click **Copy** to copy the recommended template to the template list. You can customize the template name as required.

Step 8 Click **Yes**. The copied alarm template is displayed on the template list.

Step 9 Click **Modify** in the **Operation** column and **bind nodes** to make the copied template to take effect.

----End

3.10.3 Alarm Notification

Alarms can be sent to specified terminals by SMS message, email, or function. In this way, you can obtain component exceptions in a timely manner and quickly rectify faults to avoid service loss. Ensure that you have the SMN permission. For details, see Simple Message Notification (SMN) User Guide.

If you do not create any notification object, no alarm notifications will be received. To view alarms, log in to the APM console and choose **Alarm Center** > **Alarm List** in the navigation pane.

Creating a Notification Object

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 4 In the tree on the left, click an application. The metric details page of the application is displayed.

Step 5 Click the **Notifications** tab.

Step 6 Click **Add**.

Step 7 On the displayed page, specify **Region** and **Topic**, and determine whether to enable default notification. If it is enabled, alarm notifications will be sent based on the topic and region you specify.

- If no topic is available, create one.
- If default notification is enabled, alarms will be sent to the specified region when you create an alarm policy.

Step 8 Click **Yes**.

----End

3.11 Agent Management

3.11.1 Introduction

APM Agents use bytecode enhancement technology to collect application performance data in real time. They run on the server where applications are deployed. Install Agents before using APM. For details, see section "Manually Installing Agents for Java Applications" in *APM Getting Started*.

3.11.2 Operating Agents

Agent Management allows you to check the deployment and running statuses of the Agents that are connected to APM, and to stop, start, or delete them.

Checking Agents

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Application Monitoring** > **Agent Management**.

Step 4 Check the Agent list.

1. In the upper left corner of the page, select a target region and application.
2. Set the search criteria and click  in the search box in the upper right corner of the page to filter Agents.

----End

The following table describes the Agent statuses.

Status	Description
Enabled	The Agent is running properly.
Offline	The Agent is abnormal due to a network error. Check and restore the network.
Disabled	The Agent is manually or globally disabled. Contact technical support.

Batch Operations

Step 1 In the navigation pane, choose **Application Monitoring** > **Agent Management**.

Step 2 Select target objects and click **Operation**.

Step 3 Select **Disable Agent**, **Enable Agent**, or **Delete Agent**.

Step 4 In the dialog box that is displayed, click **Yes** to disable, enable, or delete the Agents for the selected hosts.

----End

3.11.3 Upgrading Agents

Update Agent versions according to the following procedure.

Upgrading the Manually Installed Agents

To upgrade the manually installed Agent, download the new Agent by referring to section "JavaAgent Download Addresses" in *APM Getting Started*section "JavaAgent Download Addresses". For details, see section "Manually Installing Agents for Java Applications" in *APM Getting Started*.

Upgrading the Agents for Java Applications Deployed in CCE Containers

To upgrade the Agents for the Java applications deployed in CCE containers, select the new version for installation. For details, see section "Installing Agents for the Java Applications Deployed in CCE Containers" in *APM Getting Started*.

Upgrading Agents of Other Types

Install new Agents. For details, see section "Manually Installing Agents for Java Applications" in *APM Getting Started*.

3.12 Configuration Management

3.12.1 Collection Center

Collection Center displays collectors in a centralized manner. You can view and manage various collectors, metrics, and collection parameters supported by APM.

Viewing Collector Details

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **Configuration Management** > **Collection Center**.

All the supported collectors are displayed.

Step 4 In the collector list, click **View Details** in the **Operation** column in the row that contains the target collector. The collector details page is displayed.

Step 5 The collector details page consists of three modules: basic information, collection parameters, and metric set.

- Basic information

  This module displays collector information such as collector name and type.

- Collection parameters

  This module displays the custom parameter settings supported by the collector. The settings take effect after being delivered to JavaAgents and are used for custom collection.

- Metric sets

  This module displays information about the metrics collected by the collector.

----End

Collector

A collector is a plug-in for collecting metric data. It consists of the collector description, metric set, and collection parameters. Collector description describes the data collected by a collector. Metric set is the data collected according to specifications. Collection parameters are the custom data to be collected.

- Data is collected by APM Agents. For example, Java performance data is collected by JavaAgents. The data collected by APM Agents must correspond to the data models of collectors' metric sets so that servers can process the data.
- The Agent of each language and framework defines its own collector.
- After a collector is added to an environment, it is instantiated as a monitoring item. This process is generally automated. APM Agents automatically discover collection plug-ins used by applications and add collectors to the environment to form monitoring items. For example, if a Java application connects to a database through the JDBC driver for MySQL, the MySQL collector is automatically added to the environment to form a monitoring item.

Collection Parameters

Collectors corresponding to monitoring items define collection parameters. You can modify collection parameters on the page as required. These parameters will be delivered to Agents with heartbeat parameters to change collection behaviors. By default, Redis instruction content is not collected for security purposes. If necessary, modify collection parameters to collect specific instruction data. Collection parameters can also be defined on environment tags. Collectors automatically inherit collection parameter attributes of corresponding environment tags. In this way, configuration is automated. For details about how to set collection parameters, see **Application Monitoring Configuration**.

Metric Sets

A collector collects data of multiple metric sets. For example, the URL collector collects URL details, overall call condition, and status statistics. Each type of statistics corresponds to a metric set. Each metric set contains multiple metrics. For example, the metric set of URL details contains metrics such as the URL, method, number of calls, number of errors, and slowest call. Each metric corresponds to a data type.

APM supports the following types of metric data:

Table 3-36 APM metric data types

Data Type	Description	Remarks
ENUM	Enumeration	Primary key type. In the example of URL monitoring, the URL and method metrics are primary keys, and other metrics such as the number of calls correspond to the URL and method.
INT	Integer	Maximum size: 8 bytes
DOUBLE	Floating-point number	8-byte floating-point number

Data Type	Description	Remarks
STRING	Character string	Maximum length: 1,024 characters
CLOB	Large character string	Maximum size: 1 MB
DATETIME	Time	Time is automatically displayed on the page.

3.12.2 Data Masking

You can set policies to mask the data reported using APM 2.0.

NOTICE

APM will collect and store masked data. Do not upload privacy or sensitive data to APM. If you need to upload such data, encrypt it.

Querying a Data Masking Configuration

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation tree on the left, choose **Configuration Management** > **Data Masking** and select your target node. The configuration information is displayed.

Step 4 In the search box, enter a configuration name keyword and click the search icon or press **Enter**.

----End

Adding a Data Masking Configuration

Step 1 In the navigation tree on the left, choose **Configuration Management** > **Data Masking** and select your target node.

Step 2 Click **Add** and set configuration parameters.

Table 3-37 Configuration parameters

Parameter	Description
Configuration Name	Used to identify a data masking configuration. This parameter cannot be empty. Enter up to 30 characters. Only letters, digits, and special characters are allowed.
Configuration Description	Used to describe the data masking configuration. This parameter cannot be empty. Enter up to 1000 characters. Only letters, digits, and special characters are allowed.
Configuration Items	• Enter up to 32 characters. Only letters, digits, underscores (_), and hyphens (-) are allowed. • The configuration item cannot be empty. By default, an empty configuration item is displayed. If you select **Token**, content will be replaced with a globally unique random character string. If you select **Mask**, content will be replaced with a fixed number of asterisks (*). By default, **Mask** is selected. • Click the plus sign (+) to add a configuration item, or click the minus sign (–) to delete one. • Each configuration can contain up to 20 configuration items. • The **httpMethod**, **remoteAddr**, **exceptionType**, **content-type**, **charset**, **api_address**, **url**, **method**, **requestBody**, **responseBody**, **exceptionMsg**, **cookie**, and **Cookie** fields have special functions in APM traces and do not support masking. • If you use one of these fields as a key, the system will display a message indicating that an invalid name exists.

Step 3 Click **Yes**.

----End

Modifying a Data Masking Configuration

Step 1 In the navigation tree on the left, choose **Configuration Management** > **Data Masking** and select your target node.

Step 2 Click **Modify** in the **Operation** column to modify the configuration.

Step 3 Click **Yes**.

----End

Deleting Data Masking Configurations

Step 1 In the navigation tree on the left, choose **Configuration Management** > **Data Masking** and select your target node.

Step 2 Click **Delete** in the **Operation** column. In the displayed dialog box, click **Yes** to delete the configuration.

Step 3 Select multiple data masking configurations and click **Batch Delete** above the list. In the displayed dialog box, click **Yes** to delete multiple data masking configurations at a time.

----End

3.13 System Management

3.13.1 Access Keys

Access Key ID (AK) and Secret Access Key (SK) are your long-term identity credentials. JavaAgents report data with an AK. An AK is used together with an SK to sign requests cryptographically, ensuring that the requests are secret, complete, and correct.

Precautions

A user can create a maximum of two access keys with identical permissions and unlimited validity. Keep your access keys secure and change them periodically for security purposes. To change an access key, delete it and create a new one.

NOTE

APM allows you to encrypt and decrypt the SK in the **apm.config** file.

The encryption and decryption process is as follows:

1. Compile a Java class, for example, **com.demo.DecryptDemo**, and add a decryption method, for example, decrypt both the input and output to character strings.
2. Compile the decryption method to decrypt the SK and return the decrypted value.
3. Pack the **com.demo.DecryptDemo** class into a JAR package and place this JAR package and its dependent packages in the **apm-javaagent/ext** folder of JavaAgent.
4. Add the following content to the **apm.config** file:

   decrypt.className=com.demo.DecryptDemo

   decrypt.methodName=decrypt

   secret.key={*Character string encrypted by users***}**

Adding an Access Key

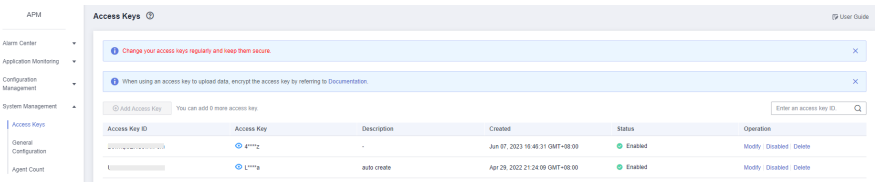
Step 1 Log in to the management console.

Step 2 Click ☰ on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **System Management** > **Access Keys**.

Step 4 On the page that is displayed, click **Add Access Key**.

Figure 3-14 Adding an AK/SK



Step 5 Add an access key description and click **Yes** to generate an access key.

To modify the description, click **Modify** in the **Operation** column in the row that contains the target access key.

----End

Deleting an Access Key

Step 1 In the navigation pane, choose **System Management** > **Access Keys**.

Step 2 On the **Access Keys** page, locate the row that contains the target access key and click **Delete** in the **Operation** column.

Step 3 On the page that is displayed, click **Yes** to delete the access key.

----End

Enabling or Disabling an Access Key

Each access key is enabled by default. To disable it, do as follows:

Step 1 In the navigation pane, choose **System Management** > **Access Keys**.

Step 2 On the **Access Keys** page, locate the row that contains the target access key and click **Disable** in the **Operation** column.

Step 3 On the page that is displayed, click **Yes** to disable the access key.

To enable it again, click **Enable** in the row that contains the access key. On the page that is displayed, click **Yes**.

----End

3.13.2 General Configuration

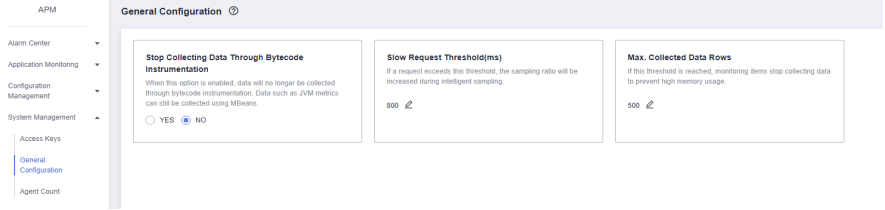
On the **General Configuration** page, you can determine whether to collect data through bytecode instrumentation, and specify the slow request threshold and maximum number of rows to collect.

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation pane, choose **System Management** > **General Configuration**.

Figure 3-15 Modifying general configuration



- Stop Collecting Data Through Bytecode Instrumentation

  Enable or disable this function as required. Data such as JVM metrics will always be collected using MBeans. The default value is **No**.

  NOTE

  When the **Stop Collecting Data Through Bytecode Instrumentation** option is enabled, data will no longer be collected through bytecode instrumentation. Data such as JVM, GC, and Tomcat thread metrics can still be collected using MBeans.

- Slow Request Threshold

  If this threshold is reached, more samples will be collected during intelligent sampling. The default value is **800**.

- Max. Collected Data Rows

  If this value is reached, data will not be collected to prevent excessive memory usage. The default value is **499**.

----End

3.13.3 Agent Count

APM can count the Agents used by tenants. You can view the number of Agents by time, region, or Agent type.

Step 1 Log in to the management console.

Step 2 Click  on the left and choose **Management & Deployment** > **Application Performance Management**.

Step 3 In the navigation tree, choose **System Management** > **Agent Count**.

- **Current Agent**: number of Agents used by the current tenant.
- **Historical Agent**: number of Agents used in each hour of today, yesterday, or a custom day.

----End

Checking the Number of Agents Used by an Application

Step 1 In the navigation pane, choose **Application Monitoring** > **Metrics**.

Step 2 On the displayed page, select an application to view. The **Application Info** tab page is displayed by default.

Step 3 Switch to the **Agent Count** tab page to view the number of Agents used by the current application.

- **Current Agent**: number of Agents used by the current application.
- **Historical Agent**: number of Agents used in each hour of today, yesterday, or a custom day.

----End

3.14 Permissions Management

3.14.1 Authorizing Users and User Groups Using Enterprise Projects

Enterprise Project Management Service (EPS) is used to control user access to APM resources. After creating IAM user groups for employees, you can create enterprise projects on the Enterprise Management console and grant permissions to the user groups in the enterprise projects to implement personnel authorization and permission control. You can create enterprise projects. Then you can manage resources across different regions by enterprise project, grant different permissions to user groups, and add them to enterprise projects.

3.14.2 Creating a User and Granting Permissions

This chapter describes how to use IAM for fine-grained permissions control for your APM resources. With IAM, you can:

- Create IAM users for employees based on your enterprise's organizational structure. Each IAM user will have their own security credentials for accessing APM resources.
- Manage permissions on a principle of least permissions (PoLP) basis.
- Entrust an account or cloud service to perform efficient O&M on your APM resources.

If your account does not need individual IAM users, skip this chapter.

This section describes the procedure for granting permissions (see **Figure 3-16**).

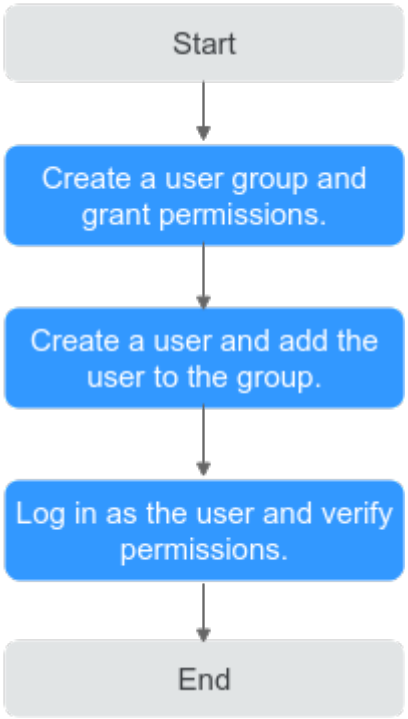
Prerequisite

Learn about the permissions supported by APM and choose policies or roles based on your requirements. For details, see **Permissions Management**.

Process Flow

Supported Cloud Services

Figure 3-16 Process for granting APM permissions



1. **Creating a User Group and Assigning Permissions**

   Create a user group on the IAM console, and assign the **APM ReadOnlyAccess** policy to the group.

2. **Creating an IAM User**

   Create a user on the IAM console and add the user to the group created in **1**.

3. **Logging In as an IAM User** and Verifying Permissions

   Log in to the APM console using the created user, and verify that the user only has read permissions for APM.

4 FAQs

4.1 Are APM Agents Compatible with Other Agents Such as Pinpoint?

APM Agents are incompatible with other Agents. Generally, APM implements bytecode instrumentation based on the ASM framework. Installing two Agents means two instrumentation operations on your code. However, code instrumentation mechanisms vary according to products. If you install Agents of different products, code conflicts may occur, affecting performance.

4.2 What Is APM's Metric Data Sampling Policy?

Metric data is collected periodically. The default collection period is 1 minute.

4.3 Why Does Metric Data Collection Fail?

1. You can view metric data several minutes after you connect Agents.
2. If data collection is stopped, the possible causes are as follows:
   - Instance level: Agents are stopped on the **Instance** tab page.
   - Monitoring item level: Monitoring items are manually disabled on the **Monitoring Item** tab page.
   - Global level: The **Stop Collecting Data Through Bytecode Instrumentation** option is enabled on the **General Configuration** page.
3. If no data is collected for a long time, the possible causes are as follows:
   - Java 9 prompts that the **sql.time** class cannot be found.

     Cause analysis: Agents are developed using JDK 1.7. However, after Java 9 modularization, no SQL package is provided by default.

     Occurrence probability: This problem occurs under certain conditions.

     Workaround: Ensure that the component can proactively import **java.sql** to **module-info.java**.

- Java 11 prompts that "Caused by: java.lang.NoClassDefFoundError: sun/misc/Unsafe class cannot be found."

  Cause analysis: Agents are developed using JDK 1.7, but the Java 11 Unsafe class is categorized to a different package.

  Occurrence probability: This problem occurs inevitably.

  Workaround: Ensure that the application can proactively import **jdk.unsupported** to **module-info.java**.
- Java 9 reports an illegal reflective access alarm. (This problem will be solved in versions later than Java 9.)

  Workaround: Set **--illegal-access** to **warn** or delete this option.

4.4 Why Is There No Monitoring Data Displayed on APM After the JavaAgent Is Enabled on CCE?

This is because the JavaAgent is of an old version or started using Tomcat.

To solve the problem, enable APM for free (10 Agents available) or upgrade to the enterprise edition. Use the JavaAgent of the latest version and then restart the container.

4.5 Why Is an AOM Trace Not Displayed on the APM Console?

Application Operations Management (AOM) traces will be displayed only when there are users accessing it. Also, it generally takes several minutes to display the traces. If AOM is not accessed by any user or there is no data, no traces will be displayed on APM.

5 Change History

Table 5-1 Change history

Date	Description
2024-07-04	This issue is the first official release.

