**[学习微服务-第4天]ServiceComb+Zipkin**

**分布式调用链追踪**能有效地监控服务间的网络延时并可视化微服务中的数据流转。ServiceComb扩展了zipkin的接口提供了服务内部的链路调用信息，能提供更完整的调用链路信息，更容易定位问题和潜在性能问题。

本文将介绍ServiceComb 提供的分布式调用链追踪能力及使用指导。

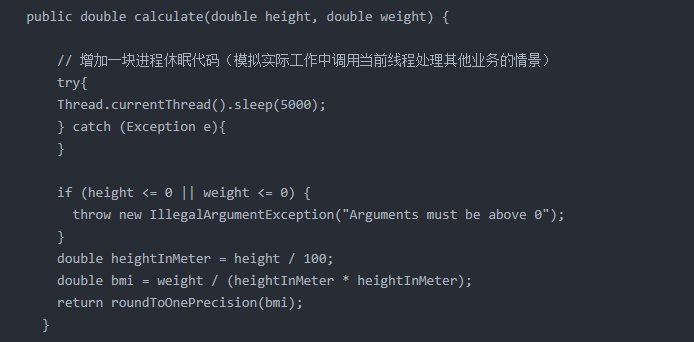
**一. 异常场景示例**

我们将使用ServiceComb的入门案例BMI（体质指数应用），展示ServiceComb的调用链追踪和自定义调用链追踪能力。BMI应用程序包含体质指数计算calculator和服务网关webapp两个服务。

参见ServiceComb的入门案例BMI指导

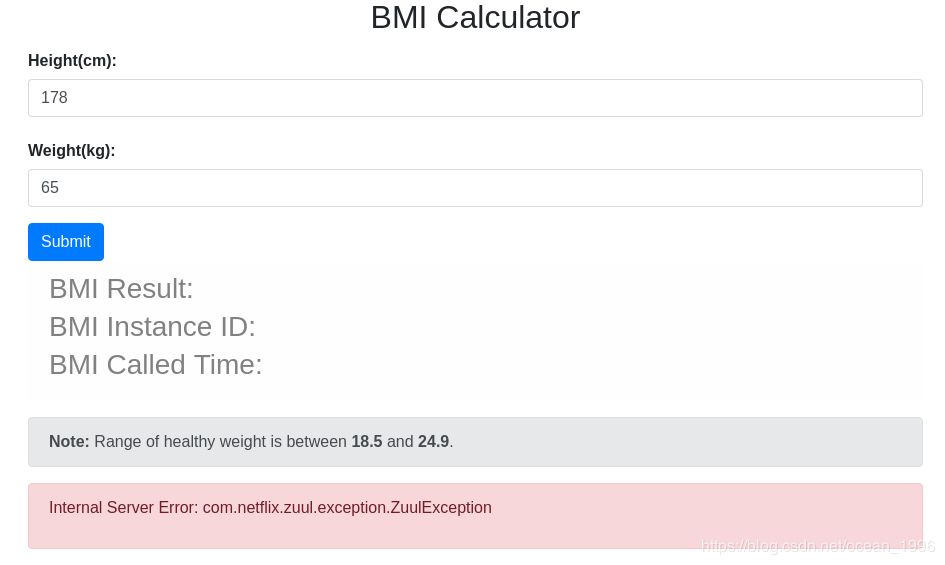
http://servicecomb.apache.org/cn/docs/quick-start/

我们给BMI的体质计算服务增加了异常代码，如下。



参见BMI程序使用指导1

http://servicecomb.apache.org/cn/docs/quick-start/，运行bmi程序，出现如下异常结果。





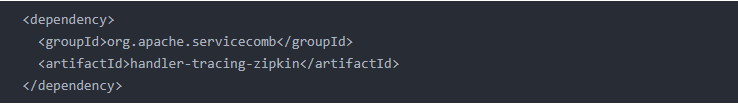
**二. 使用Zipkin定位服务级别异常**

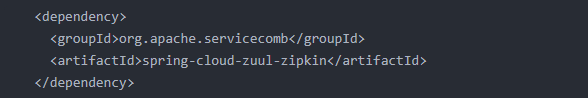
以上的BMI示例还未开启调用链追踪，下面我们将使用ServiceComb提供的分布式调用链追踪能力定位分析BMI应用的哪个服务发生了异常。

首先需要给BMI程序配置zipkin调用链追踪能力，只需添加两个配置即可。可参考分布式追踪2和Distributed Tracing with ServiceComb and Zipkin3。

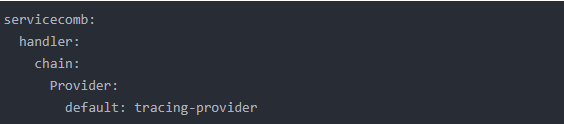
**1.添加依赖**

在 calculator微服务的pom.xml文件中添加依赖项：



在 webapp微服务的pom.xml文件中添加依赖项：

**2 .配置跟踪处理程序**  
在 calculator微服务的microservice.yaml文件中添加分布式追踪的处理链：



到此已配置完毕，您只需执行以下几步即可运行↓↓↓

**使用Docker运行Zipkin分布式追踪服务：**

https://mmbiz.qpic.cn/mmbiz_png/jaoydu4O9YdN2wW10Wb29zCziafWU8zuO6ibZ4VdTGfzFFlniaz5BBH64wd2tnafZrO52PUhxZzDjosO1qm8q75wQ/640?wx_fmt=png&tp5&wx_lazy=1&wx_co=1



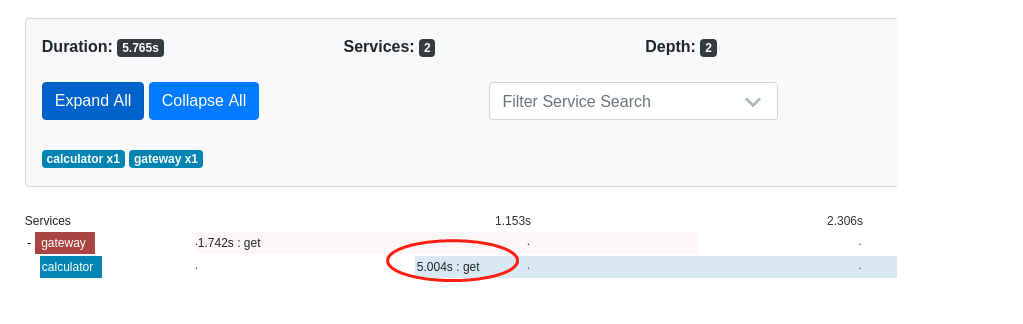
**重启 calculator 微服务：**

https://mmbiz.qpic.cn/mmbiz_png/jaoydu4O9YdN2wW10Wb29zCziafWU8zuOpQvtP0aFcUsdzegGVqGia5qpwZ6iaHuNCliczUvQ6kqKTJQw7oSYlLXGw/640?wx_fmt=png&tp5&wx_lazy=1&wx_co=1

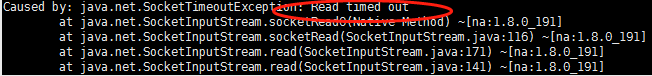
**重启 webapp 微服务：**

https://mmbiz.qpic.cn/mmbiz_png/jaoydu4O9YdN2wW10Wb29zCziafWU8zuOHiaZjWibyf919tnGib6JBtmdFwuXd2RiaKgw7STtLKeMJjLO0mMvIiaEDDw/640?wx_fmt=png&tp5&wx_lazy=1&wx_co=1

打开浏览器访问 http://localhost:9411 ，查看调用链追踪情况，结果如下。可以看到gateway服务在1.7S时就返回给了用户失败结果，而calculator服务却运行了5S的时间，但calculator为蓝色（并非内部错误），可以确定calculator 服务超时导致gateway超时异常。



我们通过在后台服务打印程序调用栈日志可以看到，异常是由于socket读取超时，可以印证我们在Zipkin中看到的结果。





**三、使用自定义追踪功能定位到函数级别**

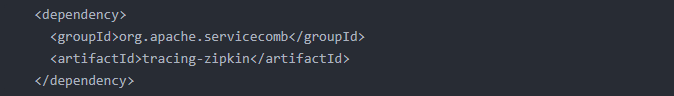
以上已确定导致超时返回的是calculator服务，在服务应用代码较大时，会较难找到具体出现异常的地点，ServiceComb支持可以在服务内部进行函数级别的调用链打点追踪，这将可以帮助我们解决这个问题。

配置BMI程序的自定义追踪功能，**只需三步即可**。

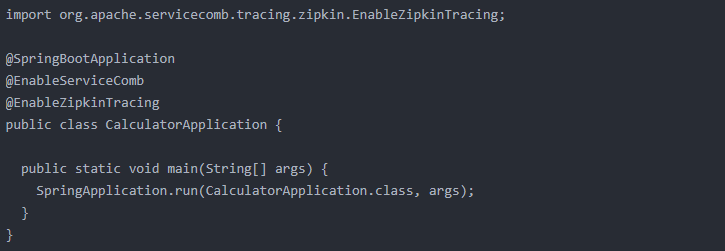
参考java-chassis使用手册—自定义调用链打点4

https://docs.servicecomb.io/java-chassis/zh\_CN/general-development/customized-tracing.html

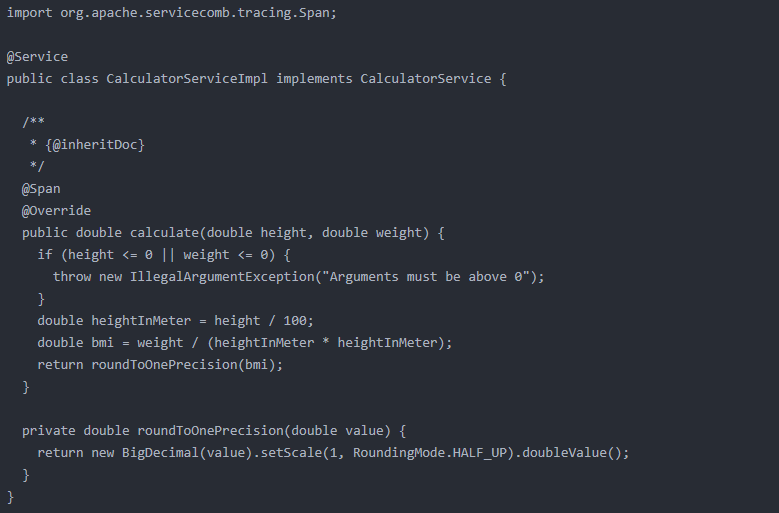
在体质指数计算器的pom.xml文件中添加依赖项：

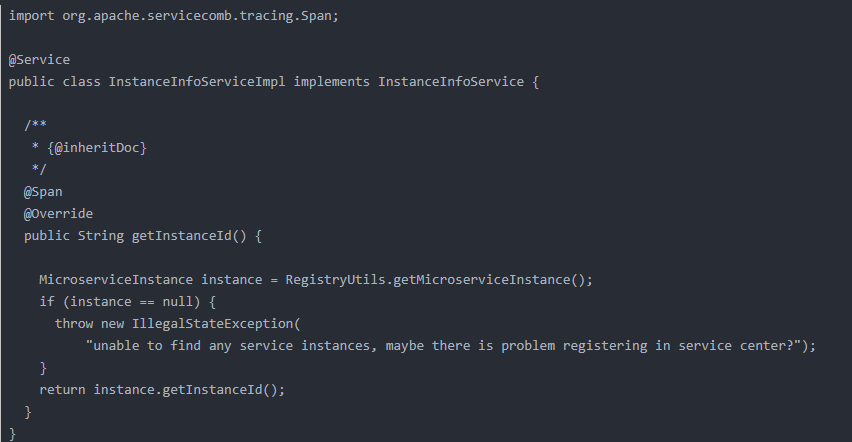


在程序入口或者配置处添加 @EnableZipkinTracing 注解

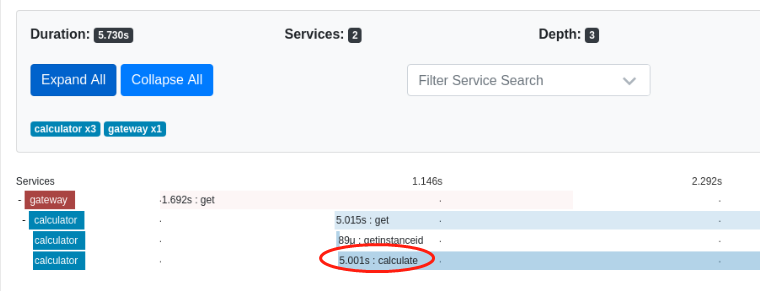


在服务程序中的调用方法处添加 @Span 注解





至此，自定义函数级别打点配置完成。  
运行程序，从Zipkin界面如下：



可以看到，调用链中新增加了calculate函数这个Span（上图红圈处），耗时5S。  
点开该Span，可以查看该函数的详细信息，我们可以根据该信息准确查找到对应的源码进行修正。

