Alibaba Cloud Application Real-time Monitoring Service

Product overview

Issue: 20200616

MORE THAN JUST CLOUD | Alibaba Cloud

Legal disclaimer

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

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.
2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company, or individual in any form or by any means without the prior written consent of Alibaba Cloud.
3. The content of this document may be changed due to product version upgrades, adjustments, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and the updated versions of this document will be occasionally released through Alibaba Cloud-authorized channels. You shall pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.
4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides the document in the context that Alibaba Cloud products and services are provided on an "as is", "with all faults" and "as available" basis. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not bear any liability for any errors or financial losses incurred by any organizations, companies, or individuals arising from their download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, bear responsibility for any indirect, consequential, exemplary, incidental, special, or punitive damages, including lost profits arising from the use or trust in this document, even if Alibaba Cloud has been notified of the possibility of such a loss.

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

Document conventions

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

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

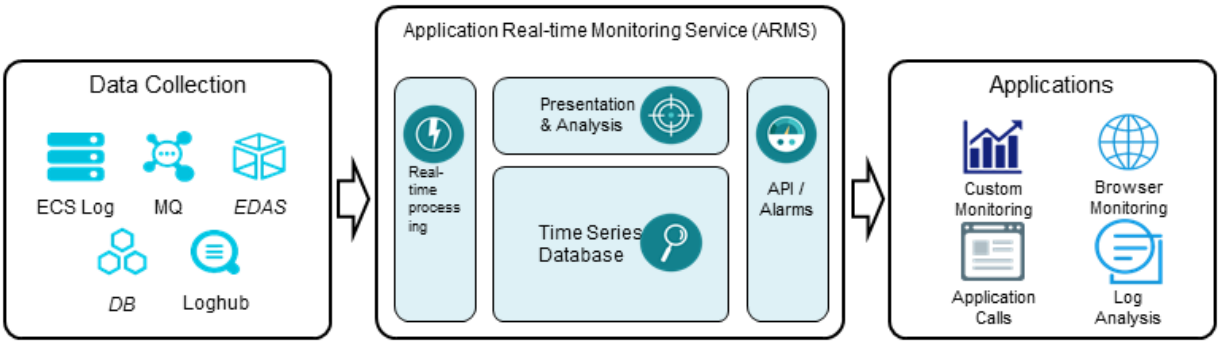
Contents

1 Overview

Application Real-Time Monitoring Service (ARMS) is an application performance management (APM) product of Alibaba Cloud. With ARMS, you can quickly and conveniently build business monitoring capabilities with few-second response time for businesses and enterprises based on custom dimensions such as the browser, application, and business.

Workflow

The following figure shows the ARMS workflow.

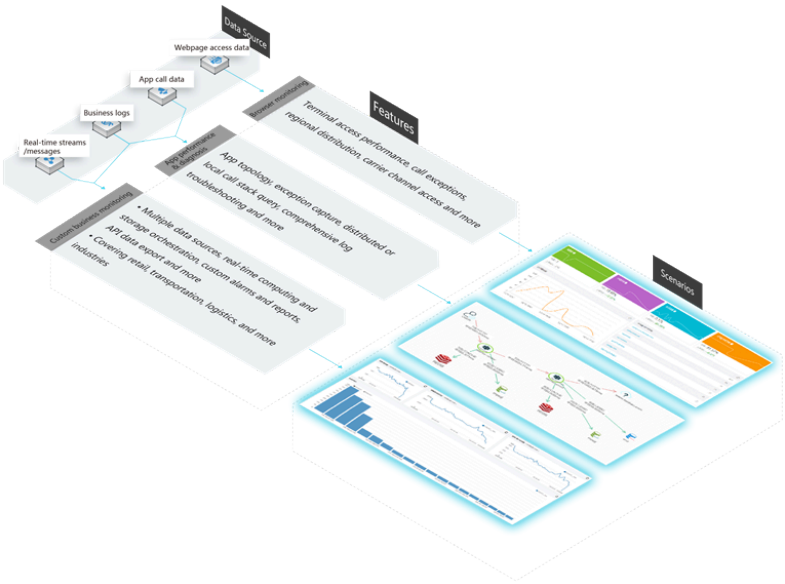


- Data collection: ARMS supports capturing logs from Elastic Compute Service (ECS) instances, Message Queue (MQ), and LogHub through configuration.
- Job definition:
  - ARMS allows you to define jobs such as real-time processing, data storage, presentation and analysis, data API, and alerts through job configuration, to define your own application scenarios.
  - ARMS directly performs business monitoring with preset scenarios, such as browser monitoring and application monitoring.
- Application scenario: In addition to custom monitoring, ARMS also provides ready-to-use preset monitoring scenarios, including browser monitoring and application monitoring.

Scenarios

- **Highly tailored business monitoring**: You can create real-time monitoring alerts and dashboards based on business characteristics and requirements. Business scenarios include the e-commerce scenario, logistics scenario, and airlines and tour scenarios.
- **Browser experience monitoring**: ARMS can show the performance and errors of pages you visited by region, channel, link, or other dimensions.

- **Application performance and exception monitoring**: ARMS provides the APM capabilities to monitor performance exceptions and query traces for distributed applications.
- **Central alert and report platform**: Custom monitoring, browser monitoring, and application monitoring are integrated on a central alert and report platform.



With ARMS, IT engineers can build and start a big data platform-based real-time applicatio n monitoring system within minutes, which maximizes the timeliness of data monitoring and improves the efficiency of IT engineers.

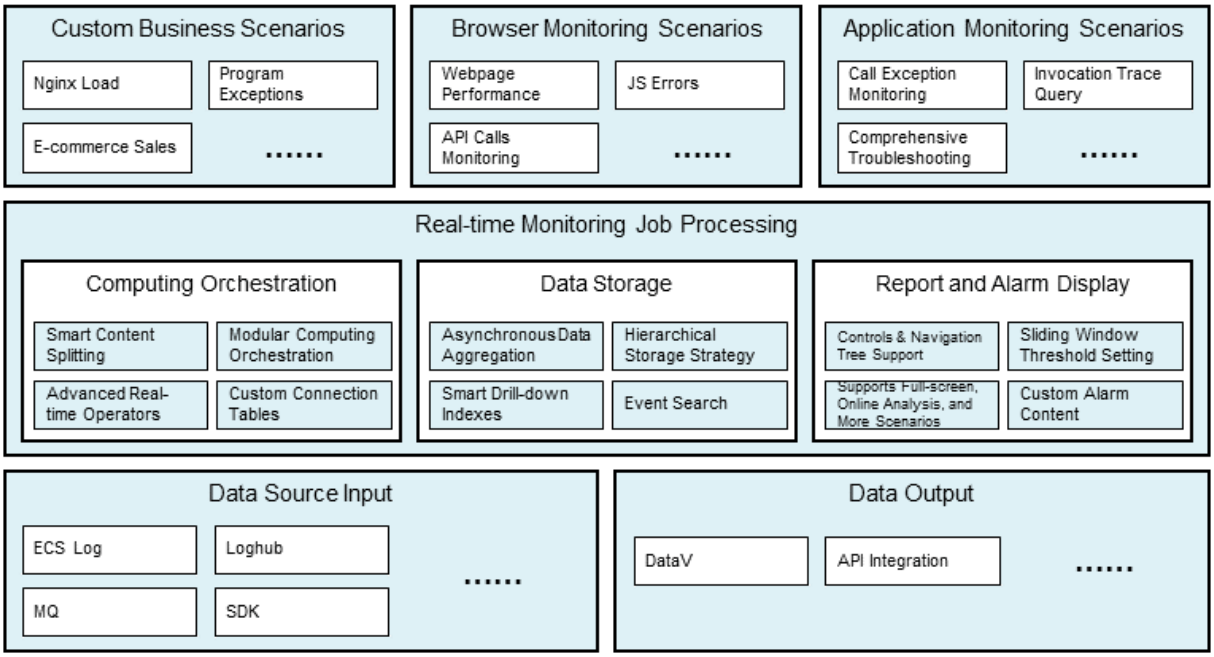
Learning path

You can use ARMS learning path to learn how to use basic functions of ARMS, such as application monitoring, browser monitoring, custom monitoring, dashboard, and alerts, and how to use rich API operations and SDKs to meet your specific needs.

2 Function overview

Application Real-Time Monitoring Service (ARMS) provides a series of custom monitoring functions, including data access, data computing, data storage, dashboard presentation, and alerts. It also can call and be called by APIs of downstream services.

The overall functions are shown in the following figure.



The functions are detailed as follows:

- Multi-dimensional browser monitoring
  - High timeliness: Detects the response time and error rate of websites accessed by users in real time.
  - Multidimensional monitoring and analysis: Analyzes user access speeds and errors by multiple dimensions, such as the region, operator, browser.
  - Page exception monitoring: Monitors and diagnoses the performance and success rate of a large number of asynchronous data calls of applications.

- Efficient and easy-to-use application monitoring
  - Application topology self-discovery: Automatically generates the call relationship graph of distributed applications based on dynamic analysis and intelligent computing of the RPC information.
  - Metrics analysis in common diagnosis scenarios: Analyzes metrics such as the application response time, request count, and error rate, and displays the analysis data by application, transaction, or database.
  - Capturing of abnormal and slow transactions: Analyzes timeout and exceptions based on traces, and effectively associates the transactions with API calls, for example, with specific SQL statement and message queue.
  - Transaction snapshot query: Intelligently collects trace-based problem transactions and identifies the sources of exceptions or errors by checking the detailed data.
- Powerful custom monitoring
  - Rich data sources: Supports various types of real-time data sources, such as logs, SDK, Message Queue (MQ), and LogHub.
  - Flexible real-time computing and storage orchestration: Allows you to orchestrate real-time computing and storage modes based on the specified dimension and computing mode.
  - Flexible integration with alerts or dashboards: Quickly integrates monitoring datasets with ARMS alerts and dashboards, to provide monitoring capabilities in various scenarios.
  - Reference scenario templates: Provides a large number of reference scenario templates, such as NGINX monitoring, exception monitoring, and e-commerce monitoring.

- Other basic functions
  - Flexible definition of real-time computing jobs
    - Supports drag-and-drop modular programming of real-time computing and most language logic, such as general mathematical operations, regular expressions, and if and else functions.
    - Supports a variety of real-time computing and storage operators, such as Sum, Count, Max, Min, Sample, TopN, and Count Distinct.
  - Stable and efficient time series and event storage
    - Aggregates data online continuously to ensure a controllable data capacity.
    - Supports intelligent hierarchical storage policies.
    - Supports up to three levels of drill-down indexes.
  - Custom alert settings
    - Supports alerts of moving average and maximum values in any continuous time period.
    - Supports custom alert content.
    - Provides multiple alert notification channels, such as email, SMS, and DingTalk.
  - Flexible customization of the interactive dashboard
    - Provides rich presentation widgets, such as bar charts, heat maps, pie charts, and flap display.
    - Supports dashboard sharing and full screen display.
  - Flexible integration with downstream applications
    - Supports integration with APIs of Java, Python, Perl, and C# applications.
    - Supports integration with other dashboard presentation tools, such as DataV.

3 Use cases

3.1 Java application monitoring and diagnosis solution

In this use case, the application monitoring solution based on Application Real-Time Monitoring Service (ARMS) is adopted to resolve pain points in monitoring distributed Java applications.

Rapid growth of Internet businesses has brought about increasing pressure on traffic, and business logic has also become increasingly complicated. In this background, traditional single-machine applications can no longer satisfy customer needs. The distributed deployment architecture has been adopted by more and more websites. The basic development frameworks, such as Spring Cloud and Dubbo, have gradually become mature. More enterprises vertically split their website architectures by business module and adopt the microservice architecture, which is more suitable for collaborative development among teams and quick iterations.

The microservice-based distributed architecture is advanced in terms of development efficiency. However, it creates huge challenges for traditional monitoring, O&M, and diagnosis technologies. For example, we encountered the following challenges during the application of the microservice-based distributed architecture to www.taobao.com:

- Difficult to troubleshoot

  The customer service center submitted customer feedback to the technical support engineers for troubleshooting about problems with buying items. A website request in the microservice-based distributed architecture always passes through multiple services and nodes for the result. Once an error occurs, the engineers usually have to go through the logs over and over to identify the preliminary issue. Multiple teams were often involved in troubleshooting a simple problem.
- Difficult to find out the bottleneck

  When a customer reports that a website gets stuck, it is difficult to quickly find out the bottleneck. Is the network between the user terminal and the server at fault? Is it a result of server overloading or high database pressure? Even though the cause is identified, it is still difficult to quickly identify the error in the code.

- Difficult to get a clear picture of the architecture

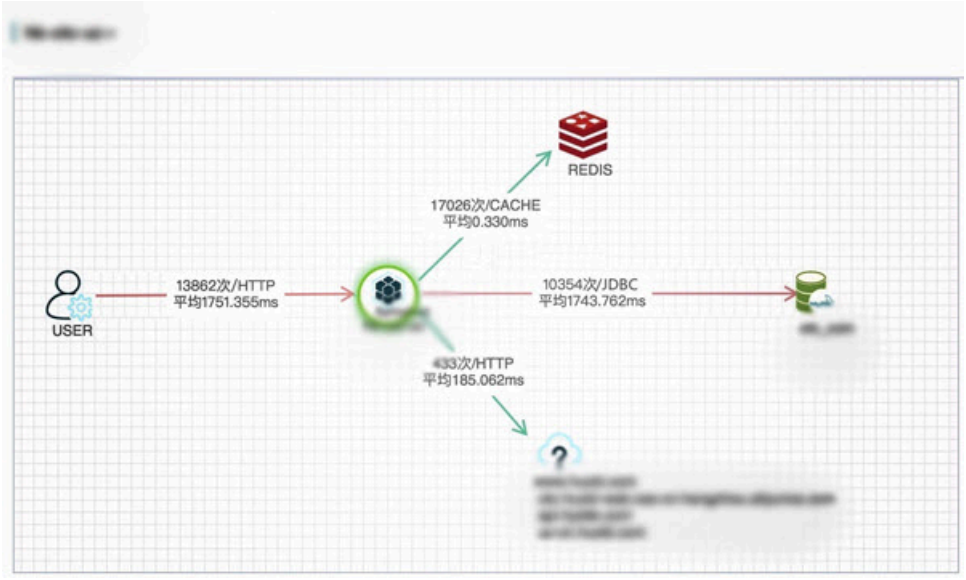
  The business logic has become more complicated. It is difficult to sort out the code that specifies the depending downstream services of an application, for example, the database, HTTP API, or cache, and to sort out the code that specifies the external calls depending on this application. It is more difficult to sort the business logic, manage the architecture, and plan the capacity. For example, during the preparations for "Double 11 " promotion campaigns, the number of servers required for each application is hard to predict.

ARMS-based application monitoring solution

The ARMS **application monitoring** function is originated from the distributed tracing and monitoring system, Alibaba EagleEye. It solves the preceding problems without touching the existing code.

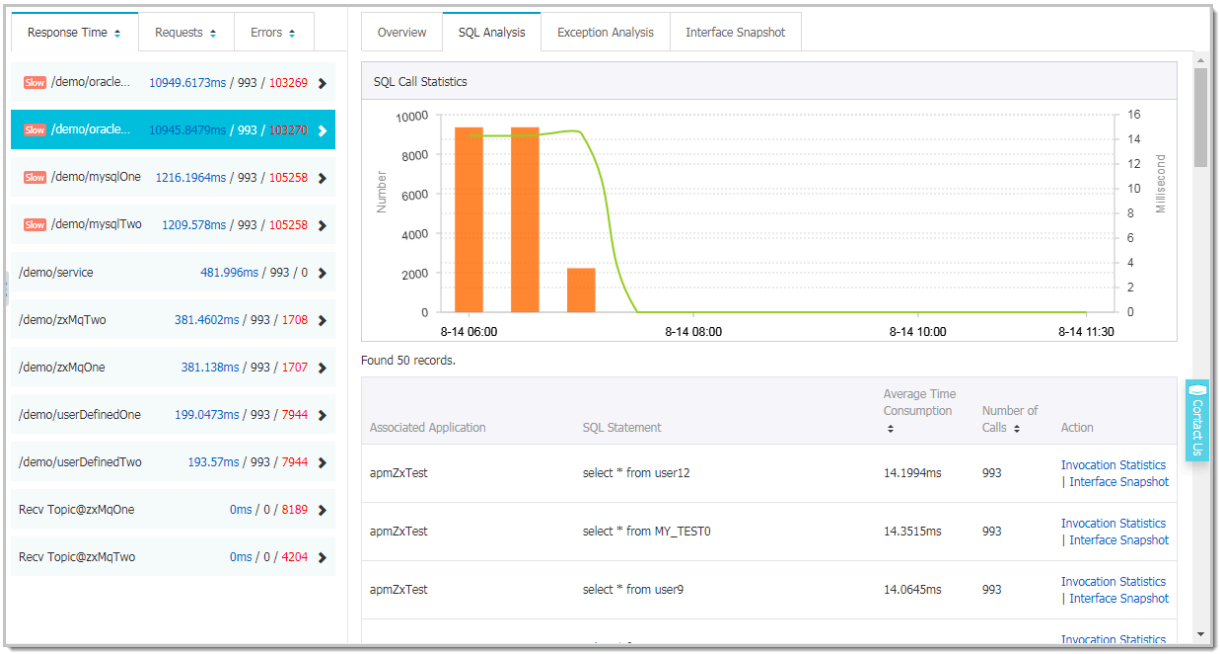
View the call topology

You can view the call topology of an application on ARMS, for example, services that depend on the application and downstream services that the application depends on. As shown in the figure, a bottleneck occurs when an unknown application calls the monitored application, with an average consumed time more than 3,000 ms.



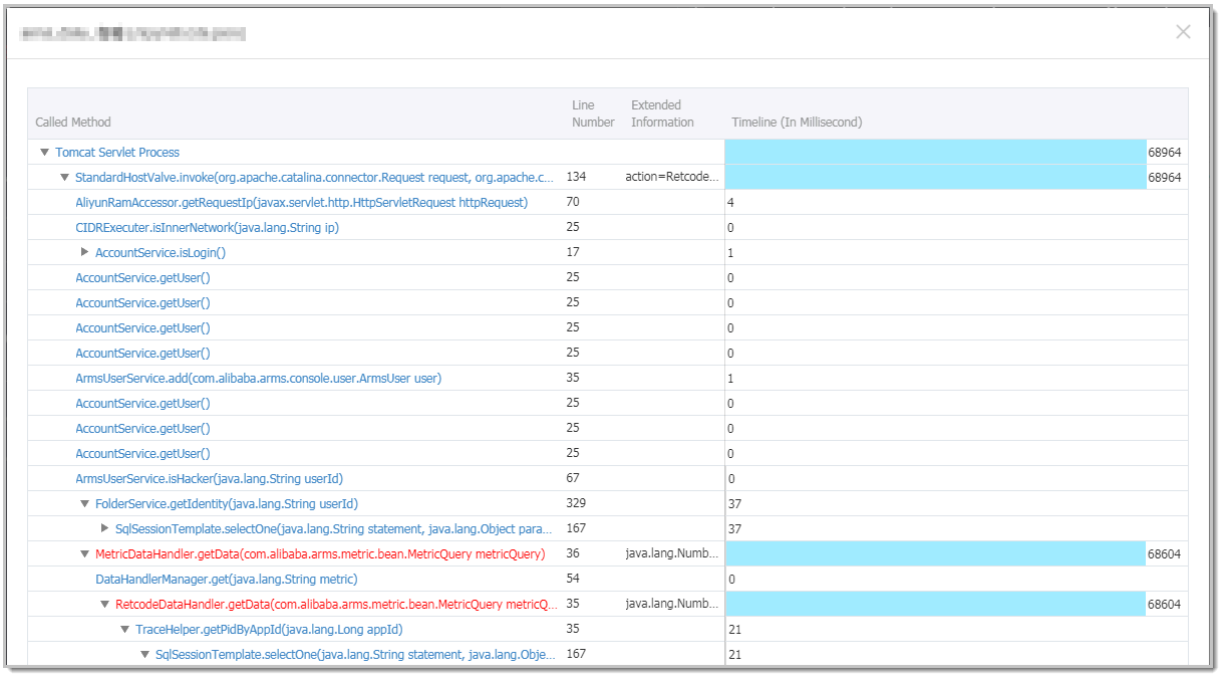
Generate a report on slow services and slow SQL statements

You can go to the SQL analysis report for an application to easily find the slow SQL statements and slow services.



Query distributed trace

You can click the interface snapshot of a slow SQL statement. Then, you can find a request that includes the SQL call, view the call stack of the method, and then identify the code of the problem.



Either from the global angle or from a single call, ARMS comprehensively resolves your pain points in the distributed Java application monitoring field. ARMS supports browser monitoring and business monitoring as well as application monitoring to provide all-

around protection for your sites from key business metrics and customer experience to application performance.

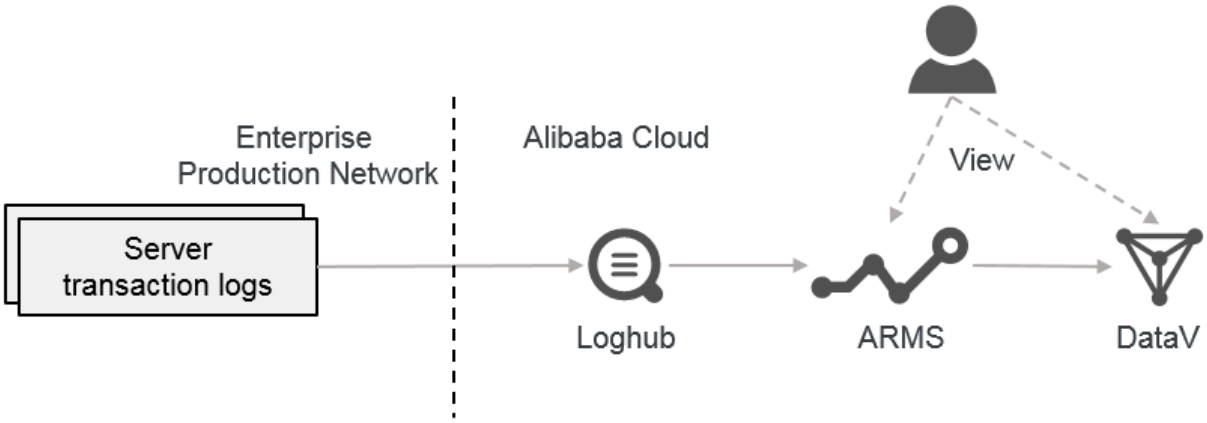
3.2 Real-time IoV monitoring solution

In this use case, a solutions provider for the Internet of Vehicles (IoV) industry from Shanghai adopted the solution based on Application Real-Time Monitoring Service (ARMS) to collect statistics about online conditions of vehicles.

- It is impossible to collect multidimensional statistics on raw data by using databases because a vast amount of data is involved (approximately 100,000 vehicle data records per second).

ARMS-based IoV monitoring solution

The following figure shows the overall architecture.



- The automaker's platform uploads real-time data on new-energy vehicles to Alibaba Cloud by using Message Queue (MQ).
- The ARMS real-time application monitoring function works with MQ to obtain the online data of all vehicles and to perform real-time statistics and storage. The following information is displayed:
  - Computing orchestration and storage: ARMS statistically analyzes the online rate and failure rate based on the reported vehicle information in multiple dimensions, such as the region, vehicle type, and enterprise, and stores the statistical analysis results in columnar storage mode by custom aggregation.
  - Data delivery: ARMS delivers data to downstream services after they call the corresponding API operations.

- Downstream Enterprise Distributed Application Service (EDAS) obtains data from ARMS by calling corresponding API operations, and presents and analyzes data externally by using applications of customers.

Business values of the IoV monitoring solution

- It tracks the running status of vehicles in real time, collects statistics on fault data in real time, and gives feedback on statistic results based on different vehicle models, achieving dramatic quality enhancement.
- It detects violation actions in the first place, for example, swindling for subsidies, by monitoring the running status of new-energy vehicles.

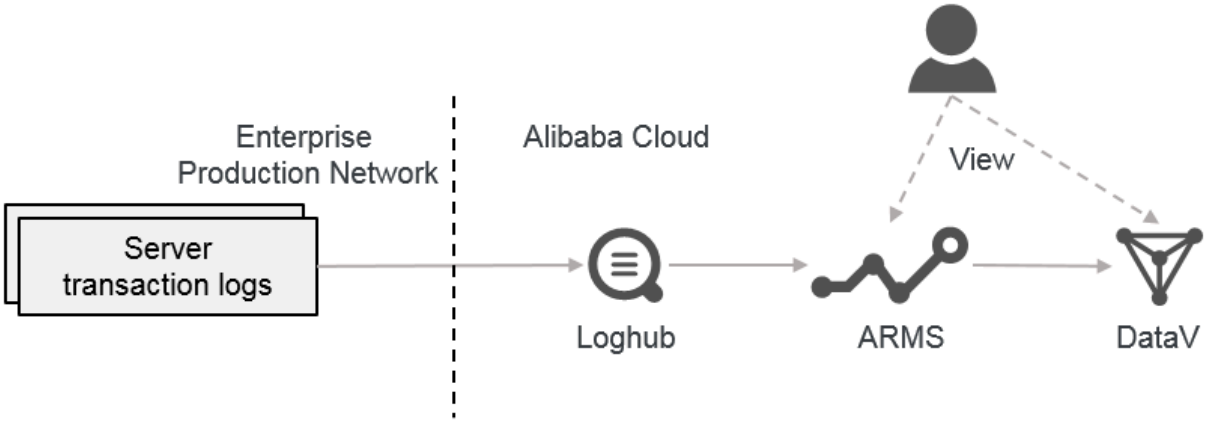
3.3 Real-time monitoring solution in the retail industry

In this use case, a leader in the apparel industry adopted a hybrid cloud solution based on Application Real-Time Monitoring Service (ARMS) to build a real-time monitoring system for its retail industry.

- The original monitoring platform for this customer adopted a traditional commercial Online Analytical Processing (OLAP) database, with considerable license expenditure.
- In addition, it cannot meet business needs in terms of scalability and real-time performance.

ARMS-based retail monitoring solution

The following figure shows the overall architecture.



- Transaction logs are uploaded in real time to LogHub of Alibaba Cloud Log Service by using Logtail.

- The ARMS real-time application monitoring function connects to LogHub for computing and storage, and uses its interactive dashboard to perform real-time analysis and display sales data. The following information is displayed:
  - Computing orchestration and storage: Extracts detailed data of each transaction from logs, including data on the total price and number of items, and then aggregates the data according to multiple dimensions, such as the transaction location, sales company name, and customer membership information.
  - Interactive presentation: Presents the sales status and analysis on various types of cases according to multiple dimensions, such as region, store, member, and product category.
- The data generated by ARMS is delivered to the downstream DataV component for dashboard presentation.

Business value of the ARMS-based monitoring system

- It decreases the total costs of operation (TCO) by hundreds of times, and achieves high-timeliness multidimensional analysis. It can not only help you grasp frontline sales details in real time, but also help you cope with challenges through sales and inventory configuration policies.
- It can also meet the needs in multiple scenarios. The DataV dashboard is used for overall presentation in the monitoring room, and the interactive ARMS dashboard is used for in-depth troubleshooting.

3.4 User experience monitoring scenario

This topic describes a monitoring case regarding user experience.

When a user accesses a service, the whole process can be divided into three phases: page production time (server status), page load time, and page run time. To ensure stable online services, the running status of services is monitored on the server. The existing server monitoring system is quite mature, but the monitoring of the page load time and run time status is far from satisfactory. This is because less emphasis is placed on browser monitoring, due to the mistaken belief that monitoring on the server can partially replace browser monitoring. In this case, the system access details cannot be perceived when the system is running online and a user accesses the system. Therefore, it is extremely difficult to identify a front-end problem that online users occasionally encounter.

Business pain points

- Difficulty in locating performance bottlenecks

  When a user gives feedback that a page loads slowly, it is difficult to quickly find out where the bottleneck is. Is the slow loading of the page caused by a faulty network, resource loading problem, or page Document Object Model (DOM) parsing problem? Is it about the province and country where the user is located, or the user's browser and terminal? These problems cannot recur quickly to allow locating detailed reasons.

- Inability to get error details for user access

  After a system goes live, too many JavaScript (JS) errors will prevent the users from using the service normally. If the error details for user access cannot be obtained promptly, are we at risk of losing many users? If a user gives feedback about page usage details, can we recur the use case in the shortest period? Can we retrieve the error details and quickly fix the issue? This gives you a glimpse of the difficulties developers may have.

- Inability to get asynchronous API call details

  Even if the HTTP status code returned from API calling is 200 every time, it does not necessarily mean that the API is completely normal. If the business logic is abnormal , can it be perceived in time? If all the HTTP status codes returned from API calling are normal, but the whole process consumes a long time, how can we get the whole picture and optimize it? With this kind of information unknown, we cannot identify the issues, and therefore cannot improve user experience.

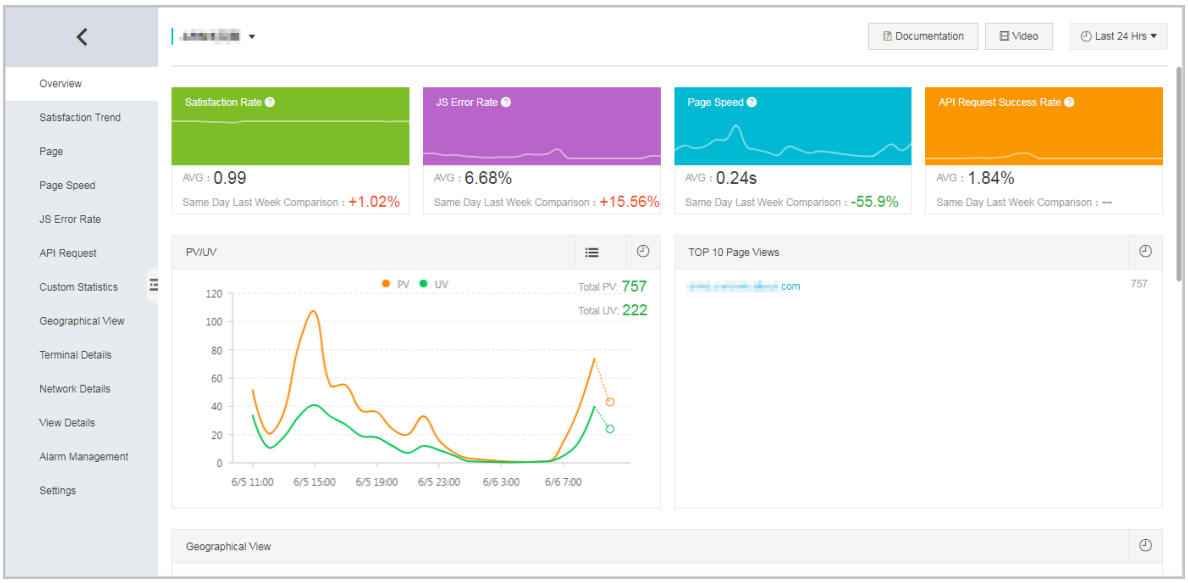
ARMS-based browser monitoring solution

Based on the massive real-time log analysis and processing services provided by Applicatio n Real-Time Monitoring Service (ARMS), the browser monitoring function monitors the access conditions of all real online users and solves the preceding problems.

- Application overview for identifying exceptions

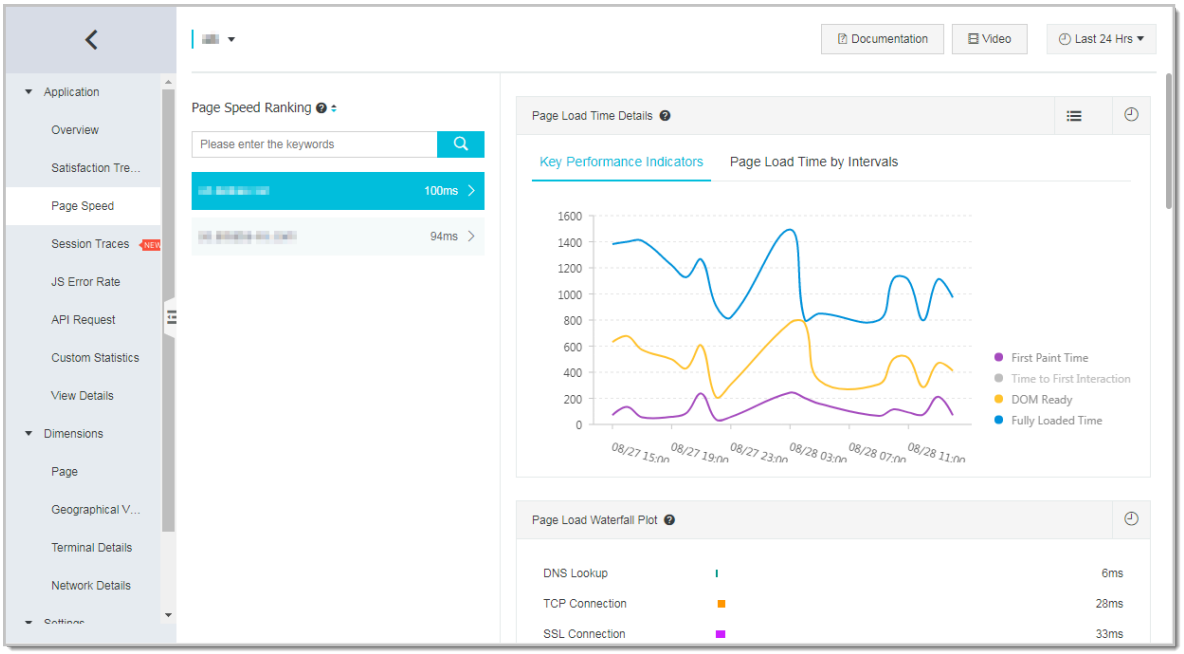
  You can view the information on the application overview page by using the ARMS browser monitoring function, including application satisfaction rating, JS error rate, page

speed, API request success rate, and page view (PV) details. In the following example, the average JS error rate is 6.68%, up 15.56% from last week.



- Performance data trend and waterfall chart

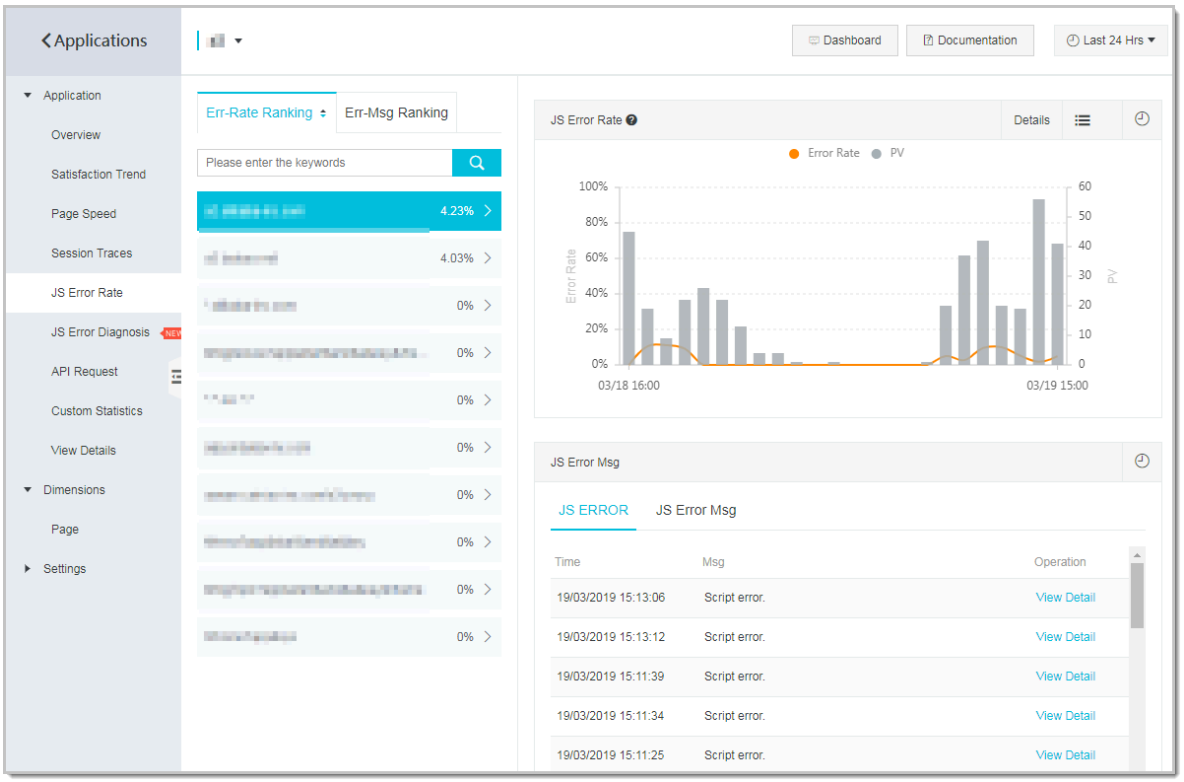
  On the **Page Speed** page, you can view detailed metrics related to the page performance and the page loading waterfall chart. Then, you can locate the performance bottleneck according to detailed data.



- JS error rate and error aggregation

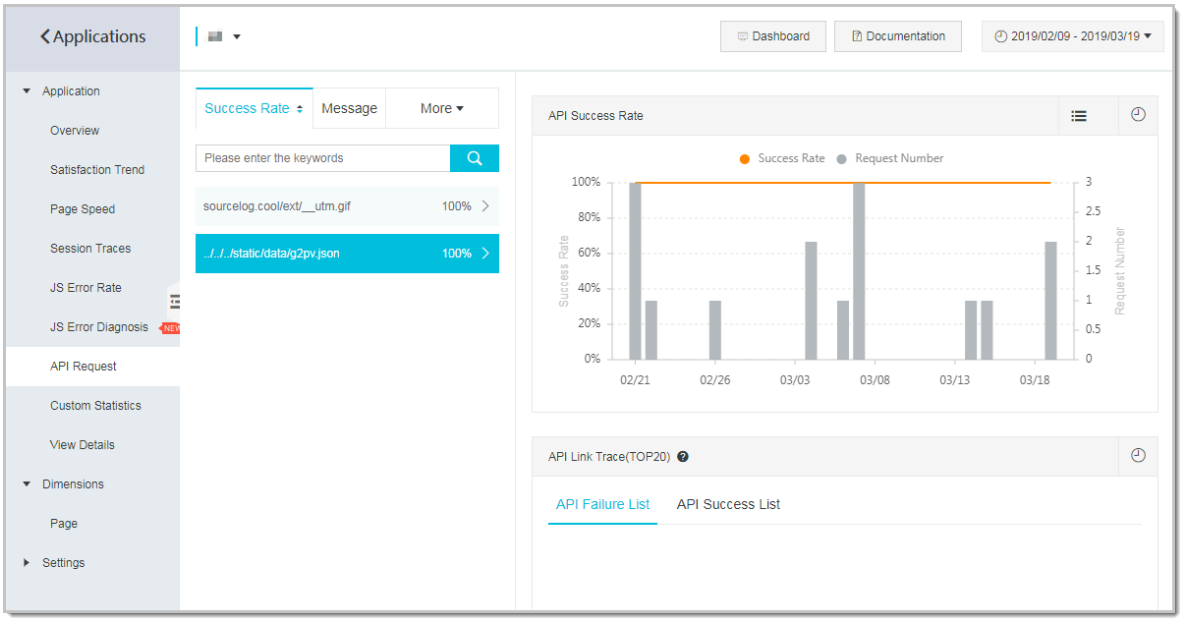
  On the **JS Error Rate** page, you can view a ranking on error rates in descending order and a ranking on error aggregation. Therefore, you can get a straightforward impression

about which pages have the highest JS error rate, and which errors are the most common.



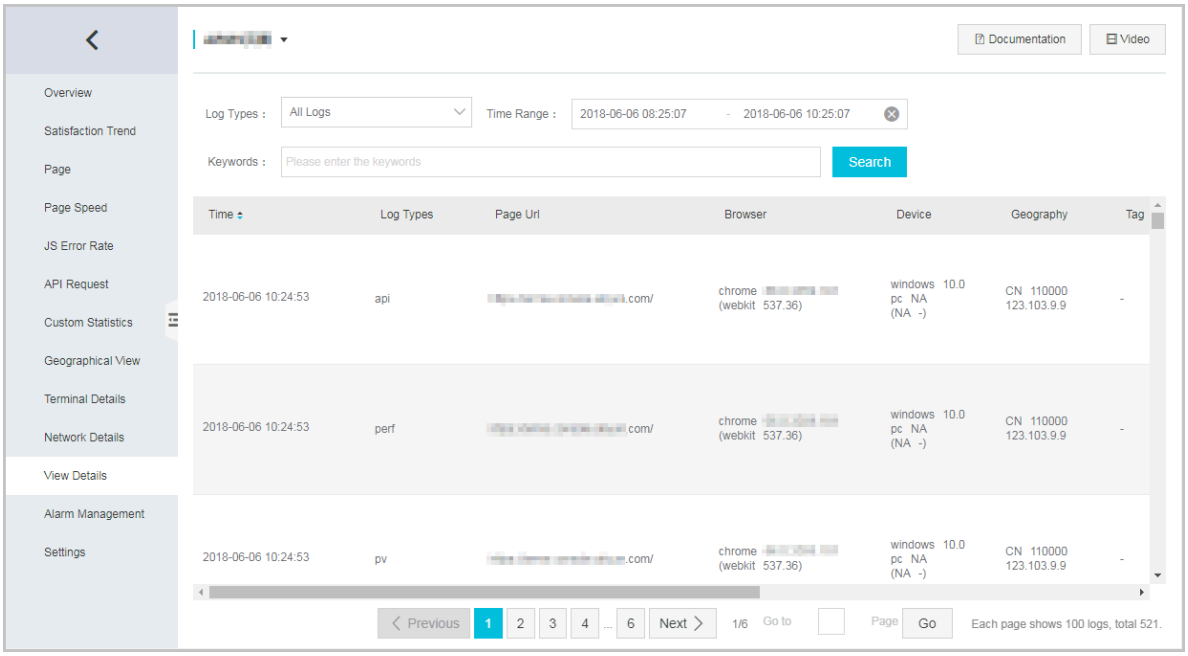
- API request

  On the **API Request** page, you can view data on API call success rates and time consumption, understanding every detail of the APIs.



- Access details

  On the **View Details** page, you can view the access details. For example, you can locate an error according to the error information, including stack, file, line, and column.



4 Terms

This topic lists the terms of Application Real-Time Monitoring Service (ARMS).

B | C | J | S | Y | Z

B

Alert rule	An alert rule defines how to trigger alerts based on datasets and send alert notifications through a specific channel. The severity of an alert can be Warning, Error, or Critical.

[Back to the top]

C

Collection rule	A collection rule defines how to collect data from data source instances in a custom monitoring job. You must define collection rules for custom monitoring jobs.

[Back to the top]

J

Dashboard	A dashboard is a set of interactive monitoring data reports customized based on datasets in ARMS. The dashboard can display datasets by using different types of charts. The query time range of the dashboard is user-defined.

[Back to the top]

S

Dataset	A dataset defines how to pre-aggregate and persistently store the logs that are collected in monitoring jobs. You can define a dataset directly or indirectly by using the dashboard and alert notification features.
Dataset dimension	A dataset dimension is the key used for aggregation when a dataset is created. It is similar to the GroupBy column name in a database, or an attribute in multidimensional Online Analytical Processing (OLAP). A dataset performs aggregation operations on the real-time data based on the configured dimensions.

Dataset metric	A dataset metric is a specific monitoring metric stored in a dataset, which is typically of the numerical type and similar to a value in multidimensional OLAP. ARMS metrics correspond to values of Count, Max, Sum, and Count Distinct after real-time computing.
Data cleansing	Data cleansing is a process during which operations such as splitting and static join are performed on custom monitoring logs to convert them into standard key-value (KV) pairs.
Data screening	Data screening is used to filter the data in datasets for dataset calculation. Data that do not meet criteria are filtered out from the dataset.
Data source	A data source is the source of logs obtained by ARMS custom monitoring jobs, including Elastic Compute Service (ECS), Message Queue (MQ), and SDK data sources.

[Back to the top]

Y

Mapping table	A custom static mapping table is used for mapping query results to business attribute fields. For example, you can map the province, city, and district names in query results to postal codes.

Z

Custom monitoring job	A custom monitoring job is a process in which ARMS captures, processes, and stores data, and then presents and exports the results. Custom monitoring jobs are divided into the following categories:

[Back to the top]

Links to other terms

- Terms of application monitoring
- Terms of browser monitoring

5 Description

This topic provides a version release record of Application Real-Time Monitoring Service (ARMS) to describe the feature changes of previous ARMS releases.

V2.5.2

Date: February 21, 2019

New features:

- Application monitoring
  - Supported one-click monitoring of applications deployed on Elastic Compute Service (ECS) instances. For more information, see Install the ARMS agent for applications on ECS instances with one click.

V2.5.1

Date: February 1, 2019

New features:

- Application monitoring
  - Supported using the ARMS console to monitor applications deployed in Container Service for Kubernetes or in open-source Kubernetes environments. For more information, see #unique_16 and Install the ARMS agent for applications in open-source Kubernetes environments.
- Browser monitoring
  - Supported browser monitoring for DingTalk, Alipay, WeChat, and other mini programs. For more information, see Monitor DingTalk mini programs, Monitor Alipay mini programs, Monitor WeChat mini programs, and Monitor other mini programs.

V2.5.0

Date: January 21, 2019

New features:

- Application monitoring
  - Added the label function. You can classify application sites by label in the monitored application list.

Optimization and improvement:

- Browser monitoring
  - Optimized the alert function.

V2.4.8

Date: January 6, 2019

New features:

- Application monitoring
  - Supported class conflict detection.
  - Supported dynamically enabling and disabling URL convergence rules.

V2.4.7

Date: December 13, 2018

New features:

- Application monitoring
  - Supported one-click monitoring of Java applications. For more information, see Quickly install the ARMS agent for Java applications by using scripts.
  - Supported PHP application monitoring. For more information, see #unique_23.

Optimization and improvement:

- Application monitoring
  - Optimized the traditional method of enabling Java application monitoring. For more information, see Manually install the ARMS agent for Java applications.
- Browser monitoring
  - Optimized the JavaScript (JS) error locating function. Online JS code is often compressed into one line, making error locating impossible based on the browser-reported erroneous line number. ARMS browser monitoring can restore the error location through source maps. For more information, see Use ARMS browser monitoring for JS error diagnosis.

V2.4.6

Date: October 26, 2018

New features:

- Application monitoring
  - Supported NoSQL database monitoring.
  - Provided a new overview page, with an optimized page layout and new modules , such as application monitoring details, browser monitoring details, and product express.

V2.4.5

Date: September 17, 2018

New features:

- Application monitoring
  - Included MongoDB in supported components and frameworks.

V2.4.4

Date: August 17, 2018

New features:

- Application monitoring:
  - Supported thread profiling. When a thread encounters request time-out, you can quickly locate the time consumption of all internal stacks. For more information, see Use ARMS TProf to analyze code problems.
- Browser monitoring:
  - Supported resource loading details. You can quickly locate all slowly loaded resources on the page, such as images, JS, CSS, and APIs. For more information, see Slow session tracking.
- Custom monitoring:
  - Commercially available.

V2.4.3.4

Date: July 16, 2018

New features:

- Application monitoring:
  - Supported comprehensive troubleshooting. You can query the distributed trace of application monitoring through business ID, which greatly improves the problem diagnosis efficiency.

V2.4.3.3

Date: June 16, 2018

Optimization and improvement:

- General:
  - Supported API calls by RAM users and improved security authentication for API calls. For more information, see Create and authorize RAM sub-accounts.
- Application monitoring:
  - Revised the homepage of application monitoring to show more core summaries.
  - Optimized the memory snapshot capture and analysis function in different network environments, which improves the efficiency of capture by more than 50%.

V2.4.3

Date: May 19, 2018

New features:

- Application monitoring:
  - Added the memory snapshot analysis function, allowing you to grasp the distribution of memory objects with a glance and quickly locate memory leaks. For more information, see Memory snapshots.
  - Supported the custom configuration of monitoring methods, allowing you to dynamically configure methods to monitor and capture exceptions, with an increased monitoring granularity. The configuration takes effect immediately without restarting the machine.
  - Added the application monitoring overview page to facilitate problem locating and troubleshooting.
  - Added Message Queue (MQ) trace monitoring to quickly locate message delay, errors , and message accumulation.

V2.4.2

Date: April 19, 2018

New features:

- Application monitoring:
  - Added the JVM monitoring function to monitor a series of important JVM metrics, including heap memory, non-heap memory, and number of threads. For more information, see JVM monitoring.
  - Added the host monitoring function to monitor a series of host performance metrics, including the CPU, memory, disks, and network traffic. [Host monitoring.
  - Added the custom configuration function, allowing you to modify configurations directly in the console, including trace sampling, agent switch setting, and threshold setting. For more information, see Custom configurations.
- Browser monitoring:
  - Added the sample reporting configuration feature, which reduces the number of user reports and loads through random sample reporting. For more information, see Configurations of the browser monitoring SDK.

V2.4.1

Date: March 22, 2018

New features:

- General:
  - Added alert rules, which support batch import and export of alerts to make alert management more convenient.
  - Supported monitoring of applications in Enterprise Distributed Application Service (EDAS), with a 50% off discount for long-term use.
- Application monitoring:
  - Supported display of an online agent list, allowing you to be informed of the installed agent versions and their statuses.
- Browser monitoring:
  - Supported display of the response time in the format of quantile rather than average values, allowing you to view the response time and distribution of slow requests more clearly.

- Application monitoring:
  - Supported quantile operators, allowing you to view more detailed metric statistics in the format of quantile values, in addition to average, maximum, and minimum values .

V2.4.0

Date: February 26, 2018

New features:

- ARMS browser monitoring and application monitoring are commercially available.

V2.3.3.1

Date: January 31, 2018

New features:

- Supported all EDAS applications. EDAS users can connect EDAS applications to ARMS with one click. For more information, see Install the ARMS agent for applications in EDAS with one click.

V2.3.3

Date: January 14, 2018

New features:

- Provided a brand new homepage and documentation with an overhauled product page. The product is officially positioned as an APM product, integrating **application monitoring**, **browser monitoring**, and **custom monitoring**.
- Optimized the APM-oriented base function to provide a centralized alert platform and interactive dashboard for the following modules: application monitoring, browser monitoring, and custom monitoring.
- Supported the following regions: China (Hangzhou), China (Beijing), China (Shanghai) (new), China (Qingdao) (new), and China (Shenzhen) (new).

Optimization and improvement:

- Optimized the console: one console used across all regions in China.
- Optimized the alert query page, and supported multi-dimensional alert query.

V2.3.1

Date: December 14, 2017

New features:

- Supported most common Java APM functions, including call topology, tracing, slow transactions reports, and slow SQL querying. Supported more than 10 types of Java stack frameworks required by common cloud users, such as Spring, Redis, MySQL (RDS), and Dubbo. You can connect applications to ARMS by installing the Java agent without modifying application code.

Optimization and improvement:

- Optimized the dashboard widget dataset setting. Basic and compound metrics can be displayed on the same page.
- Optimized the new user tour guide logic. By default, the new user prompt function is enabled only upon initial logon.

V2.2.6.2

Date: September 23, 2017

New features:

- Supported heat map on the interactive dashboard.
- Added the exception splitter to support data cleansing of Java exceptions.
- Added the IP address-to-physical address mapping module to the data cleansing process.
- Supported a value of the NULL type for a dataset filter.

Optimization and improvement:

- Optimized the alert content. The email alert now contains the log sampling content.
- Optimized the NGINX template, and provided more straightforward and easy-to-use NGINX monitoring functions.

V2.2.6

Date: August 31, 2017

New features:

- Released quality- and performance-focused browser monitoring.
- Supported MQ data sources in business monitoring.

V2.2.5

Date: July 26, 2017

New features:

- Supported MQ data sources in ARMS.
- Supported query of millions of data records in a dataset.
- Displayed alerts of the same type together for a cleaner look.
- Supported creation and access of the interactive dashboard sharing links by external users so that they can view the dashboard without logon.
- Added the black-and-white theme to the interactive dashboard for improved aesthetic effect.
- Enhanced interactive tables. The column header can be frozen, and records can be sorted in reverse chronological order.

V2.2.4

Date: June 21, 2017

New features:

- Supported common datasets.
- Upgraded the front-end visualization component to G2, and supported multiple datasets in a single widget.
- Supported external trace of ARMS interactive dashboards.
- Supported on-the-fly editing of dataset alerts.
- Supported querying of millions of data records with APIs.
- Supported splitting of XML and logs with line breaks in data cleansing.

V2.2.3

Date: March 30, 2017

New features:

- Supported Python for dataset POP APIs.
- Enhanced interactive dashboard widgets. For example, the area chart supports stacking mode.
- Added the China (Beijing) region.

Optimization and improvement:

- Optimized the overall alert function, and supported flexible setting of the alert time range, alert severity, and notification method for individual alerts.
- Supported associated output of dimension tables for API query results.

V2.2.2

Date: March 8, 2017

New features:

- Supported dimension tables. You can customize the mapping of attributes, for example, mapping a postal code to a specific district, city, or province.
- Released the standard job template, allowing you to quickly create a custom monitoring job based on the standard job template.
- Added the data recovery function. When an alert is cleared, a notification will be sent by email.

Optimization and improvement:

- Further optimized the interactive dashboard, and added top N filtering, zero fill, and time granularities.

V2.2.1

Date: February 17, 2017

New features:

- Added rate operators, applicable to scenarios such as rate change statistics.
- Supported RAM authorization rules.

Optimization and improvement:

- Reduced the response time dramatically for ARMS real-time computing. In some cases, the response time is shortened from more than 10s to less than 5s.
- Further optimized the interactive dashboard, with finer control over resizing of various windows.

V2.2.0

Date: January 23, 2017

New features:

- Released the interactive dashboard function.
- Supported duplication, import, and export of jobs, allowing you to quickly duplicate existing monitoring plans.
- Supported user job self-check, including job error statistics and error sampling.
- Supported compound operators.

V2.1.0

Date: October 8, 2016

New features:

- Added Log Service LogHub, SDK, and API data sources.
- Supported common advanced operators in alerts. You can specify the time range as the past N minutes, days, or hours, and can set the MAX, AVG, and MIN thresholds of metrics.
- Supported advanced operators: Count Distinct and Sample.
- Added the interactive dataset query page.

V2.0.0

Date: August 4, 2016

New features:

- Supported ECS data sources.
- Supported various cleansing calculations, including single, multiple, and sequential separators, KV cleansing, JSON cleansing, and other custom cleansing logic (such as exception stack).
- Supported multiple types of aggregation calculations, including all common aggregatio n calculations at various time granularities, such as SUM, COUNT, and MAX.
- Supported definition of various types of content metrics, ranking, and notification methods for alert settings.
- Supported comprehensive solutions based on time series and other similar dimensions for chart customization, integrated with common presentation formats and dashboard configurations, such as column chart, line chart, pie chart, flap display, and table, and provided data drill up and drill down capabilities.
- Supported creation of a dashboard by dragging and dropping existing alerts and charts.

6 Differences between ARMS and Tracing Analysis

Both Application Real-Time Monitoring Service (ARMS) and Tracing Analysis can solve tracing problems in a distributed environment. This topic describes the differences between ARMS and Tracing Analysis.

Background

ARMS is an application performance management (APM) product of Alibaba Cloud. With ARMS, you can quickly and conveniently build application monitoring capabilities with a response time within several seconds for businesses and enterprises based on custom dimensions such as the browser, application, and business.

Tracing Analysis provides a complete set of tools for distributed application development, including trace restoration, call statistics, trace topology, and application dependency analysis. Tracing Analysis helps you quickly analyze and diagnose performance bottlenecks in a distributed application architecture, improve the efficiency of microservice development and diagnosis, and reduce the costs of developing trace monitoring applications (such as Jaeger and Zipkin) and related storage services (such as HBase and Elasticsearch).

Comparison between ARMS and Tracing Analysis

Difference	ARMS	Tracing Analysis
Product positioning	ARMS is an APM product that supports application performance monitoring, user experience monitoring, tracing, and problem diagnosis.	Tracing Analysis is designed for distributed tracing.
Access method	Non-intrusive agent loading	Intrusive SDK programming
Billing method	Billed based on the number of agents, with competitive pricing	Billed based on the number of requests, with competitive pricing
Programming language	Java and PHP	Java, PHP, Go, Python, JavaScript, .NET, and C++

Difference	ARMS	Tracing Analysis
Thread and memory diagnosis	Supported	Not supported
Local method stack	Supported	Not supported

The following describes the differences between ARMS and Tracing Analysis in terms of the product positioning, access method, and billing method.

Product positioning

ARMS is positioned as a heavyweight APM product with rich functions. Applications can access ARMS by installing an agent, which provides functions such as performance monitoring, user experience monitoring, tracing, and fault diagnosis.

Tracing Analysis is specifically designed to solve tracing problems in a distributed environment. You can implement distributed tracing through the Tracing Analysis SDK, which only supports trace monitoring.

Access method

Positioned as an APM product, ARMS is accessed in non-intrusive mode, without code modification. You need to install an agent to the application you want to monitor and modify the application startup mode. For example, when you use ARMS to monitor a Java application, you need to add the **-javaagent** startup parameter during the application startup.

Positioned as a distributed tracing product, Tracing Analysis is based on open-source products Jaeger and Zipkin and open-source standard OpenTracing. You can access Tracing Analysis by using an SDK based on Jaeger, Zipkin, or OpenTracing, with the following advantages:

- You can seamlessly migrate applications that use SDKs based on Jaeger, Zipkin, or OpenTracing to Tracing Analysis, without code modification.
- You do not need to worry about vendor lock-in because the Tracing Analysis SDK is based on open-source standards.
- With the help of the community, you can support a large number of programming languages at a time, greatly simplifying trace monitoring during development in a heterogeneous environment.

Billing method

Similar to other APM products, ARMS is billed based on the number of agents when you use functions such as application monitoring and browser monitoring. The resulting fee may account for a large proportion of your budget. Based on a high-performance and efficient architecture, ARMS is charged at a rate only 10% to 20% of the average rate of the industry.

Designed specifically for trace diagnosis in a distributed environment, Tracing Analysis is billed based on the number of requests, at a low rate.

More information

Though positioned differently, both ARMS and Tracing Analysis are monitoring products for development in Alibaba Cloud and will be interconnected in the future.

- ARMS will support custom distributed tracing based on the Tracing Analysis SDK.
- Tracing Analysis will support queries in the ARMS console for optimized diagnosis experience.

Related topics

Tracing Analysis