Application Performance Management

Best Practices

Issue	01
Date	2024-09-13

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 Managing an E-Commerce Application

This section describes how to connect an e-commerce application to Application Performance Management (APM) for management. Information about the application:

- This is a Java application that is not deployed using JBoss. It can be deployed on the SUSE 12 SP2 VM in JRE 1.8 or later.
- The application contains five microservices, each of which contains an instance:
  - **apigw**: API gateway, which is responsible for service authentication, request throttling, and filtering.
  - **product**: Product management, including product query and purchase.
  - **user**: User management, including user login and identity authentication during product purchase.
  - **dao**: Data persistence layer, which is responsible for database request operations.
  - **MySQL database**: Database.

The following describes how to connect the e-commerce application to APM for management.

Process

1. Deploy the application on a Huawei Elastic Cloud Server (ECS). Note that APM supports Huawei ECSs only.
2. Install the ICAgent. The ICAgent is a collection agent for collecting topology and tracing data in real time. You need to install the ICAgent on the ECS where the application is deployed.
3. Modify application startup parameters to ensure that APM can monitor the application.
4. Manage the application on APM, for example, view the application status on the topology page and view user experience on the transaction page.

Procedure

Step 1 Deploy the application on a Huawei ECS.

1. Register a cloud account, top up the account, and purchase a Linux ECS. For details, see **Logging In to an ECS**.

   To ensure application performance, select an ECS with at least 2 vCPU cores and 4 GB memory. Select the image for the SUSE 12 SP2 Operating System (OS), which is the only OS supported by the application.
2. Log in to the ECS, create an application directory, and go to the directory, for example, **mall**.
3. Set up the MySQL database and record the address, port number, username, and password for accessing the database. The demo supports automatic database creation.
4. Download the application package to the **mall** directory and install it.

   ```
   curl -l http://demos.obs.myhuaweicloud.com/demo_install.sh > demo_install.sh && bash demo_install.sh
   ```

   NOTE

   During the installation, replace the database information according to the following procedure:

   1. Download and open the **.sh** file shown in the command.
   2. Download and open the **demo_03.tar** file in the **.sh** file.
   3. Open the **.yaml** file in the **demo_03.tar** file and modify the parameters under **MySQL**:
      - **host**: address for setting up the database
      - **port**: port
      - **user**: username for accessing the database
      - **pass**: password for accessing the database
   4. Save the settings and go to the next step.

Step 2 Install the ICAgent. For details, see **Initial Installation**.

Step 3 Modify application startup parameters to ensure that the application can be monitored by APM.

Specifically, add the following parameters to the application startup script:

Parameter	Description	Example
-javaagent	JAR package that the collection probe depends on. The value is always **/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar**.	/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar

Parameter	Description	Example
-Dapm_application	Application name, which can be customized.	vmall
-Dapm_tier	Application microservice name.	In this example, microservice names include **apigw**, **product**, **user**, and **dao**.

Application startup scripts:

Before the modification:

```
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/ecommerce-persistence-service-0.0.1-SNAPSHOT.jar
--spring.config.location=file:/root/testdemo/application_dao.yml > dao.log &
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/ecommerce-api-gateway-0.0.1-SNAPSHOT.jar --
spring.config.location=file:/root/testdemo/application_api.yml > api.log &
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/ecommerce-user-service-0.0.1-SNAPSHOT.jar --
spring.config.location=file:/root/testdemo/application_userservice.yml > user.log &
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/ecommerce-product-service-0.0.1-SNAPSHOT.jar --
spring.config.location=file:/root/testdemo/application_prod.yml > prod.log &
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/cloud-simple-ui-1.0.0.jar --
spring.config.location=file:/root/testdemo/ui.properties > ui.log &
```

After the modification (the modified information is shown in bold):

```
nohup java -javaagent:/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar -
Dapm_application=vmall -Dapm_tier=vmall-dao-service -Xms512m -Xmx2048m -jar /root/testdemo/
ecommerce-persistence-service-0.0.1-SNAPSHOT.jar --spring.config.location=file:/root/testdemo/
application_dao.yml > dao.log &
nohup java -javaagent:/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar -
Dapm_application=vmall -Dapm_tier=vmall-apigw-service -Xms512m -Xmx2048m -jar /root/testdemo/
ecommerce-api-gateway-0.0.1-SNAPSHOT.jar --spring.config.location=file:/root/testdemo/
application_api.yml > api.log &
nohup java -javaagent:/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar -
Dapm_application=vmall -Dapm_tier=vmall-user-service -Xms512m -Xmx2048m -jar /root/testdemo/
ecommerce-user-service-0.0.1-SNAPSHOT.jar --spring.config.location=file:/root/testdemo/
application_userservice.yml > user.log &
nohup java -javaagent:/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar -
Dapm_application=vmall -Dapm_tier=vmall-product-service -Xms512m -Xmx2048m -jar /root/testdemo/
ecommerce-product-service-0.0.1-SNAPSHOT.jar --spring.config.location=file:/root/testdemo/
application_prod.yml > prod.log &
nohup java -Xms512m -Xmx2048m -jar /root/testdemo/cloud-simple-ui-1.0.0.jar --
spring.config.location=file:/root/testdemo/ui.properties > ui.log &
```

Step 4 Restart the application. After about three minutes, application data is displayed on the APM console.

You can monitor applications and locate application exceptions on the **Dashboard** and **Topology** pages of the APM console. For details, see **APM User Guide**.

----End

2 Locating Application Problems Based on Transactions

This section describes how to locate application problems based on transactions. For transaction analysis, APM analyzes service flows on application servers in real time, and uses Application Performance Index (Apdex) to rate applications to reflect user satisfaction with applications. You can locate application problems or performance bottlenecks based on user satisfaction.

This section describes how to locate problems of an e-commerce application connected to APM based on transactions.

Process

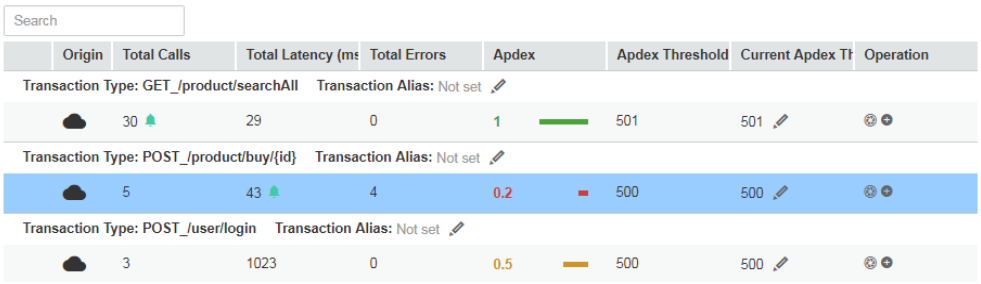
1. Identify a problem.
2. Locate the cause.

Procedure

Step 1 Identify a problem.

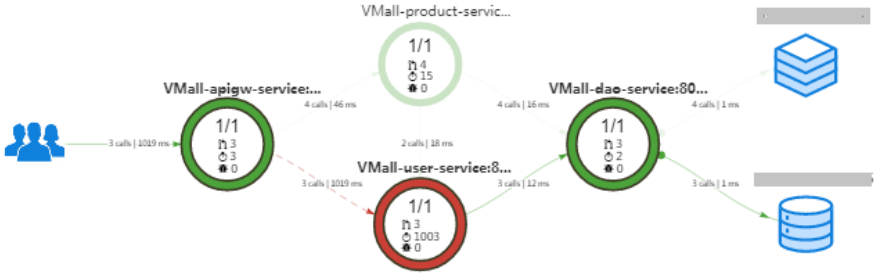
When using APM to inspect applications, detect the transaction whose Apdex is red on the **Transactions** page.

Figure 2-1 Detecting the abnormal transaction during inspection



Step 2 Locate the cause.

1. Click in the **Operation** column. On the topology page that is displayed, view the instance details and locate the abnormal microservice named **user** based on the color.



2. Right-click the **user** instance and choose **Find Call-Chain** from the shortcut menu. On the **Call Chain** page that is displayed, find out the method with the longest time line, as shown in the following figure.

Application	Method	Param	Status	Time Line(ms)	
vmall-apigw...	org.apache.catalina.core.standardhostvalve.invoke	/use...	Success		<1
vmall-apigw...	org.springframework.web.servlet.frameworkservlet.dopost		Success		<1
vmall-apigw...	com.netflix.hystrix.hystrixcommand.queue		Success		<1
vmall-apigw...	com.netflix.hystrix.hystrixcommand$2.call		Success		<1
vmall-apigw...	org.apache.http.impl.client.closeablehttpclient....		Success		2
vmall-apigw...	org.apache.http.protocol.httprequestexec...	/use...	Success		<1
vmall-user-...	org.apache.catalina.core.standardho...	/use...	Success		<1
vmall-user-...	org.springframework.web.servle...		Success		2
vmall-user-...	com.huawei.cloud.common.wor...		Success		1001

----End

3 Locating Problems Based on Topologies

Background

In scenarios where external requests increase sharply or the load changes abruptly, application performance problems may occur. For example, external requests are responded slowly or some requests are abnormal. Quickly identifying, locating, and handling application performance problems become urgent needs.

APM is a cloud application diagnosis service and has powerful analysis tools. It displays the application status, call processes, and user operations through topologies, tracing, and transaction analysis, so you can quickly locate and resolve faults and performance bottlenecks.

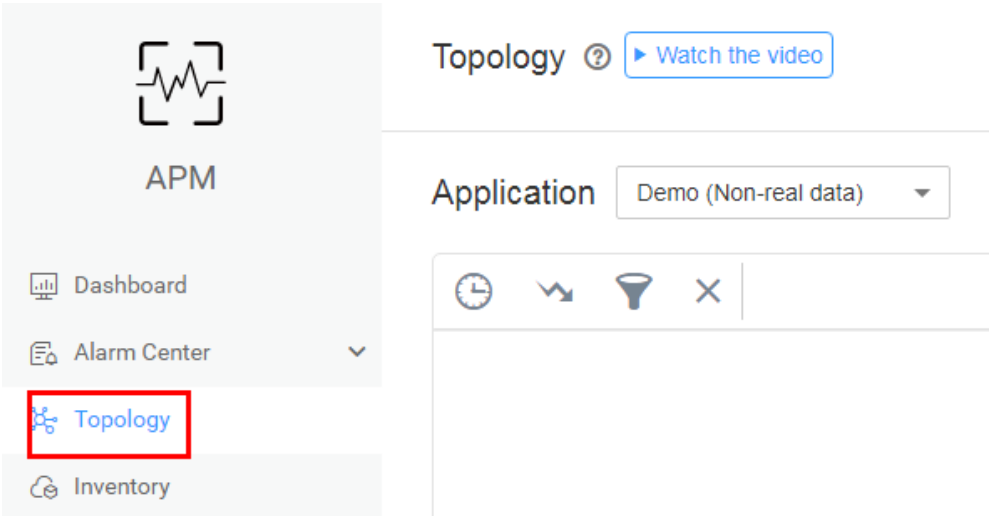
For example, you can view the call relationships between services and quickly locate abnormal instances through topologies. You can also drill down to service resources to find out the faulty method tracing and determine root causes.

Application Scenarios

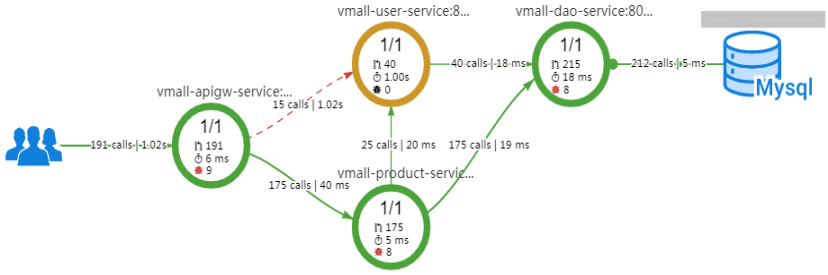
- Routine inspection of application metrics such as latency, throughput, and number of errors
- Quick locating of error calls

Procedure

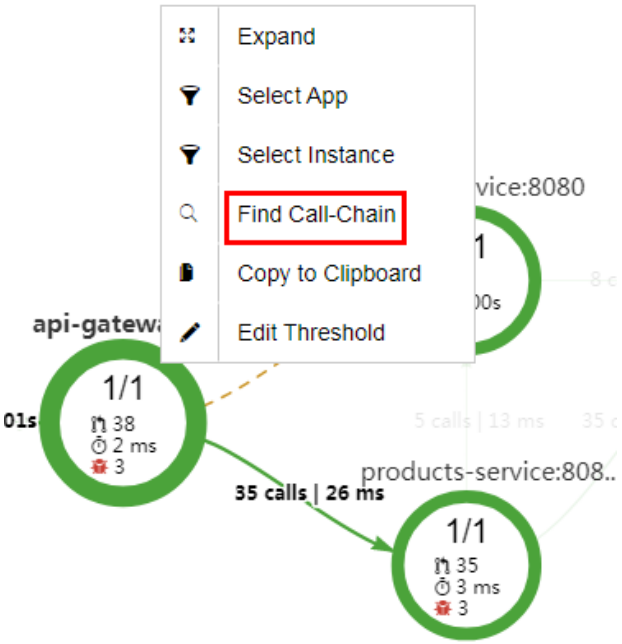
Step 1 Log in to the APM console. In the navigation pane, choose **Topology**.



Step 2 On the **Topology** page, view metrics, service relationships, and top statistics.



Step 3 Right-click the abnormal tracing and choose **Find Call-Chain** from the shortcut menu.



Step 4 On the **Tracing** page, view the error or high-latency tracing, and click **View Call Relationship** in the **Operation** column.

Step 5 On the **Call Relationship** page, obtain the error call information and determine the root cause. For example, click **View Details** to obtain the error call information.

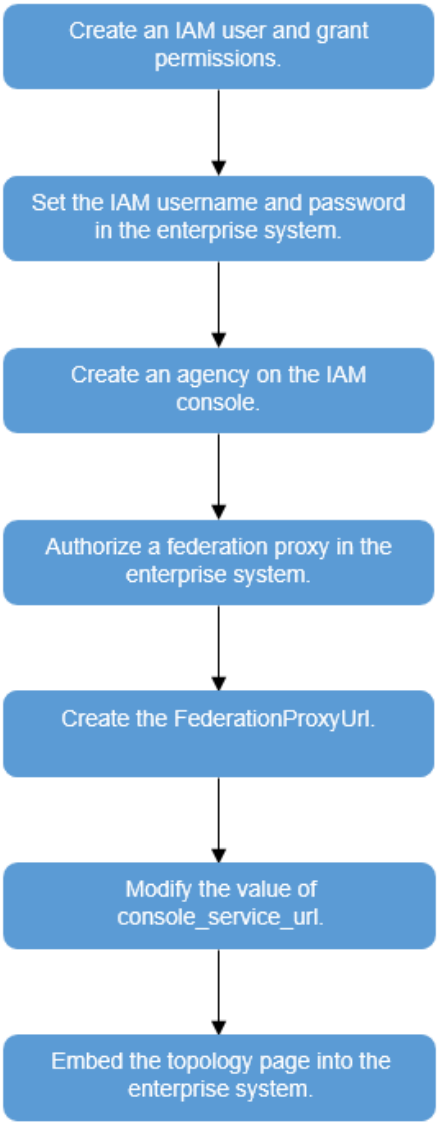
----End

4 Embedding the APM Topology Page to a Customer's Self-Built System

Background

The Application Performance Management (APM) topology page can be embedded into a customer's self-built system. Specifically, customize an identity broker through the federation proxy mechanism of Identity and Access Management (IAM) and embed a login link to the customer's self-built system. The customer can then view the topology page on its self-built system without logging in to HUAWEI CLOUD websites.

Process



Procedure

Create an identity broker and a login address (**FederationProxyUrl**) according to steps 1 to 5. These steps are closely related to IAM. For details, see **Custom Identity Broker**. After performing operations on IAM, embed the topology page according to steps 6 to 7.

Step 1 Create an IAM user, for example, **userB** in **DomainA** and grant the **Security Administrator** and **Agent Operator** permissions (global service-global project) to the user.

NOTE

DomainA and **userB** are used as examples. In practice, use actual domain and user names.

Step 2 Set the username and password of **userB** in the configuration file of the customer's system to obtain the user authentication token and call APIs. You are advised to encrypt the password.

Step 3 Create an agency on the IAM console and grant permissions to the agency as required.

Step 4 In the enterprise system, create a user group with the same name as that of the agency created in the preceding step, add local users to the group, and grant the users permissions required for logging in to HUAWEI CLOUD through a federation proxy.

Step 5 Log in to the enterprise system and access the federation proxy. This proxy will create a cloud service login address, that is, **FederationProxyUrl**.

Example of **FederationProxyUrl**:

```
https://auth.huaweicloud.com/authui/federation/login?
idp_login_url={enterprise_system_loginURL}&service={console_service_url}&logintoken={logintoken}
```

Step 6 Change the value of **console_service_url** in **FederationProxyUrl** to the address of the cloud service console.

Example of **console_service_url**:

```
https://console.huaweicloud.com/apm/?region=ap-southeast-1&locale=zh-cn&inFrame=true#/apm/atps/
topology
```

Table 4-1 console_service_url parameters

Parameter	Description
region	Region where you are located. You can obtain the value from the address box of the browser after logging in to a HUAWEI CLOUD service. For example, ap-southeast-1.
locale	Language. For example, zh-cn.
inFrame	Embedded page identifier. If the value is **true**, the header, footer, and menu bar of the HUAWEI CLOUD console page will be hidden.

Step 7 Use iframe to embed the APM topology page into the enterprise system. The example code is as follows:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>iframe test</title>
</head>
<body>
  <iframe id="apmTopo" src="${FederationProxyUrl}">
</body>
</html>
```

----End

