

1.术语(Terminology)

2.使用

Spring-Cloud-Sleuth是Spring Cloud的组成部分之一，为SpringCloud应用实现了一种分布式追踪解决方案，其兼容了Zipkin, HTrace和log-based追踪, 官网:https://cloud.spring.io/spring-cloud-sleuth/spring-cloud-sleuth.html#_running_examples

1.术语(Terminology)

Span: 基本工作单元，例如，在一个新建的span中发送一个RPC等同于发送一个回应请求给RPC，span通过一个64位ID唯一标识，trace以另一个64位ID表示，span还有其他数据信息，比如摘要、时间戳事件、关键值注释(tags)、span的ID、以及进度ID(通常是IP地址)

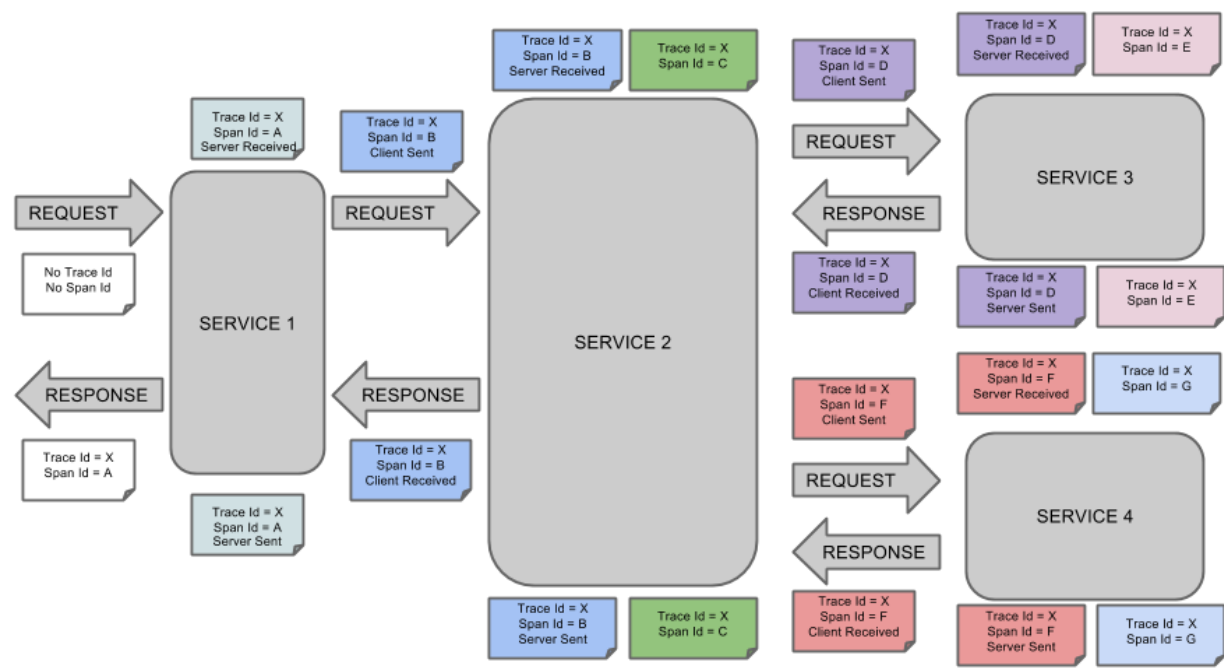
span在不断的启动和停止，同时记录了时间信息，当你创建了一个span，你必须在未来的某个时刻停止它。

Trace: 一系列spans组成的一个树状结构，例如，如果你正在跑一个分布式大数据工程，你可能需要创建一个trace。

Annotation: 用来及时记录一个事件的存在，一些核心annotations用来定义一个请求的开始和结束

- **cs - Client Sent** -客户端发起一个请求，这个annotation描述了这个span的开始
- **sr - Server Received** -服务端获得请求并准备开始处理它，如果将其sr减去cs时间戳便可得到网络延迟
- **ss - Server Sent** -注解表明请求处理的完成(当请求返回客户端)，如果ss减去sr时间戳便可得到服务端需要的处理请求时间
- **cr - Client Received** -表明span的结束，客户端成功接收到服务端的回复，如果cr减去cs时间戳便可得到客户端从服务端获取回复的所有所需时间

将Span和Trace在一个系统中使用Zipkin注解的过程图形化：



每个颜色的注解表明一个span (总计7个spans，从A到G)，如果在注解中有这样的信息：

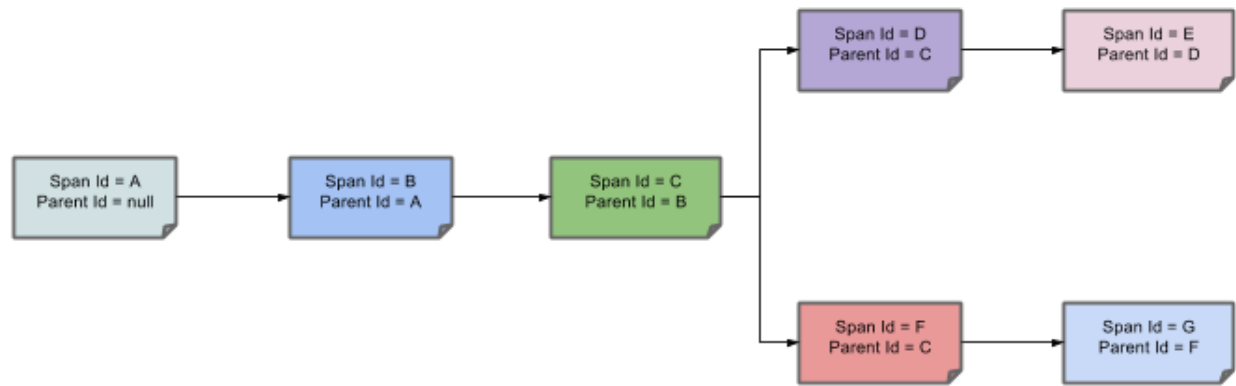
Trace Id = X

Span Id = D

Client Sent

这就表明当前span将Trace-Id设置为X，将Span-Id设置为D，同时它还表明了ClientSent事件。

spans 的parent/child关系图形化：



来源：<https://blog.csdn.net/u010257992/article/details/52474639>

2.使用

单独使用sleuth十分简单

pom.xml

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

得先有<parent>才能不写版本号

将日志格式改为

```
<layout class="ch.qos.logback.classic.PatternLayout">
  <pattern>%date [%thread] [%X{X-B3-TraceId:-},%X{X-B3-SpanId:-}] %-5level
%logger{80}:%line - %msg%n</pattern>
</layout>
```

把traceId和SpanId打印出来

然后就可以在日志中按照traceId和spanId找请求了, 如果日志收集到了es中, 也可以给traceId和spanId单独做个列, 用es来找也是可以的