



应用性能管理

最佳实践

文档版本 01

发布日期 2018-05-14

华为技术有限公司



版权所有 © 华为技术有限公司 2020。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

华为技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编： 518129

网址： <https://www.huawei.com>

客户服务邮箱： support@huawei.com

客户服务电话： 4008302118

目录

1 使用 APM 管理电商应用.....	1
2 定位事务异常原因.....	4
3 使用拓扑定位异常.....	6
4 自建系统内嵌 APM 拓扑页面.....	8

1 使用 APM 管理电商应用

本实践基于华为云服务编写，以某手机销售电商应用为例，如何将APM接入该电商应用并管理起来。该电商应用情况介绍：

- 该应用是未使用JBoss部署的JAVA应用，支持部署在SUSE12 SP2虚拟机上，运行环境支持JRE1.8及以上版本。
- 该应用包含五个微服务，每个微服务包含一个实例：
 - API网关服务：名称为apigw，主要负责应用整体的服务鉴权、限流、过滤等。
 - 商品管理服务：名称为product，主要负责商品查询、购买等。
 - 用户管理服务：名称为user，主要负责用户登录，以及购买商品时的用户身份核实等。
 - 数据持久服务：名称为dao，主要负责请求数据库操作。
 - 数据库服务：MySQL数据库。

下面介绍如何将APM接入该电商应用并管理起来。

操作流程

1. 将应用部署到华为云服务器。APM当前只支持华为云服务器，所以需要将应用部署到华为云服务器上。
2. 安装ICAgent。ICAgent是APM的采集代理，用于实时采集拓扑和调用链数据，您需要将其安装在应用所在服务器上。
3. 修改应用启动参数，以确保APM可以监控应用。
4. 在APM上管理应用，例如通过拓扑查看应用情况、通过事务了解用户体验情况等。

操作步骤

步骤1 将应用部署到华为云服务器。

1. 注册华为云账号并充值、购买Linux弹性云服务器，请参考[购买并登录Linux弹性云服务器](#)。
为保证应用性能，购买弹性云服务器时vCPU/内存请选择2核/4GB以上。应用只支持SUSE12 SP2操作系统，请在公共镜像中选择该镜像。
2. 登录弹性云服务器，创建应用目录并进入目录，本实践以目录mall为例。

3. 到[ORACLE官方网站](#)下载并安装应用需要的JRE 1.8版本。

4. 执行如下命令将应用下载至创建的mall目录并安装。

```
curl -L http://demos.obs.myhwclouds.com/demo_install.sh > demo_install.sh && bash demo_install.sh
```

步骤2 安装ICAgent，请参考[用户指南的安装ICAgent章节的首次安装](#)。

步骤3 修改应用启动参数，确保应用被APM监控。

修改应用的启动参数即在应用启动脚本增加下表参数：

参数	说明	本实践举例值
-javaagent	采集探针依赖的jar包，该参数值是固定的，均取值为： /opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar	/opt/oss/servicemgr/ICAgent/pinpoint/pinpoint-bootstrap.jar
-Dapm_application	应用名称，您可以结合您的应用自定义。	vmall
-Dapm_tier	应用层名称，指组成该应用的微服务名称。	本例中根据微服务名称，取值分别为apigw、product、user、dao

修改前后应用启动脚本对比如下：

修改前：

```
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 &
```

修改后（修改处已加粗表示）：

```
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 &
```

步骤4 修改完成后，便可重新启动应用。等待三分钟左右，在APM界面上即可看到应用相关数据。

后续可以在APM的概览、拓扑等界面中监控应用、定位应用的异常，具体可参考[APM的用户指南](#)。

----结束

2 定位事务异常原因

本文介绍使用事务定位应用异常。事务的主要功能是对应用服务端业务流的实时分析，通过Apdex直观体现用户对应用的满意度。您可以通过满意度情况定位应用异常或者性能瓶颈问题。

在使用APM管理电商应用最佳实践中，我们已经将APM接入了应用，本节我们继续以该应用为例，介绍如何使用事务功能定位问题。

操作流程

1. 发现异常。
2. 定位异常原因。

操作步骤

步骤1 发现异常。

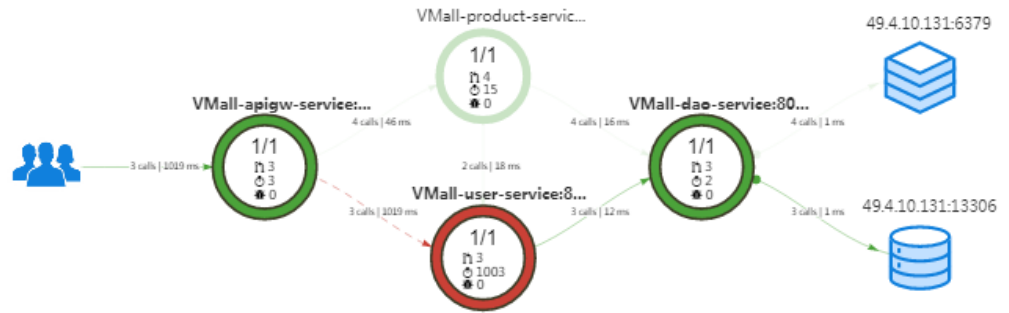
在使用APM进行应用巡检时，在事务界面中发现存在Apdex为红色的事务。

图 2-1 巡检发现异常事务

数据源	调用次数	整体时延(ms)	错误数	Apdex	Apdex阈值(ms)	当前Apdex阈值(m)	动作
事务名称: GET_/product/searchAll 事务别名: 未设置							
	30	29	0	1	501	501	
事务名称: POST_/product/buy/{id} 事务别名: 未设置							
	5	43	4	0.2	500	500	
事务名称: POST_/user/login 事务别名: 未设置							
	3	1023	0	0.5	500	500	

步骤2 定位异常原因。

1. 单击动作列的查看拓扑结构，在拓扑界面中查看实例具体情况，且通过颜色可初步确认异常微服务为user。



2. 右键单击user实例，选择“调用链搜索”，跳转到调用链页面。查找耗时最长的方法为下图所示，至此已定界到具体问题。

应用	方法	参数	状态	时间线(ms)	操作
vmall-apig...	org.apache.catalina.core.standardhostvalve.invoke	/user...	成功...	<1	扩展信息
vmall-apig...	org.springframework.web.servlet.framework...		成功...	<1	扩展信息
vmall-apig...	com.netflix.hystrix.hystrixcommand.queue		成功...	<1	扩展信息
vmall-apig...	com.netflix.hystrix.hystrixcommand...		成功...	<1	扩展信息
vmall-apig...	org.apache.http.impl.client.clo...		成功...	2	扩展信息
vmall-apig...	org.apache.http.protocol...	/user...	成功...	<1	扩展信息
vmall-user...	org.apache.catalina...	/user...	成功...	<1	扩展信息
vmall-user...	org.springframe...		成功...	2	扩展信息
vmall-user...	com.huawei.cl...		成功...	1001	扩展信息

----结束

3 使用拓扑定位异常

背景信息

在外部请求激增、负载突变等场景下，极易出现应用性能问题，比如外部请求响应变慢、部分请求异常等。快速识别发现、定位处理应用性能问题成为越来越常见的日常运维场景。

APM作为云应用性能问题诊断服务，拥有强大的分析工具，通过拓扑图、调用链、事务分析可视化地展现应用状态、调用过程、用户对应用的各种操作，快速定位问题和改善性能瓶颈。

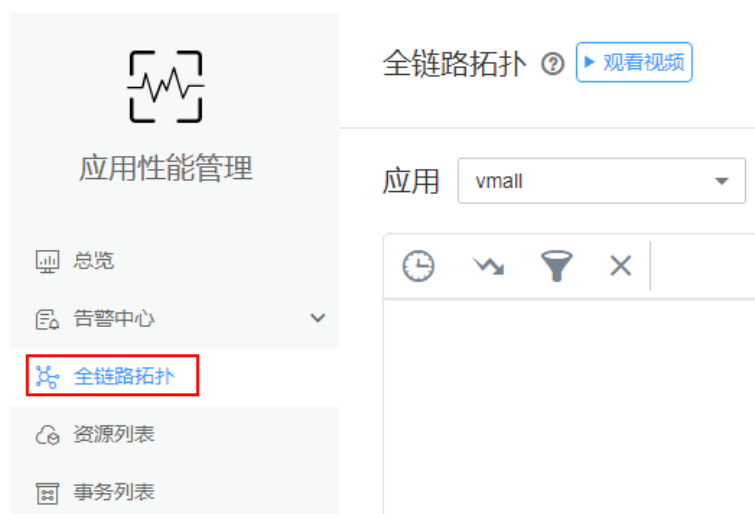
例如，通过APM拓扑功能可视化服务间的调用关系，迅速找到有问题的实例；通过APM调用链功能下钻到服务内部，抓取到有问题的方法调用链路，确认问题根因。

适用场景

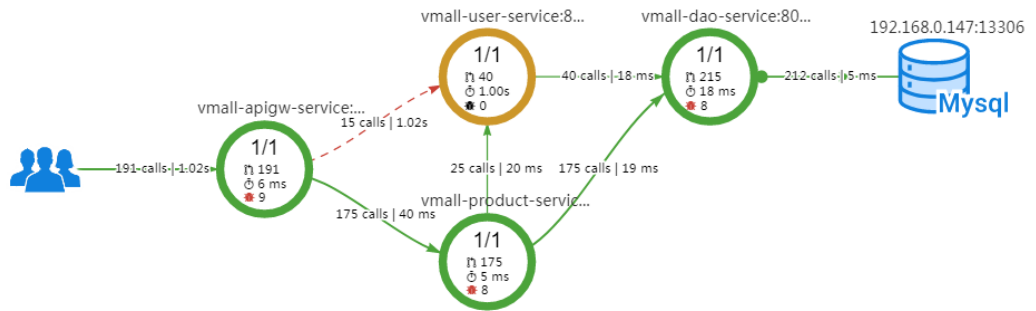
- 应用日常巡检，监控应用时延、吞吐量、错误数等性能指标。
- 应用异常调用快速定位。

操作步骤

步骤1 登录应用性能管理控制台。选择左侧导航中的“全链路拓扑”，进入APM拓扑界面。



步骤2 查看拓扑数据图中各指标、业务拓扑关系、TOP统计信息等。



步骤3 右键单击有问题的调用链，并选择“调用链搜索”。



步骤4 进入调用链界面，查看失败/高时延调用链，点击操作列的“查看调用关系”。

步骤5 在“调用关系”页面获取错误调用信息，确定问题根因。例如点击“详情”获取异常详细信息。

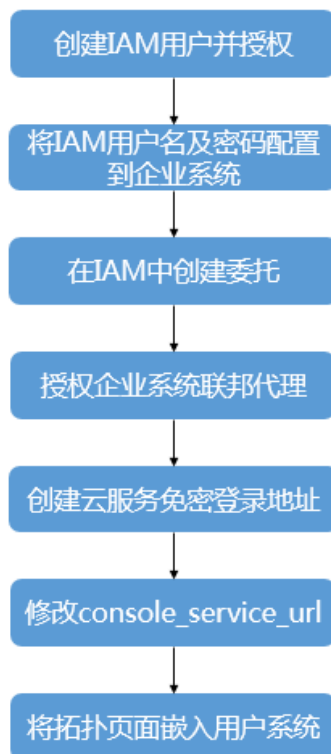
----结束

4 自建系统内嵌 APM 拓扑页面

背景信息

APM支持将拓扑界面嵌入到客户自建系统。通过IAM服务的联邦代理机制实现用户自定义身份代理，再将登录链接嵌入至客户自建系统实现无需在华为云官网登录就可在自建系统界面查看拓扑图。

操作流程



操作步骤

操作步骤1-5为用户自定义创建身份代理并创建登录地址FederationProxyUrl，与IAM服务强相关，这里只做了简单的介绍，详细可参考IAM服务的文档“[创建自定义身份代理](#)”。在IAM服务侧的操作完成后，您需要执行步骤6-7实现拓扑页面的内嵌。

步骤1 创建账号DomainA下的IAM用户userB，并授予Security Administrator和Agent Operator权限（全局服务-全局项目）。

📖 说明

DomainA和userB为示例名称，请以实际为准。

步骤2 将userB的用户名和密码配置到企业系统的配置文件中，密码建议加密存储，以便获取用户认证token和调用API。

步骤3 在统一身份认证服务创建委托并赋予您需要委托的权限。

步骤4 在企业系统建立和华为云账号“DomainA”下委托同名的用户组，将企业本地用户按需加入到不同的本地用户组，授予其联邦代理登录华为云时的权限。

步骤5 用户登录企业系统并访问企业系统的联邦代理，企业系统联邦代理将会创建云服务登录地址FederationProxyUrl。

FederationProxyUrl示例如下：

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

步骤6 将FederationProxyUrl中的console_service_url修改为云服务console地址。

console_service_url示例：

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

表 4-1 console_service_url 参数说明

参数名称	说明
region	表示当前您所在的区域，可在华为云官网登录云服务后在浏览器的地址栏中获取。例如cn-north-1
locale	语言标示，例如zh-cn
inFrame	内嵌页面标志，true表示隐藏华为云console页面页头页脚和菜单栏。

步骤7 使用iframe将apm拓扑页面嵌入用户系统，示例代码如下：

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

----结束