

应用性能监控

操作指南





【版权声明】

©2013-2025 腾讯云版权所有

本文档(含所有文字、数据、图片等内容)完整的著作权归腾讯云计算(北京)有限责任公司单独所有,未经腾讯云事先明确书面许 可,任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯,腾讯云将 依法采取措施追究法律责任。

【商标声明】

🔗 腾讯云

及其它腾讯云服务相关的商标均为腾讯云计算(北京)有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标,依法由权利 人所有。未经腾讯云及有关权利人书面许可,任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为,否则将 构成对腾讯云及有关权利人商标权的侵犯,腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况,部分产品、服务的内容可能不时有所调整。 您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定,除非双方另有约定,否则,腾讯云对本文档内 容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务,及相应的技术售后服务,任何问题请联系 4009100100或95716。

🔗 腾讯云

文档目录

操作指南 资源管理 新建业务系统 用量分析 应用监控 应用列表 应用详情 接口监控 异常监控 容器监控 应用诊断 性能剖析 线程池分析 连接池分析 GC 日志分析 链路追踪 链路追踪 链路详情 数据库调用监控 数据库概览 数据库分析 访问管理 概述 策略语法 策略授予 资源标签 告警服务 新建告警 查看告警 系统配置 业务系统配置 应用配置 采样配置 Prometheus 集成 关联日志 在链路详情中查询关联日志 在日志中注入 TraceID OpenTelementry 增强版 Java 探针 Skywalking 协议 (Java)

操作指南

资源管理

新建业务系统

最近更新时间: 2025-02-13 11:01:32

业务系统用于分类管理您的应用,您可以根据不同业务系统设置不同的存储时长、上报限额等。本文将为您介绍如何新建业务系统。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 资源管理**,选择需要新建业务系统的地域。
- 3. 在**业务系统管理**页,单击**新建**,进入创建业务系统弹框。
- 4. 根据下表配置系统信息。

配置项	说明
业务系统名 称	自定义业务系统名称。
开启免费模 式	开启免费模式后,该业务系统将永久免费。关于免费模式的限制,请参见关于免费模式的使用限制。
计费模式	支持 按量付费 和 预付费 。
上报地域	各地域数据隔离,业务系统创建后不可更改。
链路存储时 长	支持选择1天、3天、7天、15天、30天链路数据存储时长,试用期间默认存储时长为1天。存储时长越长, 收费越高。超过存储时长的链路将不会展示在应用性能监控控制台。
业务系统简 介	可以简单描述业务系统用途等。
添加标签	应用性能监控结合腾讯云资源标签功能,为您提供按标签授予子账号权限和按标签分账功能。请参见 <mark>访问</mark> <mark>管理</mark> 设置标签。

5. 配置完成后,勾选服务条款,然后单击**确定**即可。



用量分析

最近更新时间: 2025-02-14 15:41:22

通过**用量分析**页面,您可以分析每个业务系统产生的 APM 用量,以更好的评估 APM 费用。 APM 用量通过 **Span 上报数(条)、Span 存储量(条*天)、探针在线时长(探针个数*小时)**三个维度体现。为了方便评估 APM 费用,系统基于 APM 的真实使用量进行统计,不论采用哪种计费模式,接入 APM 的应用都会同时产生三个维度的数据。

 说明: 当天的用量统计会有1小时左右的延迟。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 资源管理**。
- 3. 在资源管理页面,选择用量分析,在此页面可查询最近30天内的 APM 用量。



应用监控 应用列表

最近更新时间: 2024-11-25 10:30:33

本文将为您介绍如何使用应用列表。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 应用列表**。

车 三 列表 应用过滤		(2					接入应用 告
立用名称/ID	应用状态(‡	吞吐量 ↓	平均响应时间(🛟	平均错误率/错误数 🛈 🛟	Apdex (i) ‡	组件漏洞数/严	标签	操作
ava-ma	●健康	4.52qps ↓ 4.1%	2.64ms ↓ 18%	0%↓- 0↑↓-	1↓-	4/4/0	\bigtriangledown	清理应用 编辑标签
9 V.	●警示	3.56qps ↑ 100%	504.14ms † 100%	0%↓- 0个↓-	0.99 † 100%	4/4/0	\bigtriangledown	清理应用 编辑标签
v	●健康	3.21qps ↓ 4.2%	1.18ms ↓ 15.5%	0%↓- 0个↓-	1↓-	0	0	清理应用 编辑标签
	●警示	3.2qps ↓ 5.3%	1135.3ms ↓ 8.4%	4.79% ↓ -11.7% 138↑ ↓ -16.4%	0.79 † 1.3%	4/4/0	0	清理应用 编辑标签
ي s- ع	●健康	2.97qps ↓ 3.7%	2.88ms ↓ 28.8%	0%↓- 0个↓-	1↓-	4/4/0	0	清理应用 编辑标签
ع	●健康	1.07qps ↓ 4.2%	0.82ms ↑ 0.3%	0%↓- 0↑↓-	1↓-	0	0	清理应用 编辑标签
	●健康	1.07qps ↓ 4.2%	3.8ms↓1.4%	0%↓- 0个↓-	1↓-	0	\bigtriangledown	清理应用 编辑标签

功能说明

- 应用过滤:可以基于应用名、应用 ID 和应用标签进行过滤。
- **清理应用:** 清理应用功能仅针对已经不再上报监控数据的应用,请先确保该应用的探针已经成功卸载(如果通过代码上报,请删除 相关逻辑)。在执行完清理操作后,如果 APM 收到监控数据上报,会再次根据应用名创建应用。
- 编辑标签:通过该功能,可以为应用关联多个标签,用于应用查询以及细粒度的权限配置。标签键和标签值引用自腾讯云统一的标签中心,如果需要维护标签,请前往标签列表。



应用详情

最近更新时间: 2024-11-25 10:30:33

本文将为您介绍如何查看应用详情,了解应用拓扑、请求数、响应时间、错误数、吞吐量等信息。

操作步骤

1. 登录 腾讯云可观测平台。

- 2. 在左侧菜单栏中选择**应用性能监控 > 应用列表**。
- 3. 单击具体的应用名称,进入应用详情页面。

模块说明

性能概览

性能概览模块为您展示了所服务端或客户端、某时间段内应用平均响应时间、平均吞吐量、平均错误率以及 Apdex 的变化趋势。通过 单击每个图表右上角的时钟图标,您可以自定义30天内任意日期的数据与当前时间段进行对比分析。

应用拓扑

应用拓扑当前以筛选服务为中心,展示上下游局部依赖拓扑。将鼠标悬浮在代表应用的节点上,您可以看到对应应用平均吞吐量,响应 时间,错误率。

应用性能监控使用不同的拓扑图标颜色进行标识,绿色图表示应用健康、橙色图标表示应用有延时情况、红色表示该业务出现异常情况。

JVM 监控

该模块用于展示重要的 JVM 指标变化趋势,包括 GC(Garbage Collection)平均/最大次数、CPU 利用率、Heap 空间、 NoHeap 空间、Heap 空间细化和 JVM 线程数等。

接口监控

该模块展示应用接口性能指标。您还可以单击接口名查看该接口的更多详情。

数据库调用分析

该模块展示应用访问数据库的情况,由类型、数据库名称、数据库地址组成的三元组,可以确定唯一的数据库。您还可以单击**查看概览** 和**查看 SQL** 查看更多数据库调用情况。

SQL 分析

该模块展示应用访问数据库时每一种调用语句的执行情况。您还可以单击查看调用查看相关调用链的执行情况。

实例监控

该模块展示每一个应用实例的性能指标。

链路追踪

该模块展示该应用相关调用链路,包括 Trace 和 Span 信息。您还可以单击 Trace ID 查看链路详情。

错误监控



该模块展示该应用相关异常信息,包括对每种异常类型的统计。

容器监控

对于部署在容器服务 TKE 的应用,该页面可以展示应用所关联的 Deployment 的监控信息。如果应用通过 tencentopentelemetry-operator 接入,不需要进行额外的配置;如果应用通过其他方式接入,请参见 自定义应用实例属性 设置必要的实 例属性字段。



接口监控

最近更新时间: 2024-10-14 20:29:52

接口监控展示客户端调用、服务端调用和本地调用中的接口、链路上游和链路下游的接口调用情况。包括请求次数、平均响应时间、错 误率、吞吐量等接口调用关键指标。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 应用监控 > 接口监控**。
- 3. 进入接口监控页面,即可查看相关模块信息。



模块说明

接口总览

在接口监控页面,用户在左侧选择接口,右侧会展示出该接口对应的接口分析和异常分析,可在顶部选择调用者进一步查看该调用者的 详情。

接口分析



接口分析可以选择接口对应的服务,查看该服务的总吞吐量和 TOP5 调用者、平均响应时间(以及99pct,90pct,50pct)、错误 率和错误代码分解。

指标名称	说明
TOP5 调用者	查看调用所选应用接口频率最高的5个上游应用/组件
错误代码分解	查看当前接口返回的错误代码分布趋势

() 说明:

- 99pct: 升序排列后排在99%位置的数据。
- 90pct: 升序排列后排在90%位置的数据。
- 50pct: 升序排列后排在50%位置的数据。

异常统计

除基本指标外,异常分析模块帮您智能筛出了当前应用平均响应时间和错误率最高的 TOP5 接口。您可以将鼠标移动到曲线上方单击 **查看详情**,查看选中时间点向前回溯15分钟里,途径该接口的调用链,一键下钻,完成故障排查。

上下游分析

您可以在子窗口切换上下游分析菜单,分析上下游调用情况,快速排查性能瓶颈。





异常监控

最近更新时间: 2024-10-12 15:21:32

本文将为您介绍如何查看应用异常情况,进行异常分析。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 应用监控 > 异常监控**。
- 3. 进入异常监控页面,即可查看相关模块信息。



模块说明

异常列表

在异常监控页面左侧所选时间内的所有异常,包括服务异常类型、接口、异常发生次数等信息。

异常趋势

右侧图表会显示出异常趋势所筛选时间粒度下异常的发生次数。单击曲线上的数据点可进行下钻分析异常情况。

异常分析

单击相关应用,在异常监控页面右侧将会展示该异常的异常次数时序曲线,以及昨日同时间段对比曲线。

🕛 说明:



您可以单击图表上方的**对比线选择**,添加一个月内特定日期的同比曲线。还可以单击曲线上的任意数据点,查看相关请求列 表。在列表中单击**请求详情**,可以下钻到调用链路详情中分析异常。





容器监控

最近更新时间: 2025-02-21 09:16:01

对于部署在容器服务 TKE 的应用,可以关联查询容器监控数据,包括事件和各类监控指标。

() 说明:

- 容器监控数据从容器服务 TKE 获取,暂不支持自建 Kubernetes 集群。
- 应用所在的 TKE 集群必须是当前主账号的资源,因此该功能暂不支持跨账号上报场景。
- 目前仅支持通过 OpenTelemetry 方案接入的应用,暂不支持 Skywalking 方案。

前提条件

对于 TKE 环境通过 tencent-opentelemetry-operator 自动接入的应用,可以直接使用容器监控功能。使用其他方式接入的应 用,需要通过实例属性填入必要的 TKE 环境信息。必要的环境信息包括:

字段名	说明
k8s.region	TKE 集群所在地域,例如 ap-guangzhou,详情请参见 CVM 地域和可用区 的取值
k8s.cluster.id	TKE 集群 ID
k8s.deployment.name	应用所在的 Deployment
k8s.namespace.name	应用所在的命名空间
k8s.pod.name	应用所在的 Pod 的名字,该变量在 Pod 创建后才能确定,可以通过 Downward API 获取

关于自定义应用实例属性,请参见 自定义应用实例属性。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 在左侧菜单栏中选择应用性能监控 > 应用列表,单击应用的名称/ID,选择容器监控。
 在容器监控页面将展示与该应用关联的工作负载,您可以查看该工作负载的监控图表。如果该应用的多个实例分布在不同的工作负载中,请在下拉框中选择需要监控的工作负载。



应用诊断 性能剖析

最近更新时间: 2024-11-05 12:14:22

性能剖析能力基于 async-profiler 技术实现,以极低的性能开销,生成性能剖析火焰图,帮助用户直观的分析 CPU/内存飙升的原 因,快速定位应用性能瓶颈。相比直接使用社区的 async-profiler 方案,应用性能监控提供的性能剖析能力免去了工具的安装、命令 执行、剖析结果下载等复杂的操作,大幅提升了排查应用性能问题的效率。

根据反复性能测试的结果,对一个典型的微服务应用执行性能剖析采集,采集过程中的 CPU 开销在5%以下,内存开销在50M 以下, 对于TPS和响应耗时基本没有影响。

操作前提

- 应用使用腾讯云增强版 Java 探针接入,探针版本 1.16-2024030510 及以上。
- 运行环境需要安装 OpenJDK 或其他基于 HotSpot JVM 开发的 JDK,暂不支持仅安装了 JRE 的环境。
- 支持 Linux(x64, arm64) 操作系统, 暂不支持 macOS 和 Windows。
- 不推荐使用 Java 8u352 以下的 JDK,可能存在 内存崩溃 风险。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择点击**应用性能监控 > 应用诊断 > 性能剖析**页面。
- 3. 在页面左侧的实例列表中找到需要进行性能剖析的实例,点击采集。
- 4. 在弹出对话框中,选择数据采集时长和采集类型,点击确认。
- 5. 采集任务提交后,可以在页面右侧的性能剖析记录中查询剖析结果,当采集状态为**采集完成**时, 点击**查看性能剖析**,在弹出页面查 看性能剖析火焰图。

选项名	说明
数据采集时 长	从应用实例收到采集任务开始,采集剖析数据的时间长度。目前支持5秒、10秒、30秒、3分钟、5分钟。
采集类型	采集类型决定了剖析数据代表的性能指标,目前支持 CPU、耗时、内存三种采集类型。 • CPU: 代表 CPU 执行代码块所花费的时间。 • 耗时: 耗时基于墙钟时间(Wall Clock)进行统计,从进入到退出方法所经过的实际时间。所有等待时 间、锁定和线程同步的时间都包括在内,所以墙钟时间不会短于 CPU 时间。 • 内存: 代表内存分配。

性能剖析火焰图使用说明

性能剖析基于抽样获取剖析数据,性能指标通过函数在抽样中出现的频率来体现,并不是绝对值,所以需要重点关注方法(函数)之间 的相对占比。在火焰图中,Y 轴代表方法(函数)栈的深度,X 轴代表被抽样的次数。以 CPU 时间为例,一个方法(函数)的宽度越 宽,CPU 执行代码块所花费的时间就越长。

在分析火焰图的时候,建议从深度最深的方法(函数)开始分析。栈最深,出现在火焰图中的位置会越靠近下方,也就是"火苗"的位 置(不同于自然界的火焰,在 APM 目前使用的火焰图中,火焰朝向下方)。

▲ 注意:



火苗越宽,代表性能上的消耗越大,所以宽火苗往往是引发性能问题的根源。





线程池分析

最近更新时间: 2025-02-28 15:16:43

线程池分析功能可以展示应用核心线程池的各项关键指标,包括最大线程数、核心线程数、活跃线程数、线程池任务数、线程池大小 等。

支持的线程池

腾讯云增强版 Java 探针支持线程池分析功能。

腾讯云增强版 Java 探针1.16-2024030510及以上版本

- Apache Dubbo 2.7+
- Apache Tomcat 7.0+

腾讯云增强版 Java 探针2.3-20250131及以上版本

OKhttp 2.x/3.x/4.x

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择**应用性能监控 > 应用诊断**,然后选择**线程池分析**。
- 3. 在页面左侧的实例列表中单击需要进行池程池分析的实例,在页面右侧图表即可查看线程池分析详情。

指标明细

- 最大线程数: 对应 ThreadPoolExecutor.getMaximumPoolSize()
- 核心线程数: 对应 ThreadPoolExecutor.getCorePoolSize()
- 活跃线程数: 对应 ThreadPoolExecutor.getActiveCount()
- 线程池大小: 对应 ThreadPoolExecutor.getPoolSize()
- 线程池任务数: 对应 ThreadPoolExecutor.getQueue().size()
- 线程池使用率: 活跃线程数/最大线程数*100%



连接池分析

最近更新时间: 2025-02-28 15:16:43

连接池分析功能可以展示应用核心连接池的各项关键指标,包括当前连接数、最大连接数、活跃连接数、空闲连接数、等待连接数等。

支持的连接池

腾讯云增强版 Java 探针支持连接池分析功能。

腾讯云增强版 Java 探针1.16-2024030510及以上版本

Apache Druid 1.0+

腾讯云增强版 Java 探针2.3-20250131及以上版本

- HikariCP 3.0+
- Redisson 3.0+
- Jedis 3.0+
- Letuce 5.0+
- Apache HttpClient 4.4+
- OkHttp 2.x/3.x/4.x

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择**应用性能监控 > 应用诊断**,然后选择**连接池分析**。
- 3. 在页面左侧的实例列表中单击需要进行连接池分析的实例,在页面右侧图表查看连接池分析详情。

指标明细

以 Druid 连接池为例,APM 会采集如下指标:

- 当前连接数:对应 DruidDataSource.getPoolingCount() + DruidDataSource.getActiveCount()
- 最大连接数: 对应 DruidDataSource.getMaxActive()
- 活跃连接数:对应 DruidDataSource.getActiveCount()
- 空闲连接数:对应 DruidDataSource.getPoolingCount()
- 等待连接数:对应 DruidDataSource.getWaitThreadCount()

对于其他连接池,APM 会按照类似的方式采集连接数指标。由于实现机制的差异,部分连接池可能不会上报等待连接数指标。

⁽⁾ 说明:

GC 日志分析

最近更新时间: 2024-09-27 16:03:01

GC 日志分析功能基于 JVM 输出的 GC Log 排查有可能影响应用性能的潜在风险。

操作前提

- 腾讯云增强版 Java 探针 2.3-20240831 以上版本支持 GC 日志分析功能。
- 在使用此功能前,先确保 JVM 已经配置了打印 GC 日志相关的启动参数。

打印 GC 日志相关的启动参数

Java 8

启动参数示例如下:

```
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintHeapAtGC -
Xloggc:gc.log -XX:+UseGCLogFileRotation -XX:GCLogFileSize=20M -XX:NumberOfGCLogFiles=5
```

必填参数:

```
-XX:+PrintGCDetails -XX:+PrintGCDateStamps -XX:+PrintGCTimeStamps -XX:+PrintHeapAtGC -Xloggc
```

Java 9 及以上版本

启动参数示例如下:

-Xlog:gc*:file=gc_%p_%t.log:time,pid:filecount=5,filesize=20M

或者直接指定GC日志绝对路径:

```
-Xlog:gc*:file=/path/to/gc.log:time,pid:filecount=5,filesize=20M
```

其中:file 为必填参数,用于指定 GC 日志文件路径。

```
() 说明:
```

更多详细参数设置,请参考 Java 官方文档。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择点击**应用性能监控 > 应用诊断 > GC日志分析**页面。
- 在页面左侧的实例列表中找到需要进行 GC 日志分析的实例,点击分析,在弹出对话框中选择数据时长,数据时长代表从 GC 日志 中向回溯的时间跨度,然后点击确认。
- 分析总耗时为数秒至5分钟左右,取决于当前选择数据时长内所包含的 GC 日志内容大小,应用性能监控最多可以分析100MB 的 GC 日志数据,超出部分将被截断。
- 5. 在页面右侧表格中能够查到历史分析记录,分析完成的记录将展示为**采集完成**状态,点击**查看报告**获取分析结果。

分析报告解读

分析报告中包含三部分内容:

- JVM 信息:目前仅支持在 JDK 8 中展示,记录 JVM 版本、启动参数、系统属性等信息。
- GC 总览(Summary):在选定时间段有 GC 行为时会展示。需要重点关注日志起始时间、GC 事件总数、GC 吞吐量等信息。
 GC 吞吐量是衡量 Java 垃圾回收器性能的重要指标,可以尝试不同的垃圾回收器,或者调整垃圾回收器的相关参数,以获取更低的 GC 次数以及更高的 GC 吞吐量。

分析结果(Analysis):在选定时间段有 GC 行为时会展示。分析结果总结了 GC 方面可能存在的问题,并给出了问题的原因以及和解决建议,是分析报告的核心内容,需要重点关注结果分析中的 error 和 warn 部分。

参考如下分析报告:

腾讯云

	Report
	gc-example.log
	JVM:
1. JVM 相关	<pre></pre>
	SUMMARY:
	Datestamp First: 2016-10-10T18:43:49.025-0700 Timestamp First: 1.362 secs Datestamp Last: 2016-10-11T12:34:47.720-0700 Timestamp Last: 64260.057 secs # GC Events: 36586 Event Types: PAR_NEW, CMS_INITIAL_MARK, CMS_CONCURRENT, CMS_REMARK, CMS_SERIAL_OLD
2.GC 总览	<pre># Parallel Events: 36585 # Inverted Parallelism: 2 Inverted Parallelism Max: 2016-10-11T08:06:39.037-0700: 48171.374: [GC (CMS Initial Mark) [1 Cl # Serial Events: 1 NewRatio: 54 Heap Used Max: 7092037K Heap After GC Max: 6988066K Heap Allocation Max: 8371584K</pre>
	Metaspace Used Max: 167164K Metaspace After GC Max: 167164K Metaspace Allocation Max: 1204224K GC Throughput: 96% GC Pause Max: 7.528 secs GC Pause Total: 2626.403 secs
	ANALYSIS:
	error
	*The CMS_SERIAL_OLD collector is being invoked for one of the following reasons: (1) Fragmenta *CMS promotion failed. A young generation collection is not able to complete because there is
	warn
3.结果分析	*CMS remark low parallelism: (1) If using JDK7 or earlier, add -XX:+CMSParallelRemarkEnabled, *CMS initial mark low parallelism: (1) If using JDK6 or earlier, initial mark is single-thread *Application stopped time missing. Enable with -XX:+PrintGCApplicationStoppedTime (<= JDK8) or *Inverted parallelism. With parallel (multi-threaded) collector events, the "user" + "sys" tim *GCLocker GC due to the following sequence of events: (1) An object allocation failed due to n *Explicit garbage collection is disabled with -XX:+DisableExplicitGC. The JVM uses explicit ga *Consider enabling gc log file rotation (-XX:+UseGCLogFileRotation -XX:GCLogFileSize=N[K M G]
	info
	<pre>*The JDK is very old (7.6 years). *Metaspace(unlimited) = Class Metadata(unlimited) + Compressed Class Space(1024M). *-XX:+UseParNewGC is redundant (enabled by default) and can be removed. Deprecated in JDK8 and *The number of times an object is copied between survivor spaces is set with -XX:MaxTenuringTh *The number of parallel garbage collection threads is set with -XX:ParallelGCThreads=N. Unless *Consider enabling large page support, which can provide performance improvements with large h</pre>
	*Consider enabling large page support, which can provide performance improvements with large

GC 总览 (Summary)

• GC 事件总数 (GC Events): 36586次。该数字偏高,可能意味着应用程序创建了大量的短暂对象,这可能会导致频繁的垃圾 回收。

- 🕥 腾讯云
 - GC 事件类型(Event Types):该时间内产生了PAR_NEW、CMS_INITIAL_MARK、CMS_CONCURRENT、 CMS_REMARK、CMS_SERIAL_OLD 这几种类型的 GC。
 - 并行 GC 事件 (Parallel Events): 36585次。
 - 串行 GC 事件 (Serial Events): 1次。
 - 最大堆使用量 (Heap Used Max): 7092037 K。
 - GC 后最大堆使用量(Heap After GC Max): 6988066 K。这个数字如果接近最大堆使用量(Heap Used Max),可能 意味着应用程序的内存使用效率不高,或者存在内存泄漏。
 - 最大堆分配量(Heap Allocation Max): 8371584 K。
 - 元空间最大使用量(Metaspace Used Max): 167164 K。
 - GC 后元空间最大使用量 (Metaspace After GC Max): 167164 K。
 - 最大元空间分配量 (Metaspace Allocation Max): 1204224 K。
 - GC 吞吐量(GC Throughput): 96%。这意味着应用程序在96%的时间内在执行实际的业务逻辑,而在4%的时间内在进行垃圾回收。GC 吞吐量是一个衡量 Java 垃圾回收器性能的指标,它表示的是应用程序运行的时间占总运行时间的百分比。
 - 最大 GC 暂停时间(GC Pause Max): 7.528秒。如果单次 GC 的暂停时间过长,那么可能会影响应用程序的响应时间和延迟。例如,如果应用程序需要在1000毫秒内完成交易,那么任何一次 GC 暂停超过1000毫秒都是不可接受的。优化的方法可能包括使用并发垃圾回收器(如 G1 或 CMS),这些垃圾回收器可以在应用程序运行的同时进行垃圾回收,从而减少 GC 暂停时间。

分析结果(Analysis)

分析结果(Analysis)总结了 GC 方面可能存在的问题,并给出了问题的原因以及和解决建议,需要重点关注结果分析中的 **error** 和 warn 部分。例如,在如下示例中,明确指出了 CMS_SERIAL_OLD 垃圾回收器是串行运行的,回收大内存的时候可能会需要非常 长的时间,建议通过调整 JVM 参数避免使用串行垃圾回收器。您可以基于分析结果对 JVM 进行调优,并通过应用性能监控的**实例监** 控等功能对比调优后的效果。

> 腾讯云

ANALYSIS:	
error	

*The CMS_SERIAL_OLD collector is being invoked for one of the following reasons: (1) Fragmentation. The concurrent low pause collector does not compact. When fragmentation becomes an issue a serial collection compacts the heap. If the old generation has available space, the cause is likely fragmentation. Fragmentation can be avoided by increasing the heap size. (2) Metaspace class metadata or compressed class pointers allocation failure. The GC attempts to free/resize metaspace. (3) Resizing perm gen. If perm gen occupancy is near perm gen allocation, the cause is likely perm gen. Perm gen resizing can be avoided by setting the minimum perm gen size equal to the the maximum perm gen size. For example: -XX:PermSize=256M -XX:MaxPermSize=256M. (4) Undetermined reasons. Possibly the JVM requires a certain amount of heap or combination of resources that is not being met, and consequently the concurrent low pause collector is not used despite being specified with the -XX:+UseConcMarkSweepGC option. The CMS_SERIAL_OLD collector is a serial (single-threaded) collector, which means it can take a very long time to collect a large heap. For optimal performance, tune to avoid serial collections.

*CMS promotion failed. A young generation collection is not able to complete because there is not enough space in the old generation for promotion. The old generation has available space, but it is not contiguous. When fragmentation is an issue, the concurrent low pause collector invokes a slow (single-threaded) serial collector to compact the heap. Tune to avoid fragmentation: (1) Increase the heap size. (2) Use -XX:CMSInitiatingOccupancyFraction=N (default 92) to run the CMS cycle more frequently to increase sweeping of dead objects in the old generation to free lists (e.g. -XX:CMSInitiatingOccupancyFraction=85 -XX:+UseCMSInitiatingOccupancyOnly). (3) Do heap dump analysis to determine if there is unintended object retention that can be addressed to decrease heap demands. Or move to a collector that handles fragmentation more efficiently: (1) G1 compacts the young and old generations during evacuation using a multi-threaded collector. (2) Shenandoah compacts concurrently. Temporarily add -XX:PrintFLSStatistics=1 and -XX:+PrintPromotionFailure to get

additional insight into fragmentation.



链路追踪 链路追踪

最近更新时间: 2025-02-18 09:22:22

链路追踪是应用性能监控(APM)的重要能力,可以实现跨应用的多维度调用链检索与分析。在链路追踪主页面,您可以根据多种过 滤条件组合进行调用查询,查询结果中的每一条记录代表对一次调用的记录,等同于一个 Span。您可以单击对应的 TraceID 进入链 路详情视图,进一步分析链路中的每个环节。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 链路追踪**。
- 3. 在链路追踪页面,选择合适的地域以及业务系统。
- 4. 通过右上方的时间选择器指定查询时间跨度,通过查询对话框指定更多查询条件。

字段名	说明
调用角色	调用角色等同于 Span 类型,表示这条 Span 在链路中被哪一种身份记录,包括 Server、Client、 Consumer、Producer、Internal 5种类型,分别代表服务端、客户端、消费者、生产者,以及内部 调用。 假设应用 A 向应用 B 发起了一次 HTTP 调用,应用 A 会记录一个调用角色为 Client 的 Span,应用 B 会记录一个调用角色为 Server 的 Span。关于 Span 类型的更多详情,请参见 OpenTelemetry 对 Span 类型的定义。
是否为链路入 口	链路入口代表一条链路中的第一个 Span,通常情况下,Trace ID 由这个 Span 生成。
应用名称	应用名称通常和服务名保持一致。应用名称在应用接入 APM 时指定,多个使用相同应用名称接入的进 程,在 APM 中会表现为相同应用下的多个实例。
实例	实例名称是每个接入 APM 的进程的唯一标识。通常情况下,实例名称是进程运行环境的 IP 地址。部分 接入方案可以自动设置实例名称,也可以在接入 APM 时指定,具体的设置方式请参考接入文档。
对端服务	用于标识该调用的对端服务,例如,当一个 Span 的调用角色为 Server 的时候,通常会对应另一个调用角色为 Client 的 Span,那个 Span 所在的应用,就是对端服务。如果使用腾讯云增强版 Java 探针接入,APM 会在 Span 中自动注入对端服务。如果使用其他 OpenTelemetry 方案接入,APM 也会根据调用关系尽可能的在 Span 中注入对端服务。如果对端服务未能自动注入,您可以在上报到 APM 的 Span 中通过 peer.service 属性手动指定对端服务。
接口	接口名称等同于 Span 名称。
埋点组件	埋点组件用来标识上报该 Span 的框架或组件。腾讯云增强版 Java 探针和部分开源 OpenTelemetry 探针能够自动注入埋点组件。如果埋点组件未能自动注入,您可以在上报到 APM 的 Span 中通过 component 属性手动指定埋点组件。
状态	表示该调用是否正确。
错误类型	如果一个调用的状态为错误,错误类型用于对不同种类的错误进行区分。



包含方法栈快 照的调用	如果某个接口调用耗时频繁超过慢调用监听阈值,该接口产生的调用将有机会自动生成方法栈快照。对于 包含了方法栈快照的调用,可以在链路详情中查看本地方法栈的执行详情。
	① 说明: 腾讯云增强版 Java 探针1.16-2023102808及以上版本支持该功能。
存在 SQL 注 入风险的调用	在开启应用安全后,系统可以识别存在 SQL 注入风险的数据库调用。
	① 说明: 腾讯云增强版 Java 探针2.3−20241130及以上版本支持该功能。
Span 属性	表示在 Span 中附加的其他信息,用键值对的方式表达,除了探针自动注入的 Span 属性外,用户也可 以在代码中手动添加 Span 属性。

在列表中添加自定义展示列

在链路追踪列表中,可以根据实际业务需求,针对 Span 的任何属性,增加自定义列,在链路追踪结果中展示。请先前往 应用性能监 控 > 系统配置 > 业务系统配置 页面,在**自定义配置**中添加需要展示的 Span 属性,这样就能通过单击 链路追踪 页面右侧的设置图 标,勾选需要在列表中展示的字段。

视图功能

可以将常用的查询条件保存到视图中,方便后续使用。

- 1. 在查询对话框指定查询条件,然后单击确认。
- 2. 单击页面右侧的**保存视图**,可以将当前的查询条件保存到新的视图或者已有的视图中。在下次进入**链路追踪**页面时,保存视图左侧 将展示时钟图标,单击此图标,可选择并进入任意一个已经有的视图中,将载入之前保存过的查询条件。



链路详情

最近更新时间: 2025-02-20 14:11:12

链路详情功能用于分析链路中的每个环节,您可以通过瀑布视图洞察每一个环节的调用耗时以及执行状态,在链路详情页面中,所有数 据都来自于同一条链路,具有唯一的 Trace ID。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 链路追踪**。
- 3. 在页面顶部选择合适的地域以及业务系统。
- 4. 通过右上方的时间选择器指定查询时间跨度,通过查询对话框指定更多查询条件。
- 5. 每行查询结果的第一列展示了 Trace ID, 单击 Trace ID 对应的链接,即可以进入链路详情页面。

链路详情介绍

接口维度展示

单击页面右上方的接口维度展示,瀑布图将对展示内容进行精简,只有调用角色为 Server 或 Consumer 的 Span 会被展示。该展 示方式从服务端以及消费者的角度衡量应用性能,在调用关系比较复杂时,可以从链路中快速提取重要的信息。在使用接口维度展示的 情况下,可以在接口名称列点击绿色④符号,在弹出的对话框中查看该应用的所有调用。

							查看鸟瞰图	接口维度展示	全链路展示
占:	去对应调田可查看 印 细								
应用名称		接口名称	调用角色	实例		调用时间			
•	java-schedule-service	/callOrderService (+)	Server	1	8				45.218ms
	▼ java-order-service	GET /generateOrderInfo+	Server	1	109				42.099ms
	java-delivery-service	com.tencent.cloudmonitor.dubbo.api.IDeliveryS	Server	1	104	0.061ms			
	java-market-service	com.tencent.cloudmonitor.dubbo.api.lMarketSer	Server	1	71	0.044ms			
	java-user-service	user.UserService/GetLoginInfo($+$)	Server	1	58	1.192ms			
	java-user-service	user.UserService/GetAccountInfo+	Server	1	58	0.911 ms	3		
	▼ java-user-service	user.UserService/GetUserInfo+	Server	1	58	2.	266ms		
	go-member-service	/getMembers+	Server	1	60	0.75	54ms		
	▼ java-market-service	market.MarketService/GetProductInfo	Server	1	71		9.7	782ms	
	java-stock-service	stock.StockService/GetStockInfo	Server	1	88	•	8.16	6ms	
	▼ java-delivery-service	GET /getDeliveryInfo 🕂	Server	1	104	1	8.0	96ms	
	python-transport-ser	rvice /transport+	Server	1	89		3.922m	3	
	▼ java-market-service	market.MarketService/GetShopInfo+	Server	1	71		3.412ms		
	java-stock-service	${\tt stock.StockService/GetPreDayStockInfo} \textcircled{+}$	Server	1	88		0.751ms		
	java-stock-service	stock.StockService/GetStockDetailInfo+	Server	1	88		0.717ms		
	java-market-service	market.MarketService/GetCartInfo	Server	1	71			1.159ms	

全链路展示

单击页面右上方的**全链路展示**,瀑布图将展示该链路的所有 Span。通过全链路展示,可以更全面的了解链路中每个环节的细节。



					查看鸟瞰图 接口维度展示	全链路展示	
白光对放调田可查看的细							
应用名称	调用角色	实例	调用时间				
v java-schedule-service	/callOrderService	Server				45.218ms	
v java-schedule-service	StandardHostValve.invoke	Internal				45.107ms	
▼ java-schedule-service	ScheduleController.generateOrderInfo	Internal				45.036ms	
▼ java-schedule-service	ScheduleServiceImpl.generateOrderInfo	Internal				44.868ms	
▼ java-schedule-service	HTTP GET http://{IP}/generateOrderInfo	Client				42.477ms	
▼ java-order-service	GET /generateOrderInfo	Server				42.099ms	
v java-order	OrderGenerateController.generateOrderInfo	Internal				41.838ms	
v java-o	OrderServiceImpl.generateOrderInfo	Internal				41.624ms	
j	SETEX	Client		0.33ms			
v j	OrderServiceImpl.dubboInvoke	Internal		2.096ms			
•	com.tencent.cloudmonitor.dubbo.api.IDeliveryS	Client		0.74ms			
	com.tencent.cloudmonitor.dubbo.api.IDeliveryS	Server		0.061ms			
•	com.tencent.cloudmonitor.dubbo.api.IMarketSe	Client		0.717ms			
	com.tencent.cloudmonitor.dubbo.api.IMarketSe	Server		0.044ms			
▼ [j	OrderServiceImpl.getUseInfo	Internal		13.8	381ms		
•	user.UserService/GetLoginInfo	Client		1.79ms			
•	user.UserService/GetLoginInfo	Server		1.192ms			
	SELECT mock_project_db.mock_project_userinfo	Client		0.538ms			

查看鸟瞰图

单击页面右上方的**查看鸟瞰图**,页面上方将开启鸟瞰视图,体现链路中的关键环节的调用耗时。在调用关系比较复杂的时候,通过鸟瞰 图可以更快速的掌握链路的全貌。在鸟瞰视图开启的情况下,可以单击页面右上方的**收起鸟瞰图**关闭鸟瞰视图。



点击对应调用可查看明细						
应用名称	接口名称	调用角色	实例	调用时间		
v java-schedule-service	/callOrderService	Server		45.218ms		
▼ java-order-service	GET /generateOrderInfo	Server		42.099ms		
java-delivery-service	com.tencent.cloudmonitor.dubbo.api.IDeliveryS	Server		0.061ms		
java-market-service	com.tencent.cloudmonitor.dubbo.api.IMarketSe	Server		0.044ms		
java-user-service	user.UserService/GetLoginInfo+	Server		1.192ms		
java-user-service	user.UserService/GetAccountInfo	Server		0.911ms		
v java-user-service	user.UserService/GetUserInfo(+)	Server		2.266ms		
go-member-service	/getMembers(+)	Server		0.754ms		
v java-market-service	market.MarketService/GetProductInfo	Server		9.782ms		
java-stock-service	stock.StockService/GetStockInfo+	Server		8.166ms		
v java-delivery-service	GET /getDeliveryInfo	Server		8.096ms		
python-transport-service	/transport+	Server		3.922ms		

Span 详情



在瀑布图中单击任何一行,将展开 Span 详情视图,Span 中携带的所有信息都将在 Span 详情视图中体现。在标签页上方将展示 Span 名称、调用角色、开始时间、结束时间、实例名称、对端服务等最重要的 Span 信息。根据 Span 的不同特点,Span 详情视 图中还将包含如下标签页:

- 错误:如果该 Span 的状态为错误,此标签页将展示具体的错误信息。
- 调用:如果该 Span 代表一次 RPC 或 HTTP 调用,此标签页将展示具体的调用信息,这些信息通过前缀为 http 和 rpc 的 Span 属性来承载。
- SQL:如果该 Span 代表一次 SQL 调用,此标签页将展示具体的调用信息,这些信息通过前缀为 db 的 Span 属性来承载。
- 事件:如果该 Span 中包含事件,此标签页将展示具体的事件信息。
- 更多 Span 信息:此标签页将补充展示所有 Span 信息,包括在 Span 中上报的所有自定义字段。
- CLS 日志:如果该应用关联了 CLS 日志服务,此标签页将根据 Trace ID 和 Span ID 查询并展示和该调用相关的日志,请确保 关联的 CLS 主题开启了全文索引。

如果该 Span 包含了方法栈快照,在 Span 详情中单击**方法栈快照**,将展示本地方法栈中每个方法的执行耗时,详情请参见 方法栈快 照 。

应用名称	接口名称	调用角色	实例	调用时间			
v java-schedule-service	/callOrderService+	Server					45.218ms
▼ java-order-service	GET /generateOrderInfo+	Server					42.099ms
java-delivery-service	com.tencent.cloudmonitor.dubbo.api.IDeliveryS	Server		0.061ms			
	com.tencent.c TraceID 5849a719075ad6acab42e422b713e87 开始时间 2024-09-26 17:32:02 结束时间 :	8 SpaniD d9537dcd9 2024-09-26 17:32:02 ≶	11562ff8 调用角色 s 实例 10.2.1.104 对蜡	erver 制服务	查看耗时分布	查看容器监控 🛯	就转至接口监控页查看 12
	调用 更多Span信息 CLS日			收起 全屏			
	Key	Value					
	agent.version	opentelemetry-2.3	-20240831				A
	apm.agent.source	opentelemetry-ope	erator				
	component	apache_dubbo					
	cvm.instance.id						
	cvm.region	ap-beijing					
	endTime	endTime 2024-09-26 17:32:02.446					
	host.name	host.name					
	kRe cluster id						•
java-market-service	com.tencent.cloudmonitor.dubbo.api.IMarketSe	Server		0.044ms			
L terre una service		0					



数据库调用监控 数据库概览

最近更新时间: 2024-11-28 14:06:53

数据库概览页展示当前数据库调用总体情况,包括调用拓扑、数据库响应时间、吞吐量、TOP5 慢调用、TOP5 调用者和相关异常列 表等。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 数据库调用 > 数据库概览**。
- 3. 进入数据库概览页面,即可查看相关模块信息。





数据库分析

最近更新时间: 2024-11-28 14:06:53

本文主要介绍数据库调用检索分析相关能力和操作流程。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中选择**应用性能监控 > 数据库调用 > 数据库分析**。
- 3. 进入数据库分析页面,即可查看相关模块信息。

数据库概览 数据库分析												
数据库类型 🗸 数据库地	址		✔ 数	居库名称		~						
调用语句	数据库 ;	调用应用	调用次 💲	耗↓	错 \$	SQLS	分析	异常分析	相关调用链			
select sleep(?)	mock_proje j ct_db s	java-order- service	275 † 100%	7564.68 † 100%	0 -	selec	ct ep(?)				一键复制	*
select dept, count(dept) as total from mock_project_userinfo where $id < ?$ group by dept	mock_proje j ct_db s	java-stock- service	701 † 100%	3.01 † 100%	0 -							
SELECT * FROM 'mock_project_userinfo' WHERE id < 200	mock_proje c ct_db s	go- member- service	263 † 100%	0.71 † 100%	0	4						¥
select dept, count(dept) as total from mock_project_userinfo where id < ? group by dept	mock_proje ^j r ct_db s	java- market- service	821 † 100%	0.49 † 100%	0							
select * from un_exist_table limit ?	j mock_proje c ct_db s	java- delivery- service(vuln erable)	14 † 100%	0.48 † 100%	14 † 100%							
SELECT ?	mock_proje j ct_db s	java-stock- service	1 † 100%	0.45 † 100%	0 -							0
select dept, count(dept) as total from mock_project_userinfo where id < ? group by dept	mock_proje j ct_db s	java-user- service	789 † 100%	0.42 † 100%	0 -							E



访问管理 概述

最近更新时间: 2024-05-15 18:00:42

如果您在腾讯云中使用到了应用性能监控,该服务由不同的人管理,但都共享您的云账号密钥,将存在以下问题:

• 您的密钥由多人共享,泄密风险高。

• 您无法限制其它人的访问权限,易产生误操作造成安全风险。

此时,您就可以通过子用户实现不同的人员管理不同的服务,来规避以上的问题。默认情况下,子用户无使用应用性能监控权限。因 此,我们需要创建策略来允许子用户使用他们所需要资源的权限。

简介

访问管理(Cloud Access Management,CAM)是腾讯云提供的一套 Web 服务,它主要用于帮助客户安全管理腾讯云账户下 的资源的访问权限。通过 CAM,您可以创建、管理和销毁用户(组),并通过身份管理和策略管理控制哪些人可以使用哪些腾讯云资 源。

当您使用 CAM 时,可以将策略与一个用户或一组用户关联起来,策略能够授权或者拒绝用户使用指定资源完成指定任务。有关 CAM 策略的更多相关基本信息,请参见 策略语法 。有关 CAM 策略的更多相关使用信息,请参见 策略 。

授权方式

子用户默认没有应用性能监控任何权限。需要主账号授予子用户相关权限,子用户才能正常访问应用性能监控 APM。

授予 APM 预设策略

应用性能监控 APM 默认创建了预设策略 QcloudAPMFullAccess(应用性能监控(APM)全读写访问权限)和 QcloudAPMReadOnlyFullAccess(应用性监控(APM)只读访问权限),您可以通过搜索策略名称快速进行预设策略授权,更 多信息请参考 策略授予 。

在大数多使用场景下,只需要对子用户授予 APM 预设策略,就可以让子用户正常访问应用性能监控 APM。

授予自定义策略

如果需要根据具体的资源和操作进行精细化授权,请参考 自定义策略 进行操作。



策略语法

最近更新时间: 2024-11-05 12:14:22

概述

访问策略可用于授予访问应用性能监控相关的权限。访问策略使用基于 JSON 的访问策略语言。您可以通过访问策略语言授权指定委 托人(principal)对指定的应用性能监控资源执行指定的操作。

访问策略语言描述了策略的基本元素和用法,有关策略语言的说明可参见 CAM 策略管理。

策略语法

CAM 策略:

"version":"2.0",
"statement":
[
{
"effect":"effect",
"action":["action"],
"resource":["resource"],
<pre>"condition": {"key":{"value"}}</pre>
}
]

元素用法

- 版本 version 是必填项,目前仅允许值为"2.0"。
- 语句 statement 是用来描述一条或多条权限的详细信息。该元素包括 effect、action、resource, condition 等多个其他元素的权限或权限集合。一条策略有且仅有一个 statement 元素。
 - **影响 effect** 描述声明产生的结果是"允许"还是"显式拒绝"。包括 allow (允许)和 deny (显式拒绝)两种情况。该元素是必 填项。
 - 操作 action 用来描述允许或拒绝的操作。操作可以是 API(以 name 前缀描述)或者功能集(一组特定的 API,以 permid 前缀描述)。该元素是必填项。
 - 资源 resource 描述授权的具体数据。资源是用六段式描述。每款产品的资源定义详情会有所区别。有关如何指定资源的信息,请参阅您编写的资源声明所对应的产品文档。该元素是必填项。
 - 生效条件 condition 描述策略生效的约束条件。条件包括操作符、操作键和操作值组成。条件值可包括时间、IP 地址等信息。有些服务允许您在条件中指定其他值。该元素是非必填项。

指定效力

如果没有显式授予(允许)对资源的访问权限,则隐式拒绝访问。同时,也可以显式拒绝(deny)对资源的访问,这样可确保用户无 法访问该资源,即使有其他策略授予了访问权限的情况下也无法访问。下面是指定允许效力的示例:

"effect": "allow"



指定操作

应用性能监控定义了可在策略中指定一类控制台的操作,指定的操作按照操作性质分为读取部分接口 apm:Describe* 和全部接口 apm:* 。

指定允许操作的示例如下:

```
"action": [
"name/apm:Describe*"
]
```

指定资源

资源(resource)元素描述一个或多个操作对象,如应用性能监控资源等。所有资源均可采用下述的六段式描述方式。

qcs:project_id:service_type:region:account:resource

参数说明如下:

参数	描述	是否必选
qcs	是 qcloud service 的简称,表示是腾讯云的云服务	是
project_id	描述项目信息,仅为了兼容 CAM 早期逻辑,一般不填	否
service_ty pe	产品简称,这里为 apm	是
region	描述地域信息	是
account	描述资源拥有者的主账号信息,即主账号的 ID,表示为 uin/\${OwnerUin} ,如 uin/10000000001	是
resource	描述具体资源详情,前缀为 instance	是

下面是应用性能监控的六段式示例:

"resource":["qcs::apm:ap-guangzhou:uin/1250000000:apm/apm-btzsrI123"]

实际案例

基于资源 ID,分配指定资源的读写权限,主账号 ID 为 1250000000: 示例:为子用户分配业务系统(ID:apm-btzsrl123)修改权限。

```
{
    "version": "2.0",
    "statement": [
        {
            "effect": "allow",
            "action": [
            "apm:ModifyApmInstance"
```



```
],
"resource": [
"qcs::apm:ap-guangzhou:uin/1250000000:apm-instance/apm-btzsrI123"
]
}
]
```

支持资源级授权的 API 列表

API 操作名	API 描述
DescribeGeneralApmApplicationConfig	查询应用配置信息
ModifyGeneralApmApplicationConfig	修改应用配置信息
ModifyApmApplicationConfig	修改应用配置
DescribeEventData	获取实例 GC Log 日志或者线程死锁相关信息
DescribeFlameGraphData	获取火焰图数据
CreateEventTask	创建事件任务
DescribeAgentList	查询探针 Agent 列表
DescribeApmInstanceOverview	查询业务系统概要信息
DescribeSpanThreadAnatomyList	查询线程剖析信息
DescribeTraceAnalyzeSpanList	查询链路耗时分析信息
DescribeGeneralMetricData	获取指标数据通用
DescribeApmApplicationConfig	查询应用配置
DescribeTopologyNew	根据服务名查询服务拓扑图
DescribeService	获取服务信息
DescribeMetricLineData	曲线指标数据拉取
DescribeServiceOverview	获取系统概览
DescribeTopologyMetricLineData	拓扑图下曲线拉取
DescribeMetricRecords	通用列表



策略授予

最近更新时间: 2024-05-15 18:00:42

子用户默认没有应用性能监控任何权限。需要主账号授予子用户相关权限,子用户才能正常访问应用性能监控 APM。

操作前提

使用主账号或拥有 QcloudCamFullAccess 权限的子用户登录腾讯云控制台,并参见 新建子用户 操作步骤创建子用户。

APM 预设策略

应用性能监控 APM 默认创建了预设策略 QcloudAPMFullAccess(应用性能监控(APM)全读写访问权限)和 QcloudAPMReadOnlyFullAccess(应用性监控(APM)只读访问权限),您可以通过搜索策略名称快速进行默认策略授权。

自定义策略

- 1. 使用主账号或拥有 QcloudCamFullAccess 权限的子用户进入访问控制 > 策略。
- 2. 单击新建自定义策略,根据不同的需求,选择对应的创建策略方式。
 应用性能监控支持资源级授权和按标签授权两种自定义策略方式:

资源级授权

您可以通过策略语法或默认策略给子账号单个资源的管理权限,详细请参见 策略语法 。

按标签授权

您可以通过给资源标记标签,实现给子账号对应的标签下资源的管理权限,详细请参见 <mark>资源标签</mark> 。

策略关联

- 1. 使用主账号或拥有 QcloudCamFullAccess 权限的子用户进入访问控制 > 策略。
- 2. 进入策略管理页,在策略名称搜索框中输入对应的策略名称。
- 3. 选择只读访问或全读写访问权限,在操作列中单击关联用户/组/角色。

自定义策略 删除			全部策略 预设策略	自定义策略 应用	性能监控	8
策略名	服务类型 ▼	描述		上次修改时间		操作
QcloudAPMFullAccess	应用性能监控	应用性能监控(APM)全读写访问权限		2024-04-25 18:	01:03	关联用户/组/角色
QcloudAPMReadOnlyFullAccess	应用性能监控	应用性能监控(APM)只读访问权限		2024-04-25 18:	00:33	关联用户/组/角色
1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1				1(0▼条/页 🛛 ◀	1 /1页

4. 在弹框中勾选对应的用户,单击确定即可。



资源标签

最近更新时间: 2024-05-16 11:45:51

应用性能监控结合腾讯云资源标签功能,为您提供按标签授予子账号权限和按标签分账功能。 资源标签是腾讯云提供的管理资源工具。资源标签分为标签键和标签值,一个标签键可对应多个标签值,您可以参见以下步骤进行按标 签授权和账单分账。

使用场景

某公司有多个业务系统接入了应用性能监控。这些系统分别由 A、B 两个部门独立研发、运营。现需要对 A、B 部门创建标签、绑定资 源并授予权限,说明如下:

• 创建 A 标签: 绑定 A 部门所有业务系统。

• 创建 B 标签: 绑定 B 部门所有业务系统。

按标签授权

用户 A 为 A 部门开发人员,负责 A 部门所有业务系统开发。需要授予该开发人员 A 标签权限。

按标签分账

用户 B 为公司的财务人员,负责对 A、B 部门财务支出进行独立核算。需要授予该财务人员 A、B 标签权限,并按标签进行分账核 算。

准备工作

步骤1: 创建标签

参见下列步骤,分布创建 A、B 标签。

1. 进入标签列表页。

2. 单击新建标签,进入添加标签页面,填写标签键和对应的标签值。

新建标签		×						
 	 • 输入新的标签键和标签值创建全新标签,选择已有标签键可为该键新增标签值 • 一个标签键最多具有 1000 个标签值,单次创建最多可以输入 10 个标签值 							
标签键	标签值 : 请输入标签值	删除						
添加标签键	确定取消							

3. 单击确定,完成标签创建。

步骤2:为资源分配标签

参见下列步骤,为 A 标签绑定 A 部门下的所有业务系统,为 B 标签绑定 B 部门下的所有业务系统。 1. 进入 应用性能监控控制台 > 资源管理 页面。



2. 单击新建,在弹框中填写信息并绑定标签。也可以在列表中找到已创建的业务系统,在操作列单击修改配置,选择对应的标签。

业务系统名称*	请输入业务	F系统名称	ζ			
	支持长度小	于40的中文	て、英文、娄	攻字以及分隔谷	☆("." ".")	"-")
十费模式	按量付费	۲	而付费	收费规则 🖸		
	- 修改计费横 - 预付费为预 开启按量付到	試后第二; 远先购买以。 费	天生效 Agent*hour	为计费项套餐	包的模式,	选择预付费后将默认
上报地域*	⑤ 广州					
	- 各地域数据 - 如果需要统 个地域。	쨺离,业 还一监控业	务系统创建 务系统下不	后不可更改。 同地域的应用], 您可以选	译将数据上报到同一
字储时长 *	1天	3天	77	157	T	
			1	15人	30大	自定义输入
	链路数据存f 持按需调整	诸时长将决	定您的存储	1 5 人 諸成本,考虑非	30天 基本需求和性	自定又输入 合价比,默认7天,支
业务系统简介	链路数据存付 持按需调整 简单介绍统	诸时长将决	定您的存储	着成本,考虑基	30天	日定又输入 价比,默认7天,支
业务系统简介	链路数据存行 持按需调整 简单介绍统	诸时长将决	完定您的存储	诸成本,考虑3	30天 甚本需求和性	日定又输入 的比,默认7天,支
业务系统简介	链路数据存作 持按需调整 简单介绍统 请在100个字	诸时长将决 您的业务系 "符以内进	定您的存储 统 行描述	13 八 新成本,考虑3	30 天 基本需求和性	日定又输入 5价比,默认7天,支
业务系统简介	链路数据存作 持按需调整 简单介绍统 请在100个字 请选择标袋	諸时长将洪 第101年3月 第111年3月 第11	2定您的存储 统 行描述 请选择	13人 新成本,考虑到 标签值▼ -	30天 基本需求和性	日定又输入 5价比,默认7天,支
业务系统简介 添加标签	链路数据存 持按需调整 简单介绍统 请在100个字 请选择标领 我已阅读并 等级协议》、	部时长将决 密的业务系 ○符以内进 金 ▼ = 司意相关勝概 《计费概	 定您的存储 (第) (第) (第) (第) (第) (第) (第) (第) (第) (第) (第) 	★ 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	30天 基本需求和性 ト 议》、 《腾 译	目定又输入 部价比,默认7天,支

按标签授权

按标签授权的策略,根据下列步骤给用户 A 授予 A 标签权限,用户 B 授予 A、B 标签权限。 1. 进入 策略管理页,单击左上角的**新建自定义策略**。 2. 在弹出的选择创建方式窗口中,单击**按标签授权**,进入按标签授权页面。

腾讯云



3. 在编辑策略页面填写以下信息后单击下一步。

添加服务与操作 添加									
▽ 应用性能监控 (全部導	H/E)								
服务 (Service)	・ 应用性能监控 (apm)								
操作(Action) 收起	- 请选择操作 ✔ 全部操作 (apm:*) 折叠								
	选择操作			已选择 (99)					
	筛选操作			Q		操作名	操作描述		是否支持按标签授权
	操作名	操作描述	是否支持按标签授权 ▼			CreateApmInstance	创建APM实例		否
	CreateApmInstance	创建APM实例	6	Î		CreateApmInstrumentationInfo	创建注入信息		否
	CreateApmInstrumentationInfo	创建注入信息	否			Constant and Constant and Constant	An 140 177 AN 27 189 140 171		
	CreateApmSampleConfig	创建采样配置接口	是		↔	CreateApmSampleConig	创建木件配量按口		定
						CreateEventTask	创建事件任务		是
	CreateEventTask	创建事件任务	是			CreateLogAgentConfig	新增日志agent配置		是
	CreateLogAgentConfig	新增日志agent配置	是			CreateLogClusterConfig	创建集群采集配置		
	CreateLogClusterConfig	创建集群采集配置	是						
	Createl ooOfflineExportTask	创建南线导出日志任务				CreateLogOfflineExportTask	创建南线导出日志任务		是
	支持按住 shift 键进行多选								

4. 进入关联用户/用户组/角色页面,选择需要关联的用户/用户组/角色,检查并完成策略(可修改策略名称)并单击**完成**。

\otimes	腾讯云
-----------	-----

按标签授权		
✓ 编辑策略 >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	> 2 关联用户/用户组/角色	
_		
甘土仁白		
基个信息		
策略名称 *	policygen-20240125161034	
	策略创建后,策略名称不支持修改	
描述	请输入策略描述	
关联用户/用户组/角色	4	
将此权限授权给用户	选择用户	
将此权限授权给用户组	选择用户组	
将此权限授权给角色	选择角色	
	_	

按标签分账

步骤1: 设置分账标签

根据下列步骤设置 A、B 标签为分账标签:

- 若要在账单中使用标签功能,您需要进入费用中心控制台,选择在左侧菜单分账管理>分账标签。被设置为分账标签的标签键会作为账单的单独一列展示,您可根据此标签键来对账单进行筛选和分类展示。
- 在此页面您可看到已创建的标签键列表,选择需要展示的标签键,单击设置为分账标签,即可将该标签键设置为账单中的分账标 签。

分账	标签				
C) 優蓋爆怒为時讯云流源分配的标记,可以用未管理您的资源或用于推单费用的结分。设置为分集标签的	R签键将在账单中作为列来展示(设置生效后,因数据缓存机制在账单中展示的时间可能有不超过24小时	的延迟),每个硼都会成为一个增加列,并在账单中列出每个对应行的值;没有设置为分类标签的其他标签	鍵對在账單中不做關示,另外分账标签将用作账單分组和精选的鉴度。;	分账标签使用指引 [2]
Ŀ	以需为均数转送				输入标签制要素 Q
	有无效法规	金郎标签辑 T	设置时间↓	操作	
	wegastest	分類軟體	2024-04-19 16:12:37	取將設置	
	0ukmyti40	分類範疇	2024-04-03 16:07:25	取將設置	
	Olfextzpp	分類範疇	2024-04-03 16:07:18	取將設置	
	0007knoh	分類範疇	2024-04-03 16:07:08	取將設置	
	user	分類軟語	2023-12-08 14:38:01	取消设置	
	BG	分類軟語	2021-11-26 19:12:26	取消设置	
	mingc_key	分類軟體	2021-11-26 19:11:09	取消设置	
	3xid8wbv			设置为分频标签	
	6hopxcti			设置为分账标签	
	8hi97skd			设置为分账标签	

步骤2:按标签展示账单

您可在 账单概览 页,查看并单击新的选项**按标签汇总**,通过选择具体的标签键,可查看根据该标签键汇总的相关资源的柱状图和列 表。



2024年5月账单汇	2 (单位:元)					▶ 查看成本分析
	按产品汇总	按项目(組)汇总	按地域汇总	按计费模式汇总	按标签汇总 NEW	
		総開助社 标选管理 页创建标签,在各	云资源控制台给资源分面对应的标签值;并前往分解标签页将希望应用在此账单的标签	建设置为分频标签。查 看更多说明 [2		
	标弦键: mingc_key v					
9						47,219.24元
gz	389.86元					
mingc_value1	320.94元					
bj	74.32元					
no_instance	28.56元					

告警服务



最近更新时间: 2024-10-12 15:21:32

本文将为您介绍如何为应用性能关键指标设置告警,在指标发生异常时及时通知您。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中单击告警管理 > 告警配置 > 告警策略。
- 3. 单击新建策略,配置告警策略,配置说明如下:

配置类型	配置项	说明
甘木信白	策略名称	自定义策略名称。
举个旧尽	备注	自定义策略备注。
	监控类型	选择 应用性能监控 类型。
	策略类型	默认选择性能指标。
配置告警规 则	筛选条件 (与)	筛选出符合条件的对象进行告警检测,各筛选条件之间为 AND 关系。筛选条件仅展示有上 报数据的对象。 您可以根据需求选择需要检测的告警对象。
	告警对象维 度	支持自定义告警通知内容中的告警对象,假设您选择了业务系统、应用和调用角色,则告警 对象显示 业务系统、应用、调用角色。
	触发条件	支持满足任意条件、满足所有条件或满足复合条件。
配置告警通 知	通知模板	系统为您默认配置通知模板,如需创建通知模板请参见 新建通知模板 。
高级配置	弹性伸缩	启用并配置成功后,达到告警条件可触发弹性伸缩策略并进行缩容或扩容。



\odot	腾讯云
---------	-----

基本信息									
策略名称	最多60个字符								
备注	最多100个字符								
配置告警规则									
监控类型	一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	HOT 前端他	HOT Hitlisto 元拨						
	一站式链路追踪解决方案,融合	合开源生态,高效排	查故障。了解更多日						
策略类型	性能指标 JVM指标	数据库指标	异常指标	业务日志					
所属标签	标签键	▼ 标签值	Ψ	×					
筛选条件(与) 🛈	地域	• = •	广州	_					
	业务系统	v = v	apm-HiLBZ5qpZ	【(日志压测-按量 ▼					
	业务系统 调用角色	▼ = ▼ ▼ = ▼	apm-HiLBZ5qpZ 服务端	【(日志压测-按量 ▼ ▼	+				
告警对象维度 ③ 鲉发条件	业务系统 调用角色 业务系统 调用角色 将对您筛选条件为地域=广州, 业务系统指标做告警检测	▼ = ▼ ▼ = ▼ 业务系统=apm-HiL	apm-HiLBZ5qpZ 服务端 BZ5qpZ(日志压测-	【(日志压测-按量 ▼ ▼ 按量转预付费了-1	】 + 2日生效),调用角色=服	务端 的			
告警对象维度 ① 触发条件	业务系统 调用角色 业务系统 调用角色 将对您筛选条件为地域=广州, 业务系统指标做告警检测	▼ = ▼ ▼ = ▼ 业务系统=apm-HiL	apm-HiLBZ5qpZ 服务端 BZ5qpZ(日志压测-	【日志压测-按量 ▼ 【日志压测-按量 ▼ ★ <	】 + 2日生效),调用角色=服	务端 的			
告警对象维度 ③ 触发条件	业务系统 调用角色 业务系统 调用角色 将对您筛选条件为 地域=广州, 业务系统指标做告警检测 <u>指标告警</u> 满足以下 任意	 ▼ = ▼ ■ ■	apm-HiLBZ5qpZ 服务端 BZ5qpZ(日志压测- 时,触发告警 ⊻	(日志压测-按量 《 《 按量转预付费了-1 启用告警分级功能	 2日生效), 调用角色=服 ੲ	务端 的			
告警对象维度 ③ 触发条件	业务系统 调用角色 业务系统 调用角色 将对您筛选条件为 地域=广州, 业务系统指标做告 警检测 指标告警 満足以下 任意 if 平均响应时间	 ■ ■	apm-HiLBZ5qpZ 服务端 BZ5qpZ(日志压测- 时, 触发告警 ♥ 粒度1分钟 ♥	【日志压测-按量 ▼ ★ ★ ★ ★ ★	↓ + 2日生效), 调用角色=服 ぎ * 严重: 0毫秒 ◆	务端 的 • 持续 1 个教	u掘点 ▼ ther	每1小时告誓	<u>客</u> ─次 ▼ 〔

🗸 配置告警	> 2 記遣告赞通知		
配置告警通知 添加告警「接收人	/「接收组」,需要在下方选择或新命通知模糊;添加「接口回调」可以点击模糊名称进行操作 ,了解更多 2		
通知模板	选择模板 新建模板 已选择 1 个遇如模板,还可以选择 2 个		
	和原始的资格。 2. 数据中最优化的分子。	电音器作 告警递知当前主歌户	操作
▶ 高级配置(可选,	目前(反支持指标告警条件触发弹性神蝠)		

4. 配置完以上信息后单击**完成**,即成功创建告警策略。在指标发生异常时,将会通过您配置的告警渠道发送告警通知。



查看告警

最近更新时间: 2024-10-12 15:21:32

本文将为您介绍如何查看应用性能监控告警历史。

查看告警历史

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏中单击告警管理 > 告警治理> 告警历史。
- 3. 通过上方进行条件筛选,监控类型选择"应用性能监控"。选择完后单击确定即可进行查询。
- 4. 您还可以单击上方的时间段进行筛选,筛选需要查看告警历史的时间范围。

告營管理								
告賢大益 告 誓历史 策略管理 基础配置								
斯月始后起期已用 1000 年, 約今 0 年9月, 始后起期用用之后包将不会押收到各整股份, 建议总 立即除开始后起现, 以免制能业务, 如果仍然未受到各整, 请自时检测,								
武治発展 新聞性能設在 ・ 2024-04/20 00000 - 2024-04/8 235959 首 古聖状念 ある月を登せる ・ 告疑考察 ある月を登せる 文字をつかって 使用ななわた からなら	▼ 策略类型 请选择策略类型	▼ 〒 第迭					支持按照策略名称、告	Q Ø ±
	告警时间 ↓	告警状态	告警级别	告警类型	监控类型	策略类型。告誓策略		操作
平均规范时间=>0毫秒 调码指电=服务组_业务系统+upm-HLB2Xg2(日志王调-Ip起展转扬行费了-12日生效),应用+garus-market-service	04-02 16:33:00	▲ 告璧中 持续告警 5天17小8打26分钟		指标	应用性能观测	性能指标 apm		屏蔽
最大電話が同→rの違約 適応時色・服务施 止身系統+uppm HL&ZXpg2(日き任調)-把握終新行費了12日生效,应用-ijsro-market-service	04-02 16:33:00	▲ 告警中 持续告警 5天17小时26分钟		指标	应用性能观测	性能描示 apm		屏蔽
留意思=-0% 调問角色=服务瓶 业务系統=spm-HL&ZXg-Z[日志任調-技趣終新行費了12日生效」应用-ijsro-market-service	04-02 16:33:00	▲ 告警中 持续告警 5天17小时26分钟		指标	应用性能观测	性能指标 apm		屏蔽
存在量==0次秒 调用商品=服装纸_业务系统=upm-HLBZCg2(日在正测-按量转扬行费了-12日生效),应用=java-market-service	04-02 16:33:00	▲ 告警中 持续告警 5天17小时26分钟		指标	应用性能观测	性能指标 apm		屏蔽
<mark>建設数→0个</mark> 调用商色=服务组 业务系统-spon-HLBZSg2(日志王副-按惠联系约约数7-12日生效)。应用-jaroa-market-service	04-02 16:33:00	▲ 告鑒中 持续告誓 5天17小时28分钟		指标	应用性能观测	性能指标 apm		屏蔽
共 5 条							20 * 条/页 1 4 4 1	/1页 → H

系统配置 业务系统配置

最近更新时间: 2025-02-14 15:41:22

本文将为您介绍业务系统的相关配置信息。

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择选择**应用性能监控 > 系统配置**,进入**业务系统配置**页面。
- 3. 在页面顶部选择地域和目标业务系统。
- 4. 针对如下的分组栏目,修改业务系统配置。
 - 基本信息:该分组支持业务系统基本信息配置,包括以下配置项:

配置项	说明
业务系统 名称	自定义业务系统名称。
开启免费 模式	开启免费模式后,该业务系统将永久免费。关于免费模式的限制,请参见 关于免费模式的使用限制。
计费模式	支持 按量付费 和 预付费 。
链路存储 时长	支持选择1天、3天、7天、15天、30天链路数据存储时长,试用期间默认存储时长为1天。存储时长越 长,收费越高。超过存储时长的链路将不会展示在应用性能监控控制台。
业务系统 简介	可以简单描述业务系统用途等。
添加标签	应用性能监控结合腾讯云资源标签功能,为您提供按标签授予子账号权限和按标签分账功能。请参见 访 问管理 设置标签。

○ 自定义配置:

支持**自定义展示列**配置,每添加一个自定义展示列以后,在链路追踪等列表视图中,将使用额外的一列展示对应的 Span 属 性。

○ 应用健康阈值:

支持**响应时间警示线**和错误率警示线</mark>配置,在应用列表等视图中,APM 将基于应用健康阈值来判断应用健康状态。

○ 应用安全:

支持**组件漏洞扫描**和 SQL 注入分析应用安全功能项开关。在业务系统中打开或关闭应用安全功能项,将对该业务系统的所有应 用生效。此外,APM 也提供了应用级别的应用安全功能项开关,用于覆盖业务系统中的配置。

〇 日志关联:

支持**日志地域、日志集、日志主题**配置项,用于实现链路与日志服务 CLS 的关联。此外,APM 也提供了应用级别的日志关联 配置项,用于覆盖业务系统中的配置。关于链路日志关联分析的更多介绍,请参见 <mark>在链路详情中查询关联日志</mark> 。

○ Dashboard 关联:

用于在应用详情中关联展示云监控 Dashboard。此外,APM 也提供了应用级别的 Dashboard 关联配置项,用于覆盖业务 系统中的配置。关于 Dashboard 关联的更多介绍,请参见 Prometheus 数据集成以及 Dashboard 关联展示。





应用配置

最近更新时间: 2025-02-21 09:16:01

操作步骤

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择**应用性能监控 > 系统配置**,进入**应用配置**页面。
- 3. 在下拉框选择目标业务系统和目标应用。
- 4. 针对如下的分组栏目,修改应用配置。
 - 指标配置:用于自定义指标数据的处理规则以及展现形式,包括如下配置项:

配置项	说明
接口过滤	通过接口名过滤不需要展示的接口,例如:operationName(.*?),operation(*)。多个规则之间 用英文逗号分割。
错误类型过滤	被过滤的错误类型不会在错误分析中被统计。请按照正则表达式输入需要过滤的错误类型,多个错误 类型之间请使用英文逗号分隔。
HTTP 状态码 过滤	被过滤的 HTTP 状态码不会在错误分析中被统计。默认情况下,HTTP 状态码在大于400被认为 是错误,多个 HTTP 状态码之间请使用英文逗号分隔。
URL 收敛	开启后,系统将根据自定义的收敛规则正则,以及收敛阈值,将同一类接口进行聚合展示。
收敛阈值	收敛阈值是指要进行 URL 收敛的最低数量条件,例如:当阈值为100时,则符合规则正则表达式的 接口数量达到100时进行收敛。
收敛规则正则	以正则表达式自定义 URL 收敛规则。例如:当前半部分都为 /service/demo/ 的接口数超过100 个时,将该类接口进行聚合展示,则收敛阈值需填写为100,收敛规则正则 为/service/demo/(.*?)。
排除规则正则	排除规则正则表示具有该特性的 URL 将不纳入 URL 收敛。例如:前半部分为 service/demo/example 不需要进行 URL 收敛,则表达式 为/service/demo/example/(.*?)。

- **日志关联**: 支持**日志地域、日志集、日志主题**配置项,用于实现链路与日志服务 CLS 的关联,可以覆盖业务系统中的配置。关 于链路日志关联分析的更多介绍,请参见 在链路详情中查询关联日志 。
- **应用安全**: 支持**组件漏洞扫描**和 SQL 注入分析应用安全功能项开关,可以覆盖业务系统中的配置。
- **Dashboard 关联**:用于在应用详情中关联展示云监控 Dashboard,可以覆盖业务系统中的配置。关于 Dashboard 关联 的更多介绍,请参见 Prometheus 数据集成以及 Dashboard 关联展示 。
- 探针配置: 仅适用于腾讯云增加版 OpenTelemetry Java 探针,在修改探针配置后,系统会将配置项推送给探针,应用不需 要重启。

配置项	说明
探针总开关	可以通过探针总开关临时关闭数据上报,关闭后该应用不再产生 APM 费用,应用无需重启。如果需 要永久关闭数据上报,请卸载探针,以免不必要的性能开销。
	① 说明:



	该功能仅支持1.16-20241031及以上版本探针。
	如果某个接口调用耗时频繁超过 慢调用监听阈值 ,该接口将有机会自动生成方法栈执行快照。您可以在 <mark>链路追踪</mark> 页面过滤出包含方法栈快照的调用。在 链路详情 页面,如果一个包含了方法栈快照,会展示 方法栈快照 按钮,单击后可以查看本地方法栈的执行详情。
方法栈快照	 说明: 该功能仅支持1.16-20240831及以上版本探针。
	对于被匹配到的接口,探针不会上报相关链路数据。请按照正则表达式输入需要过滤的接口,多个接口 之间请使用英文逗号分隔。
接口过滤	 ① 说明: 该功能仅支持1.16-20240328及以上版本探针。



采样配置

最近更新时间: 2024-10-12 15:29:41

应用性能监控提供链路采样能力,通过尾部采样(Tail Based Sampling)技术实现。合理配置采样,能够降低链路数据存储量。关 于采样配置的详细信息,请参考 通过采样策略降低 APM 使用成本 。

操作步骤

() 说明:

目前采样策略需要通过开白名单的形式提供,请您通过 提交工单 进行申请。

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择点击**应用性能监控 > 系统配置 > 采样配置**页面。

配置全局采样比例

- 1. 在全局采样配置模块,点击编辑。
- 2. 填写采样百分比,有效的采样百分比在10%到100%之间。

() 说明:

请根据实际情况配置是否保存错误链路,以及慢调用保存阈值。如果链路上存在错误的调用或慢调用,整条链路都会被完整保存。推荐开启保存错误链路,并将慢调用保存阈值设置在500ms – 2000ms 之间。

配置全采样接口

当全局采样比例小于100%的时候,可以自定义需要全采样的接口,如果链路经过了需要全采样的接口,整条链路都会被完整保存。 1. 点击**新增全采样接口**。

2. 在对话框中输入策略名,并指定需要进行全采样的接口。

如果在策略中指定了具体的应用,可以不填写接口匹配规则,此时代表经过此应用的所有链路都会被完整保存。您可以基于**精确匹配**、 前缀匹配以及后缀匹配来指定接口。在实际使用场景中,可以将关注度比较高的重要接口定义为全采样接口,确保不错过任何一条针对 重要接口的调用。

() 说明:

对全局采样比例和全采样接口的更新,都可以即时生效。



Prometheus 集成

最近更新时间: 2024-11-25 10:30:33

对于通过 OpenTelemetry 方案接入 APM 的应用,可以使用 OpenTelemetry API 上报自定义指标,APM 服务端可以将指标同 步到腾讯云 Prometheus 监控服务。该页面用于配置 APM 业务系统和 Prometheus 实例之间的关联,关于腾讯云 Prometheus 监控服务,请参见 Prometheus 监控概述,关于 APM-Prometheus 数据集成,请参见 Prometheus 数据集成 以及 Dashboard 关联展示。

操作步骤

- 1. 登录 APM 控制台,前往系统配置 > Prometheus 集成。
- 2. 在**关联配置**中,开启 Prometheus 关联,并选择当前地域的任何一个 Prometheus 实例。每个 APM 业务系统只能关联同地域 的最多一个 Prometheus 实例。

在当前账号第一次做关联操作的时候,需要授予 APM 访问 Prometheus 资源的权限,根据控制台的提示完成服务授权即可,系统会在访问管理(CAM)自动创建名为 APM_QCSLinkedRoleInPromInstance 的角色,单击**同意授权**即可。

服务授权	
同意赋予反	立用性能监控 权限后,将创建服务预设角色并授予 应用性能监控 相关权限
角色名称	APM_QCSLinkedRoleInPromInstance
角色类型	服务相关角色
角色描述	当前角色为应用性能监控(APM)服务相关角色,该角色将在已关联策略的权限范围内访问您的Prometheus监控服务资源。
授权策略	预设策略 QcloudAccessForAPMLinkedRoleInPromInstance
同意授	权取消

3. 单击新增指标同步规则,指定需要同步到 Prometheus 实例的指标。

对于每一条同步规则,您可以基于**精确匹配、前缀匹配**以及**后缀匹配**这三种匹配方式来匹配指标名,也可以指定该规则的生效范围 (可以是该业务系统的全部应用,或某一个具体的应用)。当 APM 服务端收到应用上报的自定义指标后,满足同步规则的指标将 被写入到被关联的 Prometheus 实例中,不满足同步规则的指标将被丢弃。

关联日志 在链路详情中查询关联日志

最近更新时间: 2024-11-20 14:51:01

如果您使用腾讯云日志服务(CLS)保存日志,可以实现链路数据与日志数据的关联,当应用出现性能问题的时候,可以在应用性能监 控 APM 控制台快速查询一条链路所对应的业务日志,及时定位分析并解决问题。

前提条件

腾讯云

已了解并使用腾讯云日志服务(CLS),详情可参见 日志服务相关指引。

操作步骤

步骤1:在日志中注入 TraceID

不同语言不同协议注入 TracelD 方式不同,详情请查看以下文档指引,在日志中注入 TracelD**(其它语言和协议将会陆续支持)**。

- 通过 skywalking 协议 (Java) 上报
- 通过 OpenTelementry 增强版 Java 探针上报

步骤2: 在系统配置页面关联日志

- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择单击**应用性能监控 > 系统配置 > 业务系统配置**页面。
- 3. 选择对应的业务系统,在日志关联页面单击编辑。
- 4. 配置日志关联信息。
 - 开启关联日志。
 - 选择对应的日志地域、日志集和日志主题。
- 5. 配置完后单击确定即可。

日志关联	
关联日志 🛈	
日志源	CLS日志服务
日志地域	♥ 广州 ▼
日志集	
日志主题	·
	确定取消

除了可以在业务系统级别配置日志关联以外,也可以在应用级别配置日志关联,请到 应用性能监控 > 系统配置 > 应用配置 中进行操 作,具体的配置方式与业务系统级别相同。

步骤3:在链路查询页面查看日志信息



- 1. 登录 腾讯云可观测平台。
- 2. 在左侧菜单栏选择单击**应用性能监控 > 链路追踪**,进入链路追踪页面。
- 3. 选择对应的业务系统,找到对应的 TraceID,单击 TraceID 进入链路详情。
- 4. 在右侧窗口切换日志菜单,即可查看日志相关信息,排查故障。

← 链路详情		java-delivery-service Mysql/JDBI/PreparedStatement/execute (#8#: 4ms)					
設用: I加口: (GET)/generateOrderInfo TraceID:	20ms	component: mysql span.kind: client status code: 1 traceID: p: hest.name: spanID:					
		sql 环境 自定义参数 日志					
		查看全部日志	祥情 🖸				
		-					
点击对应调用可查看明细		msg : 15471 [nio-9301-exec-8] o. a. c. c. C. [. [. [/]. [dispatcherServlet] : Servlet. service error 2022-01-23 16.51.21	20 •				
	oms Ioms	展开					
U java-order-service (GET)/generateOrderInfo	danc.						
java-order-service invisional provide a service and the service invision of th	41115	msg · 15471 [nio-9301-exec-8] c.t.c.m.o.service.impl.OrderServiceImpl : into generateOr	rde				
java-order-service /getDeliveryinto iava-delivery-additional/additiona		info 2022-01-23 16:51:21 (raceID :	* }				
java-delivery-service (GET)/getDeliverymo		殿开					
 java-market-service market.MarketService.getCartInfo 		1159 15473 [lt-executor-894] io.grpc.internal.SerializingExecutor : Exception while	9 e				
 java-market-service market.MarketService.getCartInfo/ser 		error 2022-01-23 16:51:21	+				
java-market-service Mysql/JDBI/PreparedStatement/e		展开					
java-market-service market.MarketService.getCartInfo		men					
java-order-service market.MarketService.getCartInfo/client/R		15473 [lt-executor-894] c.t.c.m.m.services.WarketServiceImpl : into getCartInf	fo 📕				
java-order-service market.MarketService.getCartInfo/client/R		(110 2022-01-23 10.31.21 traceID :	÷				
 iava_order_service market MarketService getProductInfo 		展开					



在日志中注入 TraceID OpenTelementry 增强版 Java 探针

最近更新时间: 2024-05-20 14:17:51

通过腾讯云 OpenTelementry 增强版 Java 探针接入 APM 后,可在应用日志中轻松输出上下文 TraceID 和 SpanID,只需要修 改日志配置文件中的 pattern, 即可实现 TraceID 和 SpanID 的注入。OpenTelementry 增强版 Java 探针的日志注入功能支 持如下日志框架:

- Log4j 1.2+
- Log4j2 2.7+
- Logback 1.0+

在日志配置文件中,TracelD 和 SpanID 分别使用 ^{&X{trace_id}} 和 ^{&X{span_id}} 字段注入,如果上下文没有TracelD 或 SpanID,不会影响整行日志的输出。

Log4j

log4j.properties 文件修改示例:

```
log4j.appender.warn.layout=org.apache.log4j.PatternLayout
log4j.appender.warn.layout.ConversionPattern=traceId: %X{trace_id} spanId: %X{span_id}
- %m%n
```

Log4j2

Log4j2支持多种格式配置文件,包括 XML、Json、YAML、Properties 等。以 XML 格式为例,Pattern 有多种写法,请根据 已有的写法参考其中一种:

方式一

<PatternLayout pattern="traceId: %X{trace_id} spanId: %X{span_id} - %m%n"/>

方式二

<pattern>traceId:</pattern>	%X{trace_id}	spanId:	%X{span_id}	- %m%n

方式三



<Layout type="PatternLayout" pattern="traceId: %X{trace_id} spanId: %X{span_id} -%m%n"/>

Logback

logback.xml 文件修改示例:

```
<encoder>
    <pattern>traceId: %X{trace_id} spanId: %X{span_id} %msg%n</pattern>
</encoder>
```

输出效果

当上下文存在 TraceID 和 SpanID 时:

traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	50784806406e989(
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	50784806406e9896	envi ot '	
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	50784806406e9896		
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	d250c1d1319a3af6		
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	d250c1d1319a3af6		
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	d250c1d1319a3af6		
traceId:	db9d6ee1073fae12d6bd07fae2fb39df spanId:	d250c1d1319a3af6		

当上下文不存在 TraceID 和 SpanID 时:

traceId:	spanId:	
traceId:	spanId:	
traceId:	spanId:	
traceId:	spanId:	

Skywalking 协议 (Java)

最近更新时间: 2024-07-19 15:30:01

本文将介绍 Java skywalking 上报方式如何关联 TraceID 。

操作步骤

配置 Logback 输出 TraceID

1. 引入依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-logback-1.x</artifactId>
<version>8.3.0</version>
</dependency>
```

2. 修改 logback-spring.xml 中的 Appender 的 Pattern 格式。



3. 启动项目,打印结果如下:

%tid 会打印 TraceID,默认 TID: N/A,当有请求调用时,会显示 TraceID。



配置 log4j-1x 输出 TraceID

1. 通过 maven 或 gradle 引入 toolkit 依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-log4j-1.x</artifactId>
<version>{project.release.version}</version>
</dependency>
```

2. 修改 log4j1.properties 配置 layout。





3. 启动项目,打印结果如下:

```
假设 TracelD 存在,当您使用 -javaagent 激活 skywalking tracer 后,log4j 将会输出 TracelD。如果 tracer 未激活,
输出将是 TID: N/A 。
```

2822-81-87 16:51:29.343 [TID:8e6/7c87259464dd2a721bdbc1bbb2cf7.67.16415454893168081] [http-nio-9281-exec-1] INF0 o.s.web.servlet.DispatcherServlet -Initializing Servlet 'dispatcherServlet'
2022-01-07 16:51:29.345 [TID:0e67c87259464dd2a721bdbclbbb2cf7.67.16415454893160001] [http-nio-9201-exec-1] INF0 o.s.web.servlet.DispatcherServlet -Completed initialization in 2 ms
2022-01-07 16:51:29.392 [TID:N/A] [main] INFO n.d.b.g.s.s.GrpcServerLifecycle -gRPC Server started, listening on address: *, port: 9090
2822-01-87 16:51:29.412 [TID:W/A] [main] INFO c.t.c.m.o.OrderServiceApplication -Started OrderServiceApplication in 6.981 seconds (JVM running for 14.558)
2822-01-07 16:51:29.423 [TID:0e67c87259464dd2a721bdbc1bbb2cf7.67.16415454893160001] [http-nio-9201-exec-1] INFO c.t.c.m.o.s.impl.OrderServiceImpl -into generateOrderInfo
2022-01-07 16:51:29.870 [TID:0e6/7c8725946/dd2a721bdbc1bbb2cf7.67.16/15454893160001] [http-nio-9201-exec-1] ERROR o.a.c.c.[.[.[.[dispatcherServlet] -Servlet.service() for servlet [dispatcherServlet] in context with path
java.net.ConnectException: Connection refused (Connection refused)
at java.net.PlainSocketImpl.socketConnect(Native Method)

配置 log4j-2x 输出 TraceID

1. 使用 maven 或 gradle 引入 toolkit 依赖。

```
<dependency>
<groupId>org.apache.skywalking</groupId>
<artifactId>apm-toolkit-log4j-2.x</artifactId>
<version>{project.release.version}</version>
</dependency>
```

- 2. 在 log4j2.xml 的 pattern 中配置 [%traceId] 。
- 支持在 log4j2.xml 的 pattern 中配置 [%traceId]。

```
<Appenders>
    <Console name="Console" target="SYSTEM_OUT">
        <PatternLayout pattern="%d [%traceId] %-5p %c{1}:%L - %m%n"/>
        </Console>
    </Appenders>
```

• 支持 log4j2 AsyncRoot,无需其他配置。请参阅下文的 log4j2.xml 演示。有关详细信息: Log4j2异步记录器。

```
<Configuration>

<Appenders>

<Console name="Console" target="SYSTEM_OUT">

<pr
```



</Loggers> Configuration>

```
    支持 log4j2 AsyncAppender,不需要其他配置。请参阅下文的 log4j2.xml 演示。
    有关详细信息: All Loggers Async
```

() 说明

Log4j-2.9 和更高版本要求在类路径上使用 disruptor-3.3.4.jar 或更高版本。在 Log4j-2.9 之前,需要使用 interrupter-3.0.0.jar 或更高版本。这是最简单的配置,并提供最佳性能。要使所有记录器异步,请将 disruptor jar 添 加到类路径中并且 设置系统属性 log4j2.contextSelector 为

org.apache.logging.log4j.core.async.AsyncLoggerContextSelector •

详细可参见: [Mixed Sync & Async] #MixedSync-Async

🕛 说明

Log4j-2.9 及更高版本需要类路径上使用 disruptor-3.3.4.jar 或更高版本。在 Log4j-2.9 之前,需要 disruptor-3.0.0.jar 或更高版本。不需要将系统属性 "Log4jContextSelector" 设置为任何值。

```
<Configuration status="WARN">

<Appenders>

<!-- Async Loggers will auto-flush in batches, so switch off immediateFlush. -->

<RandomAccessFile name="RandomAccessFile" fileName="asyncWithLocation.log"

immediateFlush="false" append="false">

<PatternLayout>

<PatternLayout>

</Pattern>%d %p %class{1.} [%t] [%traceId] %location %m %ex%n</Pattern>

</PatternLayout>

</RandomAccessFile>

</Appenders>

<Loggers>
```



pattern layout actually uses location, so we need to include it
<asynclogger includelocation="true" level="trace" name="com.foo.Bar"></asynclogger>
<pre><appenderref ref="RandomAccessFile"></appenderref></pre>
<root includelocation="true" level="info"></root>
<pre><appenderref ref="RandomAccessFile"></appenderref></pre>

• 支持 log4j2 AsyncAppender,详细信息请参见: Log4j2 AsyncAppender

```
<Configuration>

<Appenders>

<Console name="Console" target="SYSTEM_OUT">

<PatternLayout pattern="%d [%traceId] %-5p %c{1}:%L - %m%n"/>

</Console>

<Async name="Async">

<AppenderRef ref="Console"/>

</Async>

</Async>

</Async>

</Async>

</Async>

</Async>

</Async>

</Async>

</Async>

</Configuration>
```

```
    3. 启动项目,打印结果如下
    假设 TracelD 存在,当您使用 -javaagent 激活 skywalking tracer 后, log4j 将会输出 TracelD 。如果 tracer 未激
    活,输出将是 TID: N/A 。
```





演示 Demo

最近更新时间: 2025-02-14 11:17:12

应用性能监控 APM 提供了演示 Demo,方便用户直观的感受产品功能,快速开启高效的应用可观测旅程 。

使用限制

- 业务系统被固定为北京地域的 APM-Demo,不能接入新的应用到此业务系统。
- 涉及写操作的功能被限制。
- 演示 Demo 通过腾讯云备用域名 cloud.tencent.com.cn 进行展示,因此,登录 腾讯云控制台 请不要使用此备用域名,否则 会造成会话冲突。如果误用了备用域名登录腾讯云控制台,可通过 cloud.tencent.com (国内站)或 tencentcloud.com (国际站)主域名重新登录,再次进入演示 Demo 即可。

Demo 进入方式

演示 Demo 支持免登录使用,您可以点击 演示 Demo 进入体验。同时,我们在 应用性能监控 APM 的控制台 提供了进入演示 Demo 的入口,您可自行进入体验。在演示 Demo 中,相关应用已经提前完成接入并上报可观测数据,用户可以直接体验。

应用性能监控	◆ 限时持患 对象存储COS支持海量多媒体数据存储,安全稳定低成本存储图片、音视频 查看详情 >									
 ⊙ 应用列表 	⑦ 您已进入Demo模式									
	应用列表 ③ 北京 ~ APM-Demo (apm-CUSfIWrsy) ~ 扫码关注公众号器 扫码加技术交流群器								15分钟	台 ♀ 关闭 ▼
(2) 应用诊断	Ħ	矩阵 三列表 应用过滤			Q					接入应用
■ 数据库调用										
国 MQ 监控		应用名称/ID	应用状态 ‡	吞吐量 ↓	平均响应时间 ‡	平均错误率/错误数 ‡	Apdex 🤅 ‡	组件漏洞数/	标签	操作
の 链路追踪		java-market-service 🗗	●健康	5.82qps 🕹	28ms 1 19%	0% 🕹 -	11-	4/4/0	a	清理应用
🕢 应用安全		svc-S419HT1kp 🖻	- perfek	11.6%	2.0110 • 1.070	0个↓-		4700	\checkmark	编辑标签
《 系统配置		java-delivery-service P svc-G6Cflhgnu	 ● 警示 	4.49qps ↓ 13%	529.04ms † 5%	0%↓- 0个↓-	0.99 ↓ 0.3%	4/4/0	0	清理应用编辑标签
···· 资源管理										
		java-user-service D svc-hIOXHT1Ov D	●健康	4.18qps↓ 12.6%	1.18ms † 1.9%	0%↓- 0个↓-	14-	0	0	清理应用 编辑标签
		java-stock-service ம svc-Ty4JKXchm ம	●健康	3.98qps ↓ 12.1%	2.96ms ↓ 21.8%	0%↓- 0个↓-	14-	4/4/0	0	清理应用 编辑标签
		java-order-service ம svc-ItXflhTnu ம	●警示	3.39qps↓ 12.2%	1063.54ms ↓ 9.6%	4.49% ↓ -3.1% 137↑ ↓ -14.9%	0.81 ↑ 0.4%	4/4/0	0	清理应用 编辑标签
		go-member-service ம svc-PGGqXT1Op ம	●健康	1.39qps ↓ 12.8%	0.78ms ↑ 2.1%	0%↓- 0个↓-	14-	0	0	清理应用 编辑标签
		python-transport-service D svc-IWIZi01Op D	●健康	1.39qps↓ 12.8%	3.8ms ↓ 10.3%	0%↓- 0个↓-	14-	0	4	清理应用 编辑标签
		java-schedule-service ம svc-cUSjqMBOv ம	● 异常	0.52qps↓ 13.3%	41.59ms ↓ 9.2%	16.67% ↓ - 78个 ↓ -13.3%	1 🕇 0.2%	0	0	清理应用 编辑标签