





分布式系统关键技术：全栈监控

陈皓

- 00:00 / 09:14

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。

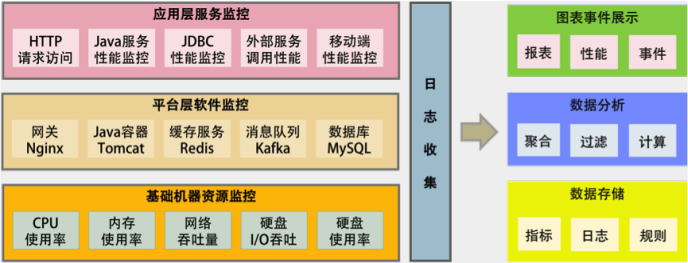
而在分布式或Cloud Native的情况下，系统分成多层，服务各种关联，需要监控的东西特别多。没有一个好的监控系统，我们将无法进行自动化运维和资源调度。

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也是各是各的，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，并可以帮助开发人员深入代码。

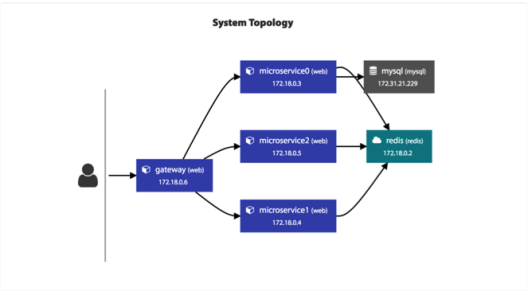
只有做到了上述的这些才能是一个好的监控系统。

如何做出一个好的监控系统

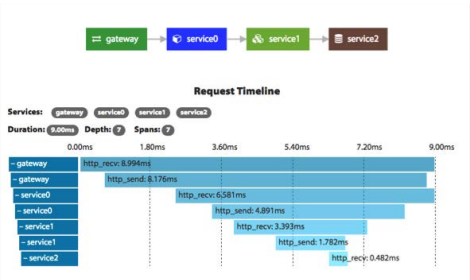
下面是我认为一个好的监控系统应该实现的东西。

- 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，再将这个服务的依赖服务给关联起来，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）。



- 服务调用链长分布。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）。

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。

文末给出了《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](#)



曹林华	2018-01-06
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	
_CountingStars	2018-01-18
请问老师的架构图 示意图 是用什么软件画的	
毛洪博	2018-04-07
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？	
shniu	2018-04-06
请问浩哥，有哪些好用的开源监控平台吗 作者回复	2018-04-08
ELK	
怪盗キッド	2018-04-30
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	
enrique	2018-01-24
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？	
V	2017-12-20
实用	
MarksGui	2018-06-15
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
陈老师，这些好的监控系统的特性zipkin都能实现吗？	
whhbbq	

陈老师，请问zipkin和pinpoint哪个更好用点 作者回复	2017-12-21
Zipkin	2017-12-24
永靖	2017-12-20
关于监控组件，有没有开源的可以参考 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。 曹铮	2018-04-22
皓哥，我们团队基于微服务的架构，做了大量的监控，比如elk针对错误日志的告警，promethues针对系统容器和应用的监控，数据库使用阿里云本身的监控，全部整合到一套仪表盘中但即使如此，我还作为负责人还是感到很“虚”，体现在几处：1.没法预防，这个感觉很难 2.很难把握监控的粒度，粗了，定位问题太模糊，细了，工作量大，又怕影响性能（就当我要流氓吧）。皓哥说的字节码技术应该就是动态AOP吧，基于现在spring系，做aop简单但是很容易出错 毛洪博	2018-04-07
监控和报警的关系怎么界定？用报警来定位故障，还是需要通过监控来定位？如果出现了大故障，整个链路都会报警，报警数量特别多，很容易将最初最核心的报警给淹没了，想知道这个问题是怎么解决的？ 张祥	2018-03-01
学习了一些zipkin的东西，几点疑问？就拿Java服务举例，服务增加这种侵入式监控，开发延迟受影响的程度如何衡量？collector如何承受高并发去扩容？集群模式确定单个实例故障延迟容易吗？ sonnyching	2017-12-28
我们这也没有地图..... 郎哲	2017-12-19
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	

