

24 | 分布式系统关键技术：全栈监控

2017-12-21 陈皓

左耳听风

[进入课程 >](#)



讲述：柴巍

时长 09:14 大小 4.24M



首先，我们需要一个全栈系统监控的东西。它就像是我们的眼睛，没有它，我们就不知道系统到底发生了什么，我们将无法管理或是运维整个分布式系统。所以，这个系统是非常非常关键的。

而在分布式或 Cloud Native 的情况下，系统分成多层，服务各种关联，需要监控的东西特别多。没有一个好的监控系统，我们将无法进行自动化运维和资源调度。

这个监控系统需要完成的功能为：

全栈监控；

关联分析；

跨系统调用的串联；
实时报警和自动处置；
系统性能分析。

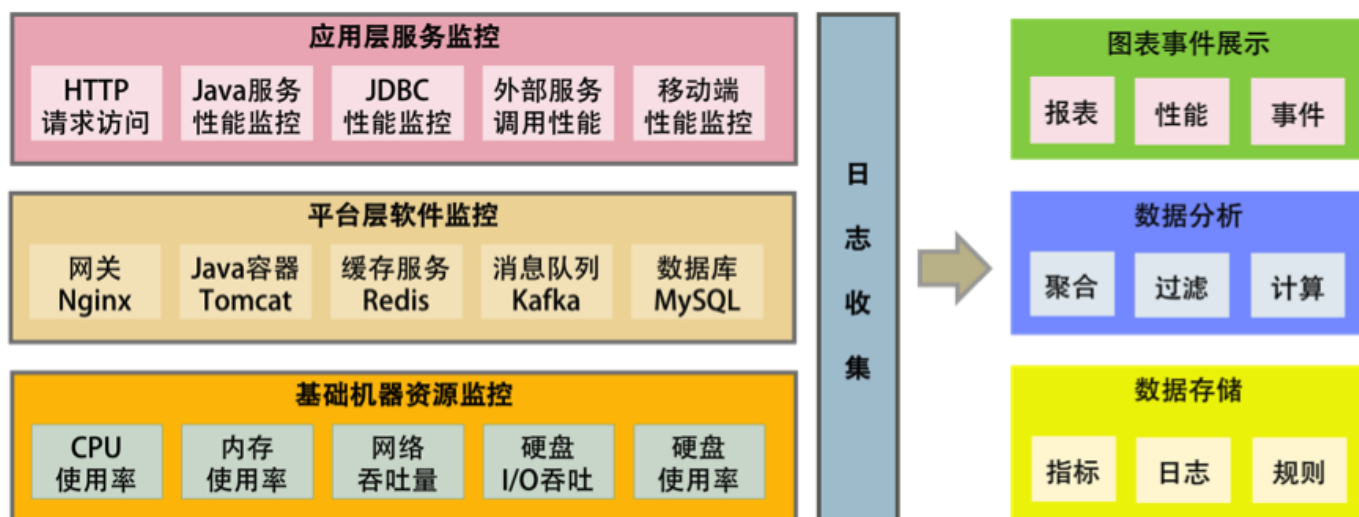
多层体系的监控

所谓全栈监控，其实就是三层监控。

基础层：监控主机和底层资源。比如：CPU、内存、网络吞吐、硬盘 I/O、硬盘使用等。

中间层：就是中间件层的监控。比如：Nginx、Redis、ActiveMQ、Kafka、MySQL、Tomcat 等。

应用层：监控应用层的使用。比如：HTTP 访问的吞吐量、响应时间、返回码，调用链路分析，性能瓶颈，还包括用户端的监控。



这还需要一些监控的标准化。

日志数据结构化；
监控数据格式标准化；
统一的监控平台；
统一的日志分析。

什么才是好的监控系统

这里还要多说一句，现在我们的很多监控系统都做得很不好，它们主要有两个很大的问题。

1. **监控数据是隔离开来的。**因为公司分工的问题，开发、应用运维、系统运维，各管各的，所以很多公司的监控系统之间都有一道墙，完全串不起来。
2. **监控的数据项太多。**有些公司的运维团队把监控的数据项多做为一个亮点到处讲，比如监控指标达到 5 万多个。老实说，这太丢人了。因为信息太多等于没有信息，抓不住重点的监控才会做成这个样子，完全就是使蛮力的做法。

一个好的监控系统应该有以下几个特征。

关注于整体应用的 SLA。主要从为用户服务的 API 来监控整个系统。

关联指标聚合。把有关联的系统及其指标聚合展示。主要是三层系统数据：基础层、平台中间件层和应用层。其中，最重要的是把服务和相关的中间件以及主机关联在一起，服务有可能运行在 Docker 中，也有可能运行在微服务平台上的多个 JVM 中，也有可能运行在 Tomcat 中。总之，无论运行在哪里，我们都需要把服务的具体实例和主机关联在一起，否则，对于一个分布式系统来说，定位问题犹如大海捞针。

快速故障定位。对于现有的系统来说，故障总是会发生的，而且还会频繁发生。故障发生不可怕，可怕的是故障的恢复时间过长。所以，快速地定位故障就相当关键。快速定位问题需要对整个分布式系统做一个用户请求跟踪的 trace 监控，我们需要监控到所有的请求在分布式系统中的调用链，这个事最好是做成没有侵入性的。

换句话说，一个好的监控系统主要是为以下两个场景所设计的。

“体检”

容量管理。提供一个全局的系统运行时数据的展示，可以让工程师团队知道是否需要增加机器或者其它资源。

性能管理。可以通过查看大盘，找到系统瓶颈，并有针对性地优化系统和相应代码。

“急诊”

定位问题。可以快速地暴露并找到问题的发生点，帮助技术人员诊断问题。

性能分析。当出现非预期的流量提升时，可以快速地找到系统的瓶颈，并帮助开发人员深入代码。

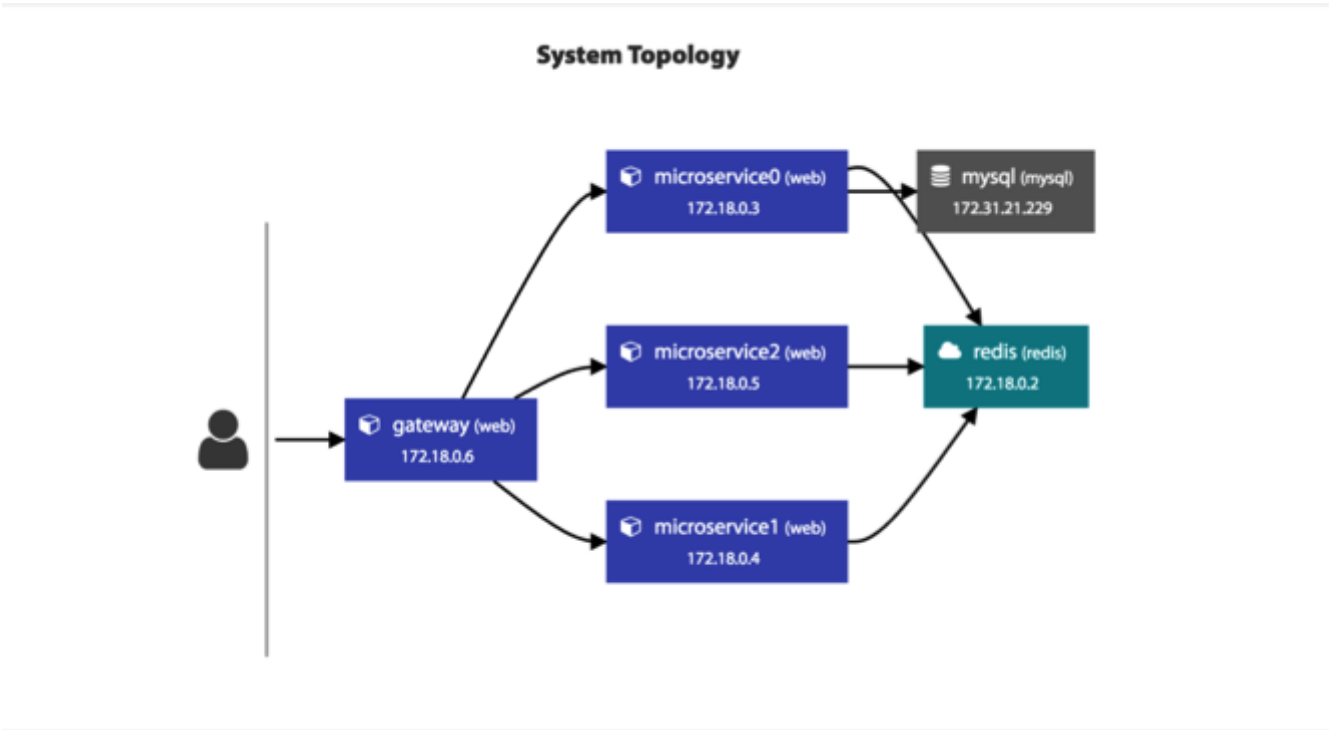
只有做到了上述的这些关键点才能是一个好的监控系统。

如何做出一个好的监控系统

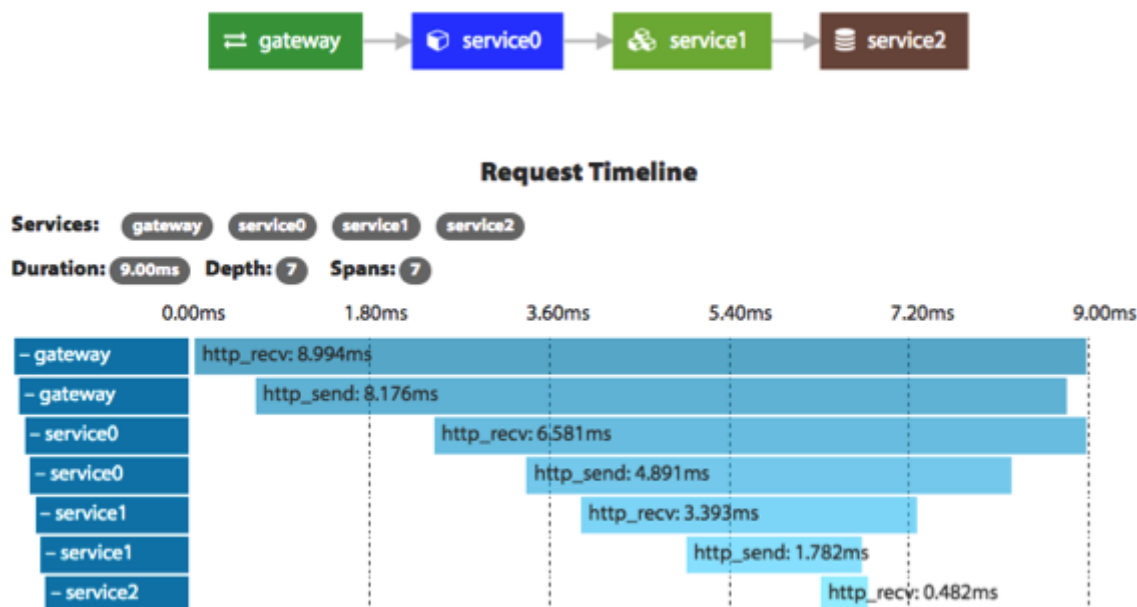
下面是我认为一个好的监控系统应该实现的功能。

服务调用链跟踪。这个监控系统应该从对外的 API 开始，然后将后台的实际服务给关联起来，然后再进一步将这个服务的依赖服务关联起来，直到最后一个服务（如 MySQL 或 Redis），这样就可以把整个系统的服务全部都串连起来了。这个事情的最佳实践是 Google Dapper 系统，其对应于开源的实现是 Zipkin。对于 Java 类的服务，我们可以使用字节码技术进行字节码注入，做到代码无侵入式。

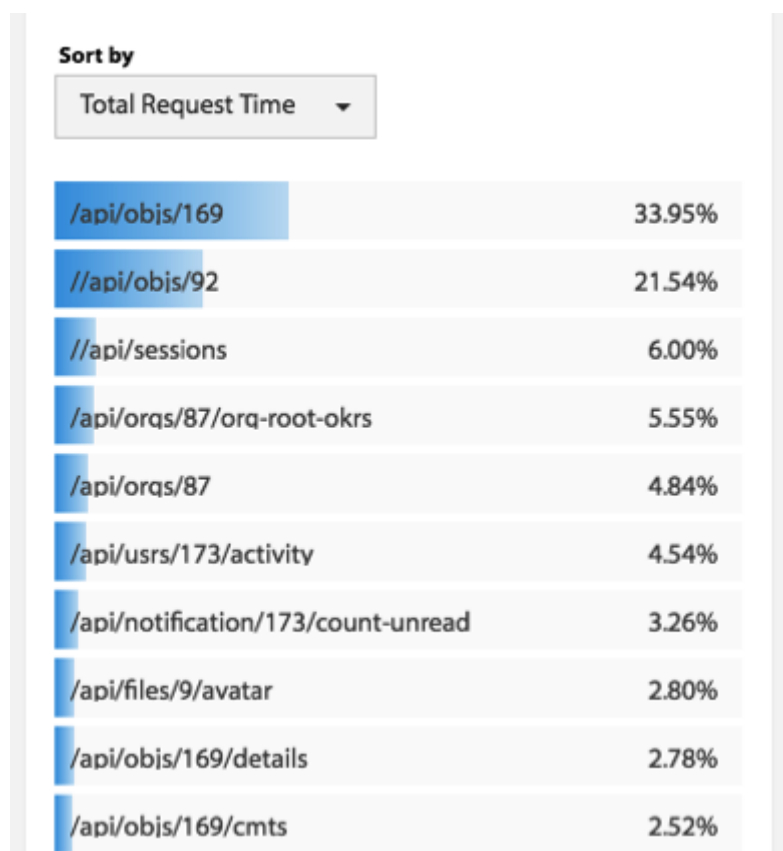
如下图所示（截图来自我做的一个 APM 的监控系统）。



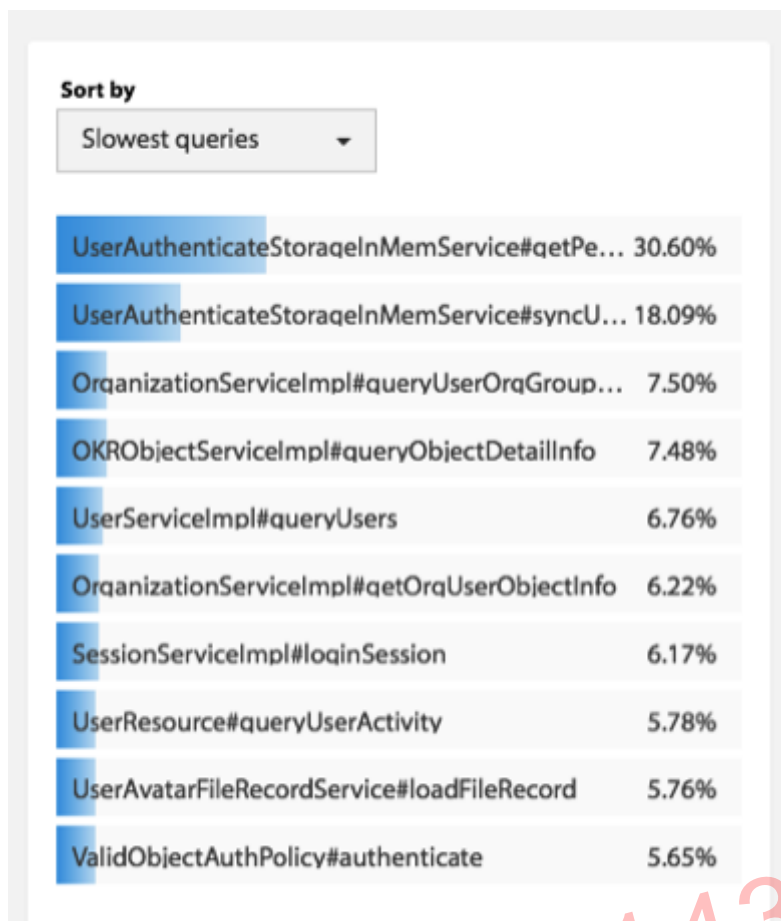
服务调用时长分布。使用 Zipkin，可以看到一个服务调用链上的时间分布，这样有助于我们知道最耗时的服务是什么。下图是 Zipkin 的服务调用时间分布。



服务的 TOP N 视图。所谓 TOP N 视图就是一个系统请求的排名情况。一般来说，这个排名会有三种排名的方法：a) 按调用量排名，b) 按请求最耗时排名，c) 按热点排名（一个时间段内的请求次数的响应时间和）。



数据库操作关联。对于 Java 应用，我们可以很方便地通过 JavaAgent 字节码注入技术拿到 JDBC 执行数据库操作的执行时间。对此，我们可以和相关的请求对应起来。



服务资源跟踪。我们的服务可能运行在物理机上，也可能运行在虚拟机里，还可能运行在一个 Docker 的容器里，Docker 容器又运行在物理机或是虚拟机上。我们需要把服务运行的机器节点上的数据（如 CPU、MEM、I/O、DISK、NETWORK）关联起来。

这样一来，我们就可以知道服务和基础层资源的关系。如果是 Java 应用，我们还要和 JVM 里的东西进行关联，这样我们才能知道服务所运行的 JVM 中的情况（比如 GC 的情况）。

有了这些数据上的关联，我们就可以达到如下的目标。

1. 当一台机器挂掉是因为 CPU 或 I/O 过高的时候，我们马上可以知道其会影响到哪些对外服务的 API。
2. 当一个服务响应过慢的时候，我们马上能关联出来是否在做 Java GC，或是其所在的计算节点上是否有资源不足的情况，或是依赖的服务是否出现了问题。
3. 当发现一个 SQL 操作过慢的时候，我们能马上知道其会影响哪个对外服务的 API。
4. 当发现一个消息队列堵塞的时候，我们能马上知道其会影响哪些对外服务的 API。

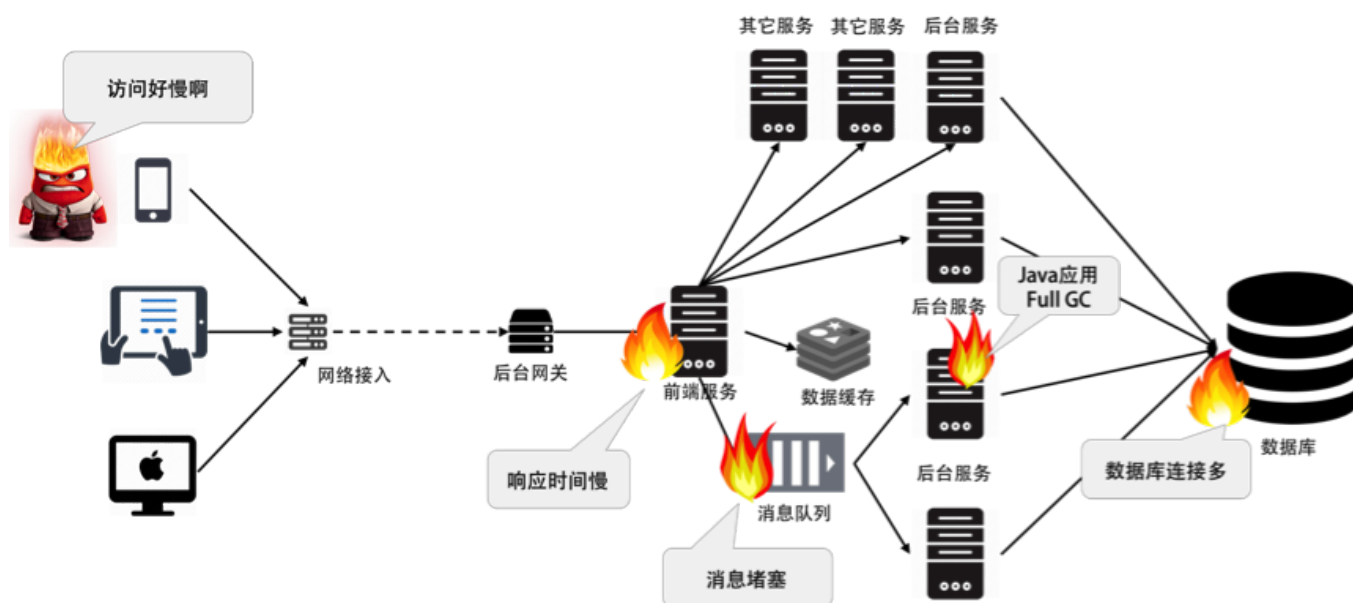
总之，我们就是想知道用户访问哪些请求会出现问题，这对于我们了解故障的影响面非常有帮助。

一旦了解了这些信息，我们就可以做出调度。比如：

一旦发现某个服务过慢是因为 CPU 使用过多，我们就可以做弹性伸缩。

一旦发现某个服务过慢是因为 MySQL 出现了一个慢查询，我们就无法在应用层上做弹性伸缩，只能做流量限制，或是降级操作了。

所以，一个分布式系统，或是一个自动化运维系统，或是一个 Cloud Native 的云化系统，最重要的事就是把监控系统做好。在把数据收集好的同时，更重要的是把数据关联好。这样，我们才可能很快地定位故障，进而才能进行自动化调度。



上图只是简单地展示了一个分布式系统的服务调用链接上都在报错，其根本原因是数据库链接过多，服务不过来。另外一个原因是，Java 在做 Full GC 导致处理过慢。于是，消息队列出现消息堆积堵塞。这个图只是一个示例，其形象地体现了在分布式系统中监控数据关联的重要性。

小结

回顾一下今天的要点内容。首先，我强调了全栈系统监控的重要性，它就像是我们的眼睛，没有它，我们根本就不知道系统到底发生了什么。随后，从基础层、中间层和应用层三个层面，讲述了全栈监控系统要监控哪些内容。然后，阐释了什么才是好的监控系统，以及如何做出好的监控。最后，欢迎你分享一下你在监控系统中的比较好的实践和方法。

下一篇文章中，我将讲述分布式系统的另一关键技术：服务调度。

下面我列出了《分布式系统架构的本质》系列文章的目录，方便你快速找到自己感兴趣的内容。

[分布式系统架构的冰与火](#)

[从亚马逊的实践，谈分布式系统的难点](#)

[分布式系统的技术栈](#)

[分布式系统关键技术：全栈监控](#)

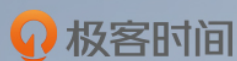
[分布式系统关键技术：服务调度](#)

[分布式系统关键技术：流量与数据调度](#)

[洞悉 PaaS 平台的本质](#)

[推荐阅读：分布式系统架构经典资料](#)

[推荐阅读：分布式数据调度相关论文](#)



左耳朵耗子

全年独家专栏《左耳听风》

20000 名程序员的练级攻略

陈皓 资深技术专家
骨灰级程序员



新版升级：点击「 请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。

© 版权归极客邦科技所有，未经许可不得传播售卖。页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 23 | 分布式系统的技术栈

下一篇 25 | 分布式系统关键技术：服务调度

精选留言 (24)

写留言



怪盗キッド

2018-04-30

12

我使用asm写了一个java接口性能监控和统计的工具MyPerf4J :

<https://github.com/ThinkpadNC5/MyPerf4J>

展开



曹林华

2018-01-06

9

Zipkin 更适合做全链路跟踪，主要有下面两个好处

1. 接入sdk来实现，比较灵活，管理起来方便
2. Pingpoint 通过依赖编织，接入一个 jar 包在jvm 启动命令中，接入简单，但是管理麻烦，不透明



毛洪博

2018-04-07

6

对于业务，中间件，基础监控，三者相互关联，非常认同，想问下，这块如何实现？一个监控系统，如何正确的显示三个层次的信息？

展开



_CountingS...

2018-01-18

5

请问老师的架构图 示意图 是用什么软件画的

展开



shniu

2018-04-06

4

请问浩哥，有哪些好用的开源监控平台吗

展开

作者回复: ELK



whhbbq

2017-12-21

👍 4

陈老师，请问zipkin和pinpoint哪个更好用点

展开 ▾

作者回复: Zipkin



enrique

2018-01-24

👍 2

本文中提到的开源全栈监控系统zipkin适合以go语言为主的开发团队吗？如果不适合，go语言该如何做到同样的效果？

展开 ▾



V

2017-12-20

👍 2

实用

展开 ▾



MarksGui

2018-06-15

👍 1

安卓什么时候才有倍速播放啊？这个功能真的太需要了！



whhbbq

2018-02-02

👍 1

陈老师，这些好的监控系统的特性zipkin都能实现吗？

展开 ▾



永靖

2017-12-20

👍 1

关于监控组件，有没有开源的可以参考

展开 ∨



郎哲

2017-12-19

👍 1

没有做监控的可以参考做一下啦，基础监控一般像阿里云和AWS提供商都做了



it-spurs

2019-04-21

👍

对比过各个开源分布式调用链系统，其中skywalking各方面都更优秀，代码无侵入，性能好，界面强大，监控指标多，有监控告警，提供非常多的插件，可以监控市面上大部分主要的开源框架和中间件，比如rabbitmq，dubbo,springcloud,mysql，是一款国内的优秀开源软件，已经在Apache孵化器，生态也在扩展。

展开 ∨



Lincoln

2019-04-14

👍

皓哥，我们公司也有做数据监控，包括业务层，中间层和底层，但是三者之间没有什么关联，平时定位问题只能通过时间点来分析三者的关系，但是感觉很麻烦。今天看了这篇文章，刚好提到了关联的重要性，但我还是不知道用什么好的技术手段去进行监控的关联？

展开 ∨



YSS

2019-03-21

👍

皓叔，APM是基于zipkin开发的？

展开 ∨



Wander

2019-02-20

👍

陈老师，我在一家硅谷startup工作，感觉我们的工程师都没有重视全栈监控的意识（比如到处200加err code）。想请问您有没有推荐的英文资料能解释清楚全栈监控重要性的？谢谢！



海

👍



2019-01-10

耗子哥，现在分布式监控中间件繁多，做好分布式中间件需要选型哪些比较好的组件可以提及一下么，另外每个组件功能并不全面，如何优雅的将监控信息进行整合到一个或多个仪表盘



godtrue

2018-12-30



对比一下公司的各种基础工具，确实如此，各个工具虽有打通，但是还不够，排查一个问题需要登录许多的工具操作平台。上下游服务也是隔离的，比如：调用其他组的服务超时，只能找他们继续定位具体原因；还有报警系统和日志系统是隔离的，查一个错误需要多个系统配合，业务问题必须扒代码，然后找日志来定位。

...

展开 ∨



Wendy

2018-11-17



老师，用zipkin方便对业务日志的收集吗

展开 ∨



对酒当歌

2018-06-04



陈老师，zinkin在线上实时监控消耗性能吗，我是用在线上好还只是性能压力测试。