

调用链调研分享

- - 1.调用链产生的背景
 - 2. 生产级链路追踪系统Dapper
 - 2.1 采样率
 - 2.2 透明性
 - 3. 链路追踪系统的常见的架构
 - 4. 流行的链路追踪系统
 - 4.1 zipkin
 - 4.2 PinPoint
 - 4.3 SkyWalking
 - 4.4 CAT
 - 4.5 Spring Cloud Sleuth
 - 4.6 系统对比
 - 5. Spring Cloud Sleuth
 - 5.1 Spring Cloud微服务集成Sleuth的步骤
 - 6. 总结

1.调用链产生的背景

互联网服务通常都是集群化部署的分布式系统，2点考虑：

- 成本
- 性能

一个前端请求过来通常在后台涉及到多个的服务调用

所以需要链路追踪系统。链路追踪系统主要就是需求就是持续的监控大规模的分布式服务系统，所以要求有3个：

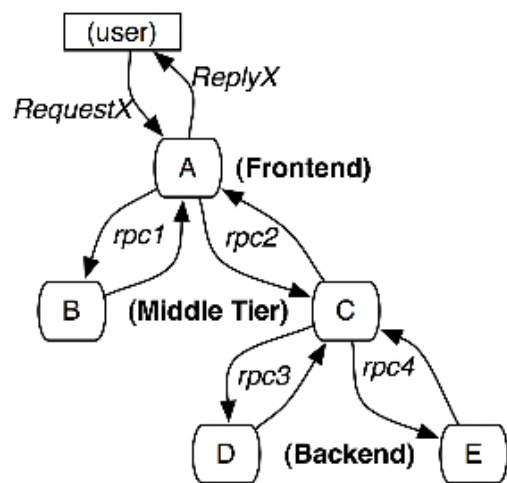
- 低消耗；
- 代码侵入低或者也可以叫做应用级透明；
- 足以应付的规模也足够大；

一些额外的扩展：

- 在此基础构建的分析工具；
- 开放的API。

2. 生产级链路追踪系统Dapper

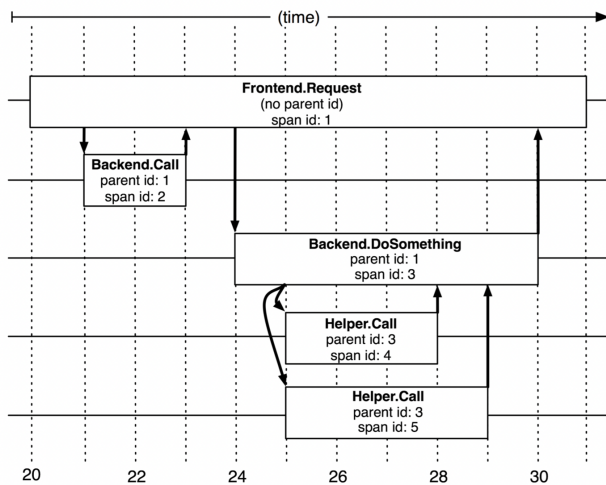
Dapper是经过生产实践的链路追踪系统，现在大多数的链路追踪系统都借鉴了Dapper。



链路追踪系统需要把链路经过的所有节点信息记录并串联到一个特定的请求上，的2种基本实现原理：

- block-box: 记录所有节点信息，通过统计回归技术，通常需要全局的逻辑时钟或者拓扑排序等确定事件的先后顺序；
- annotation: 使用一个额外的ID来记录，通过一个全局的ID串联，Dapper采用了这个方式。

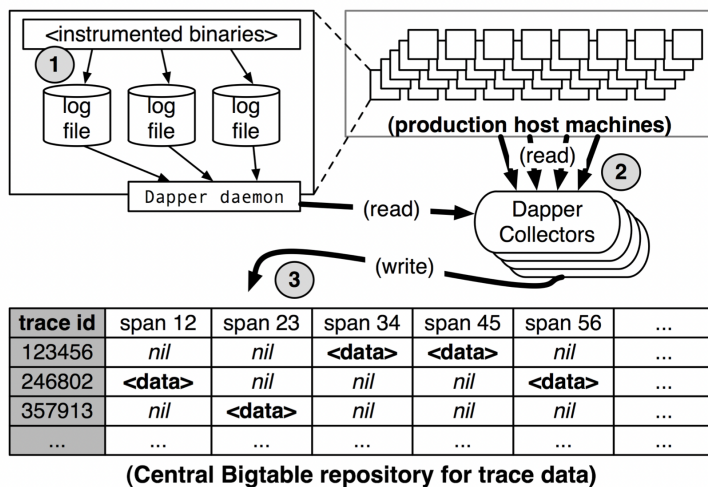
每一个请求对应一个Trace，Trace包含很多的Span，通常来说，每个RPC调用会开启一个Span，Span内可以记录Annotation，基本的原理如下：



Dapper称为生产级联路追踪系统的2个主要优势：

- 低消耗，提出了采样率的概念
- 对应用做到了高度的透明

2.1 采样率

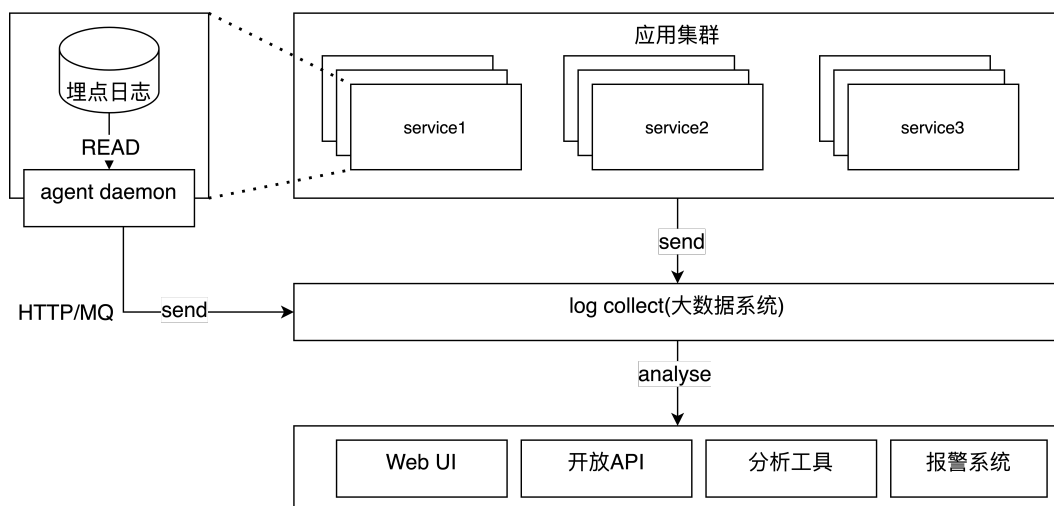


2.2 透明性

对通用组件库的改造，只有几百行，植入埋点的地点：

- HTTP请求使用ThreadLocal的方式保存Trace、Span、Annotation的信息；
- 对线程池的任务做特定的包装，ThreadLocal复制；
- RPC调用框架中植入。

3. 链路追踪系统的常见的架构

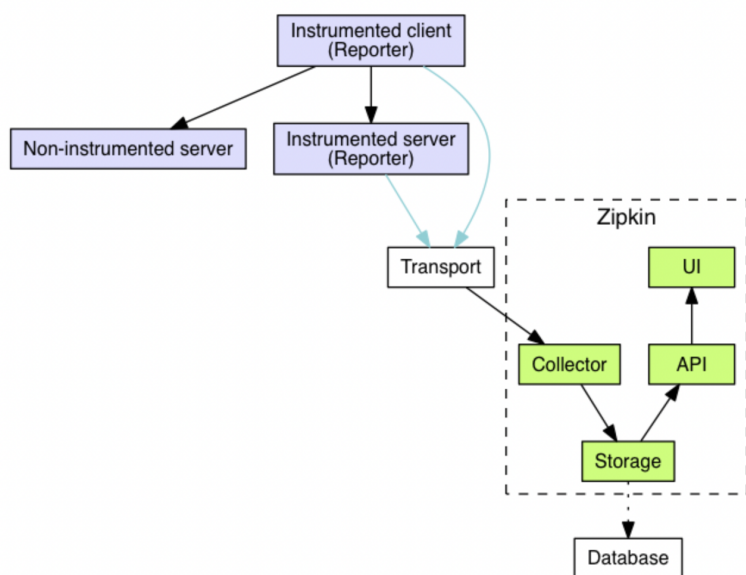


4. 流行的链路追踪系统

目前比较流行的开源的分布式链路追踪系统有如下几种。

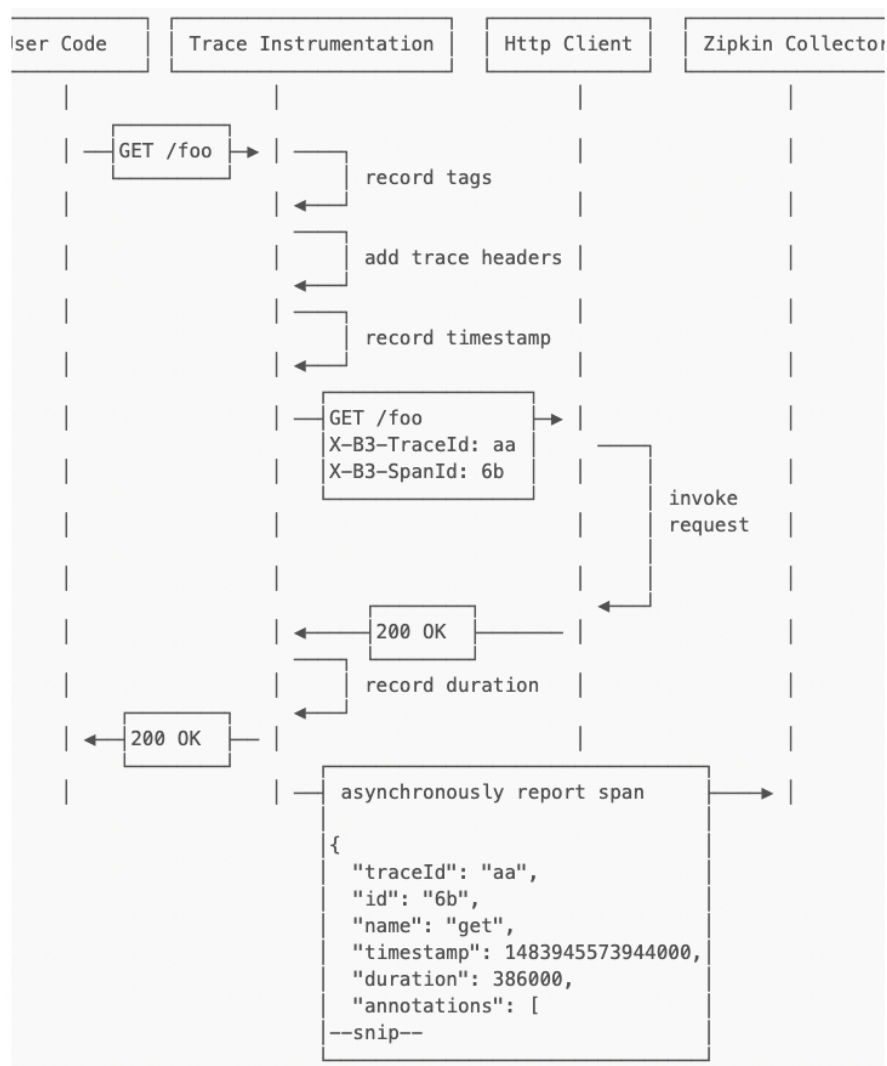
4.1 zipkin

由Twitter公司开源，开放源代码分布式的跟踪系统，用于收集服务的定时数据，以解决微服务架构中的延迟问题，包括：数据的收集、存储、查找和展现。架构如下：



主要分为4个部分：

- collector
- storage
- search
- web UI



4.2 PinPoint

韩国人开发的APM系统，基于google Dapper开发，pinpoint 有4个主要组件组成。

- HBase (用于存储)
- pinpoint Collector (收集数据)

- pinpoint Web (使用web 界面)
- pinpoint Agent (java 探针)注：修改Java的字节码

文档不是很全

4.3 SkyWalking

国产的优秀APM组件，2015年由个人吴晟（华为开发者）开源，2017年加入Apache孵化器；也是一个功能强大的APM工具，主要是面向云原生、容器化服务的分布式系统设计的。

SkyWalking由4个部分组成

- Probes：类似与agent的角色，探针还有对字节码的处理；
- Platform backend：数据聚合分析流处理等；
- Storage：存储组件，ES、MySQL等；
- UI：可视化组件。

4.4 CAT

CAT 作为服务端项目基础组件，提供了 Java, C/C++, Node.js, Python, Go 等多语言客户端，已经在美团点评的基础架构中间件框架（MVC框架，RPC框架，数据库框架，缓存框架等，消息队列，配置系统等）深度集成，为美团点评各业务线提供系统丰富的性能指标、健康状况、实时告警等，优势是提供的功能比较全面。

4.5 Spring Cloud Sleuth

Spring Cloud官方出的链路追踪组件，只用于数据收集，没有存储与展示的功能，方便Spring Boot的项目集成。使用spring-cloud-sleuth-zipkin可以与zipkin集成。使用简单，配置方便，也是基于Dapper的原理开发的。

4.6 系统对比

	Cat	Zipkin+Sleuth	SkyWalking	PinPoint
代码埋点方式	通用组件库	通用组件库	java探针	java探针
透明性	低	中	高	高
采集数据-机器	支持	不支持	支持	支持
采集数据-系统JVM，链路，Web	支持	支持	支持	支持
采集数据-业务监控	支持	支持	不支持	不支持
数据传输	netty	async HTTP/MQ	gRPC	thrift
粒度	代码	代码	方法	方法
性能影响	低	低	低	中
存储支持	MySQL/HDFS	Cassandra/MySQL/ES	ES/MySQL/TiDB	Hbase
应用层	强	弱	较强	较强

综上，如果需要底层监控需要选择SkyWalking与PinPoint，其中SkyWalking是更好的选择，是Apache孵化的项目，文档全，社区比较丰富，功能也比较强大，但是业务层的监控能力必较弱，Spring Cloud项目使有Sleuth方式配置简单，可以灵活的监控系统与业务层面的日志，但是功能比较单一，需要比较完整的监控功能，使用Cat是较好的。

5. Spring Cloud Sleuth

Spring Cloud Sleuth是基于OpenZipkin Brave实现的，OpenZipkin Brave是一个开源的链路追踪日志收集实现库，可以与Zipkin server集成做一个完整的链路追踪系统。Spring Cloud Sleuth也使用了Dapper中的Trace、Span、Annotation\Event的概念，Spring Cloud Sleuth还定义了4种事件：

- cs: Client Sent. 客户端发起请求，通常意味着一个创建一个新的Span；
- sr: Server Received: 服务端接受到请求并处理，sr时间戳- cs时间戳=网络延迟；
- ss: Server Sent. 服务端请求处理完毕，ss时间戳-sr时间戳=请求处理事件
- cr: Client Received. 接收到返回，cr-cs=完整的请求处理时间。

下图是Trace与Span在系统内的状态图，来自Spring Cloud Sleuth官网。

5.1 Spring Cloud微服务集成Sleuth的步骤

1. 添加POM

```
<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>

<dependency>
  <groupId>org.springframework.cloud</groupId>
  <artifactId>spring-cloud-starter-sleuth</artifactId>
</dependency>
```

2. 增加配置属性

```
#
spring:
  sleuth:
    sampler:
      rate: 30 //
      probability: 1.0 //
zipkin:
  base-url: http://localhost:9411//Zipkin
```

输出的日志中会有[application name,trace id, span id]的结构代表接入成功。

3. 安装Elasticsearch与Zipkin服务器

6. 总结

目前主流的分布式链路的追踪系统的设计与架构基本与Dapper是一致的，都主要分为了4个部分：

- agent或者java探针；
- 日志收集分析器；
- 存储；
- 分析工具与可视化UI。

在具体的服务层，各个链路追踪系统实现日志采样的方式有不同，有通过Java探针的方式实现，有通过拦截器的方式，有的需要机器上具有agent daemon，有的不需要。存储层目前都支持一些主流的存储中间件，比如Elasticsearch、MySQL等；分析工具与可视化UI等大同小异，新开发的一些链路追踪系统的分析工具与报警等APM功能更强大一些。由于时间有限，对于SkyWalking、PinPoint、Cat、ZipKin等没有做深入的了解，也是后面TODO的事情。