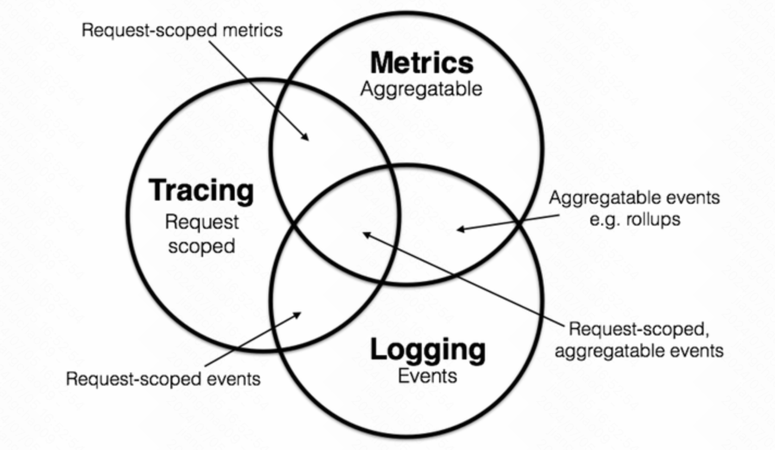
# 背景

一般会将可观测性分解为三个更具体方向进行研究，分别是：事件日志、链路追踪和聚合度量，这三个方向各有侧重，又不是完全独立，它们天然就有重合或者可以结合之处。



它们的特征转述如下：

● 日志（Logging）：日志的职责是记录离散事件，通过这些记录事后分析出程序的行为，譬如曾经调用过什么方法，曾经操作过哪些数据，等等。

● 追踪（Tracing）：单体系统时代追踪的范畴基本只局限于栈追踪（Stack Tracing），调试程序时，在 IDE 打个断点，看到的 Call Stack 视图上的内容便是追踪；编写代码时，处理异常调用了 Exception::printStackTrace()方法，它输出的堆栈信息也是追踪。微服务时代，追踪就不只局限于调用栈了，一个外部请求需要内部若干服务的联动响应，这时候完整的调用轨迹将跨越多个服务，同时包括服务间的网络传输信息与各个服务内部的调用堆栈信息，因此，分布式系统中的追踪在国内常被称为“全链路追踪”（后文就直接称“链路追踪”了），许多资料中也称它为“分布式追踪”（Distributed Tracing）。追踪的主要目的是排查故障，如分析调用链的哪一部分、哪个方法出现错误或阻塞，输入输出是否符合预期，等等。

● 度量（Metrics）：度量是指对系统中某一类信息的统计聚合。Java 天生自带有一种基本的度量，就是由虚拟机直接提供的 JMX（Java Management eXtensions）度量，诸如内存大小、各分代的用量、峰值的线程数、垃圾收集的吞吐量、频率，等等都可以从 JMX 中获得。度量的主要目的是监控（Monitoring）和预警（Alert），如某些度量指标达到风险阈值时触发事件，以便自动处理或者提醒管理员介入。

日志收集和分析大多被统一到 Elastic Stack（ELK）技术栈上，如果说未来还能出现什么变化的话，也就是其中的 Logstash 能看到有被 Fluentd 取代的趋势，让 ELK 变成 EFK，但整套 Elastic Stack 技术栈的地位已是相当稳固。度量方面，跟随着 Kubernetes 统一容器编排的步伐，Prometheus 也击败了度量领域里以 Zabbix 为代表的众多前辈，即将成为云原生时代度量监控的事实标准，虽然从市场角度来说 Prometheus 还没有达到 Kubernetes 那种“拔剑四顾，举世无敌”的程度，但是从社区活跃度上看，Prometheus 已占有绝对的优势，在 Google 和 CNCF 的推动下，未来前途可期。

追踪方面的情况与日志、度量有所不同，追踪是与具体网络协议、程序语言密切相关的，收集日志不必关心这段日志是由 Java 程序输出的还是由 Golang 程序输出的，对程序来说它们就只是一段非结构化文本而已，同理，度量对程序来说也只是一个个聚合的数据指标而已。但链路追踪就不一样，各个服务之间是使用 HTTP 还是 gRPC 来进行通信会直接影响追踪的实现，各个服务是使用 Java、Golang 还是 Node.js 来编写，也会直接影响到进程内调用栈的追踪方式。这决定了追踪工具本身有较强的侵入性，通常是以插件式的探针来实现；也决定了追踪领域很难出现一家独大的情况，通常要有多种产品来针对不同的语言和网络。近年来各种链路追踪产品层出不穷，市面上主流的工具既有像 Datadog 这样的一揽子商业方案，也有 AWS X-Ray 和 Google Stackdriver Trace 这样的云计算厂商产品，还有像 SkyWalking、Zipkin、Jaeger 这样来自开源社区的优秀产品。

## 事件日志

事件日志是记录系统中关键事件和操作的重要手段之一。它能够收集系统的关键日志信息，并提供丰富的上下文数据，用于故障排查、行为分析和安全审计。在实践中，我们可以通过引入分布式日志收集工具，如 Google Cloud Logging、Microsoft Azure Monitor 和Amazon CloudWatch Logs，来收集和管理分布式系统的日志数据。

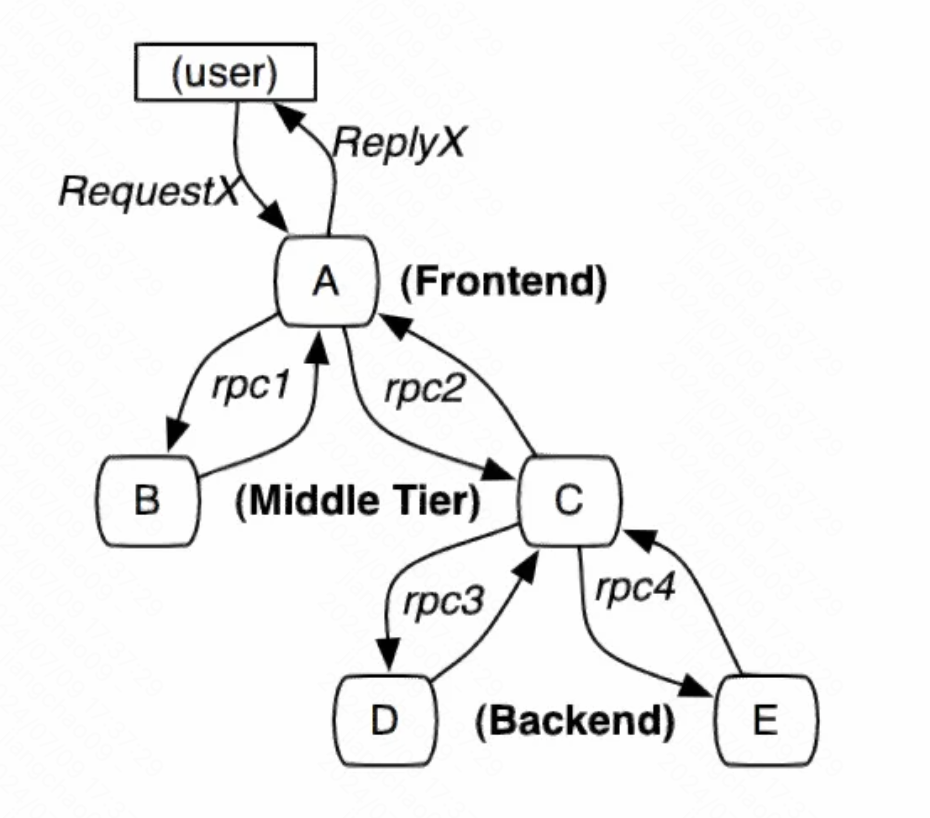
云原生架构的发展推动了日志管理的创新。通过在容器中集成日志代理和收集器，我们可以方便地将应用程序的日志发送到集中式日志存储中。同时，日志存储和分析工具也不断进化，提供了更强大的查询和分析功能。例如，Google 的 Stackdriver Logging提供了实时日志监控和查询的能力，帮助我们更高效地分析和诊断分布式系统的问题。

## 链路追踪

链路追踪是追踪分布式系统中请求的调用路径和性能的技术。通过在请求中添加唯一标识符，并记录请求经过的各个服务和组件，我们可以了解请求的流程、延迟以及调用链中的异常情况。链路追踪技术在诊断系统延迟、性能优化和故障排查等方面起着重要作用。

大公司如 Google、Microsoft 和Amazon都提供了成熟的链路追踪解决方案。例如，Google的Dapper、Microsoft的 Application Insights和 Amazon 的 AWS X-Ray等。这些工具使用了各自的追踪协议，并提供了可视化界面和查询接口，帮助开发者深入了解分布式系统的调用关系和性能瓶颈。

在实际应用中，我们需要在系统的各个关键服务中集成链路追踪库，并确保请求在整个系统中保持唯一标识。通过分析和聚合链路追踪数据，我们可以获得系统整体的性能指标，并发现潜在的性能瓶颈和异常情况。



## 聚合度量

聚合度量是对分布式系统中的关键指标进行收集、计算和汇总的过程。通过聚合度量，我们可以获得对系统整体性能和行为的洞察。在大规模分布式系统中，有许多指标需要跟踪和监控，如请求吞吐量、错误率、延迟等。Google的 Prometheus、Microsoft 的 Azure Monitor和 Amazon 的CloudWatch Metrics是一些常用的聚合度量工具，它们提供了强大的度量指标收集和查询功能，帮助我们监控和优化分布式系统的性能。

在应用聚合度量时，我们需要定义合适的指标，根据系统的需求和关注点选择合适的度量工具，并设置合理的采样频率和存储策略。通过分析聚合度量数据，我们可以发现系统的趋势和异常情况，从而优化系统的性能和可靠性。

# 开源组件

分布式可观测性：<https://mp.weixin.qq.com/s/Hdt3Abl_E3YoXHBniZNQbQ>

开源日志分析工具：<https://blog.csdn.net/zl1zl2zl3/article/details/89817424>

## Syslog

Syslog是一种工业标准的协议，可用来记录设备的日志。有多种变体，包括builds和add-ons。

## ELK stack

ELK stack工具包含Elasticsearch、Logstash和Kibana三个开源软件。

## Nagios

它的主要产品是日志服务器，旨在简化数据收集并使系统管理员更容易访问信息。Nagios 日志服务器引擎将实时捕获数据，并将其提供给一个强大的搜索工具。通过内置的设置向导，可以轻松地与新端点或应用程序集成。

## Fluentd

Fluentd是一个强大的数据收集解决方案，它是完全开源的。它没有提供完整的前端界面，而是作为一个收集层来帮助组织不同的管道。Fluentd 在被世界上一些最大的公司使用，但是也可以在较小的组织中实施。

## DeepFlow

参考：

<https://mp.weixin.qq.com/s/IJntZDqBpLOWP2-JGY6Hmw>

DeepFlow开源项目旨在为复杂的云原生应用提供深度可观测性。DeepFlow基于 eBPF 实现了零插桩（Zero Code）、全覆盖（Full Stack）的指标、追踪、日志采集，并通过智能标签技术实现了所有观测数据的全关联（Universal Tagging）和高效存取。使用 DeepFlow，可以让云原生应用自动具有深度可观测性，从而消除开发者不断插桩的沉重负担，并为 DevOps/SRE 团队提供从代码到基础设施的监控及诊断能力。

DeepFlow 基于eBPF实现的零侵扰分布式追踪可以分解为微服务间追踪、微服务内追踪两部分：

微服务间追踪，技术核心包括：

1、eBPF 编程：通过 eBPF 编程对操作系统 Process 调用数据采集；

2、零插桩追踪信息采集：通过算法计算客户端 Process、服务端 Process 位置的 TCP 流追踪信息；

3、流量采集：采集解析应用调用在客户端网卡、服务端网卡位置的 TCP 流追踪信息；

4、追踪关联：TCP 流追踪信息串联客户端 Process、客户端网卡、服务端网卡、服务端 Process 的追踪关系。

微服务内追踪，技术核心包括：

1、eBPF 编程：通过 eBPF 编程对操作系统对 Process 应用调用数据采集；

2、零插桩追踪信息采集：eBPF 采集同一 Process 前端、后端 2 次应用调用的 Syscall，通过算法计算 2 次调用的关联关系，生成 Syscall Trace ID；

3、追踪关联：利用 2 次调用的 Syscall Trace ID 串接同一 Process 前、后端应用调用的追踪关系。

## Prometheus/Zabbix

## Grafana

## Splunk

# 分布式链路追踪模型

参考：

各大厂分布式链路跟踪系统架构对比：

<https://www.jianshu.com/p/cd2696879877>

分布式链路追踪—常用技术选型：<https://developer.aliyun.com/article/1143845>

## Open Tracing模型

参考：

<https://www.lixueduan.com/posts/tracing/01-opentracing/>

## Google Dapper

参考：

全链路追踪系统技术分析：<https://chenquan.me/posts/tracing-system-analysis/>

## Twitter Zipkin

## Uber Jaeger

参考：

[全链路追踪](https://jckling.github.io/2021/04/02/Jaeger/%E5%85%A8%E9%93%BE%E8%B7%AF%E8%BF%BD%E8%B8%AA%E4%B8%8E%20Jaeger%20%E5%85%A5%E9%97%A8/index.html)

## Elastic Stack：Elastic APM

## Apache SkyWalking

## Naver：Pinpoint

## 阿里：鹰眼

## 大众点评：Cat

## 京东：Hydra

# 业界方案

## MySQL

### Percona-Toolkit

Percona-Toolkit工具包，包括pt-query-digest、pt-online-schema-change、pt-stalk等。

pt-stalk的主要功能是在出现问题时收集OS及MySQL的诊断信息，这其中包括：

1、OS层面的 CPU、IO、内存、磁盘、网络等信息；

2、MySQL层面的行锁等待、会话连接、主从复制，状态参数等信息。

### MEM

MySQL 企业版里的MySQL Enterprise Monitor工具可用于监控MySQL实例和主机性能。

## PostgreSQL

## APM/DBBrain

APM：<https://mp.weixin.qq.com/s/b5y_132TTvouKQepGs7rAw>

DBBrain智能运维：<https://mp.weixin.qq.com/s/rXvVB_Hhginab-TmADQpzg>

DBBrain数据库自治：

<https://mp.weixin.qq.com/s/nkM5ZM-VVVUO_GcJs6-WFg>

DBBrain全链路分析：<https://mp.weixin.qq.com/s/fHh4C8SzIoATu3dAYP5zbw>

DBBrain智能优化引擎：

<https://mp.weixin.qq.com/s/BxBh6bWi5nvFJUAoj-efQw>

## TiDB

参考：

TiDB可观测性：<https://mp.weixin.qq.com/s/LrWEUrve0Bi_AuZ4WpUkXQ>

基于 DeepFlow 的 TiDB 全栈全链路可观测性最佳实践：

<https://mp.weixin.qq.com/s/Tb4RQac4c9xnx33ajfY5AA>

## OceanBase

OceanBase全链路追踪：<https://open.oceanbase.com/blog/1775421184>