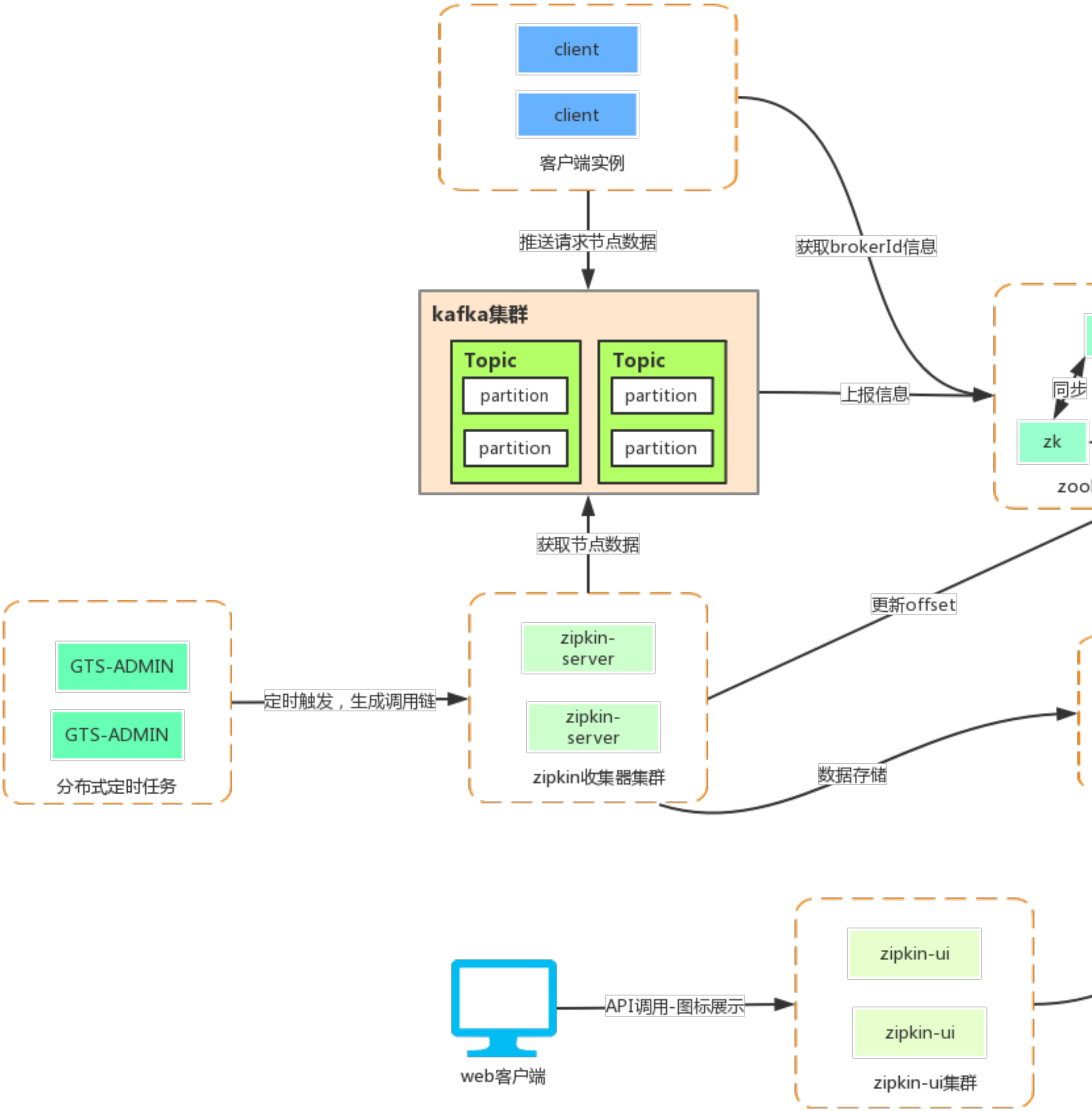


sleuth+zipkin+kafka+logstash链路追踪二次开发方案

原创 sharedCode 最后发布于2019-07-01 14:31:27 阅读数 901 ☆ 收藏

系统架构方案

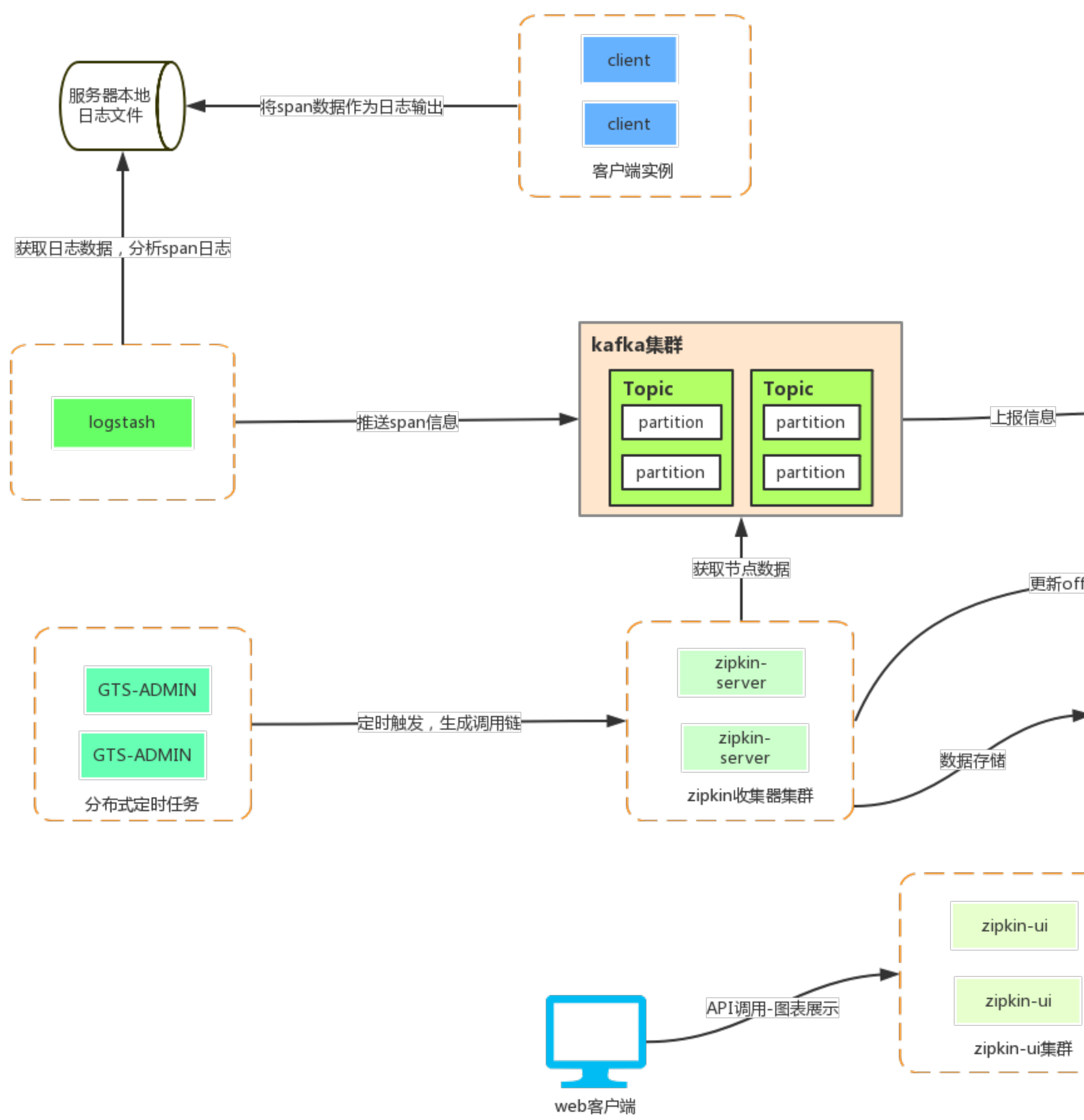
方案一



架构说明：

- 1.应用接入zipkin客户端，将span的信息直接推送给kafka
- 2.zipkin-server定kafka中订阅主体为sleuth的消息，将span中的信息推送到elasticsearch中
- 3.zipkin-ui项目负责从elasticsearch读取信息，分析信息，呈现图标给用户。

方案二



架构说明：

- 1.接入链路追踪的客户端，仅需对本地日志输出进行配置，输出span的信息到本地文件里面
- 2.部署logstash，读取应用的span日志信息，然后将span信息发送给kafka
- 3.zipkin-server通过消费kafka中，topic为“sleuth”的消息，将span信息存入elasticsearch

4.zipkin-ui项目负责从elasticsearch读取信息，分析信息，呈现图标给用户。

通过logstash分析本地日志的方式，对应用的侵入性，性能影响，高可用的影响 都是降到最低的。

比较

方案一	方案一	
优点	部署复杂度较低，开发成本相对较低	应用的侵入性，性能
缺点	当kafka或者zookeeper宕机时，已经启动的应用没有影响，对于正在启动的应用会启动失败	

功能设计

首页

zipkin分布式链路追踪



系统吞吐量TOP10

1.MSG-MS

接口最慢TOP10

1./get/list	5945.12ms
2./get/list	5945.12ms
3./get/list	5945.12ms

应用最慢TOP10

- 1.MSG-MS
- 2.ACS-MS
- 3.AIM-MS
- 4.GTS-ADMIN
- 5.MSG-MS
- 6.ACS-MS
- 7.ACS-MS
- 8.MSG-MS

总应用数

接入全链路追踪系统的总应用数

总接口数

链路追踪系统所监测到的总接口数

错误数

最近15分钟之内发生的接口调用错误数量

系统吞吐量

以应用为维度，计算15分钟内的总请求数，求出每秒钟处理的接口数量。当然也可以是前一分钟内的请求总数，然后求出每秒

接口最慢TOP10

查询所追踪到的系统，根据相应时间，最近15分钟内，平均响应时间最慢的接口前10

应用最慢TOP10

以应用为维度，查询应用在这15分钟内，所有的请求数据，求出平均响应时间，得到最慢的10个应用。

接口分析

接口列表

Zipkin 链路追踪系统

查找调用链

接口分析

全局应用依赖

根据ID查找

服务：

uas-ms

接口：

http:/poi/list-with-

调用时间：

09-19-2018 11:59

 至

09-20-2018 11:59

查询

显示

接口地址	服务名	请求次数	平均响应时间	错误次数	错误率
/permission/switch/record/record	bas-ms	1000	300ms	100	10%

接口分析提供两种 展现形式

1. 列表展示

显示 接口地址，服务名，请求次数，平均响应时间，错误次数，错误率这六个字段

点击 查看调用链 这个按钮可以跳转到图形调用链，并直接显示当前接口的调用链。

2. 图形调用链

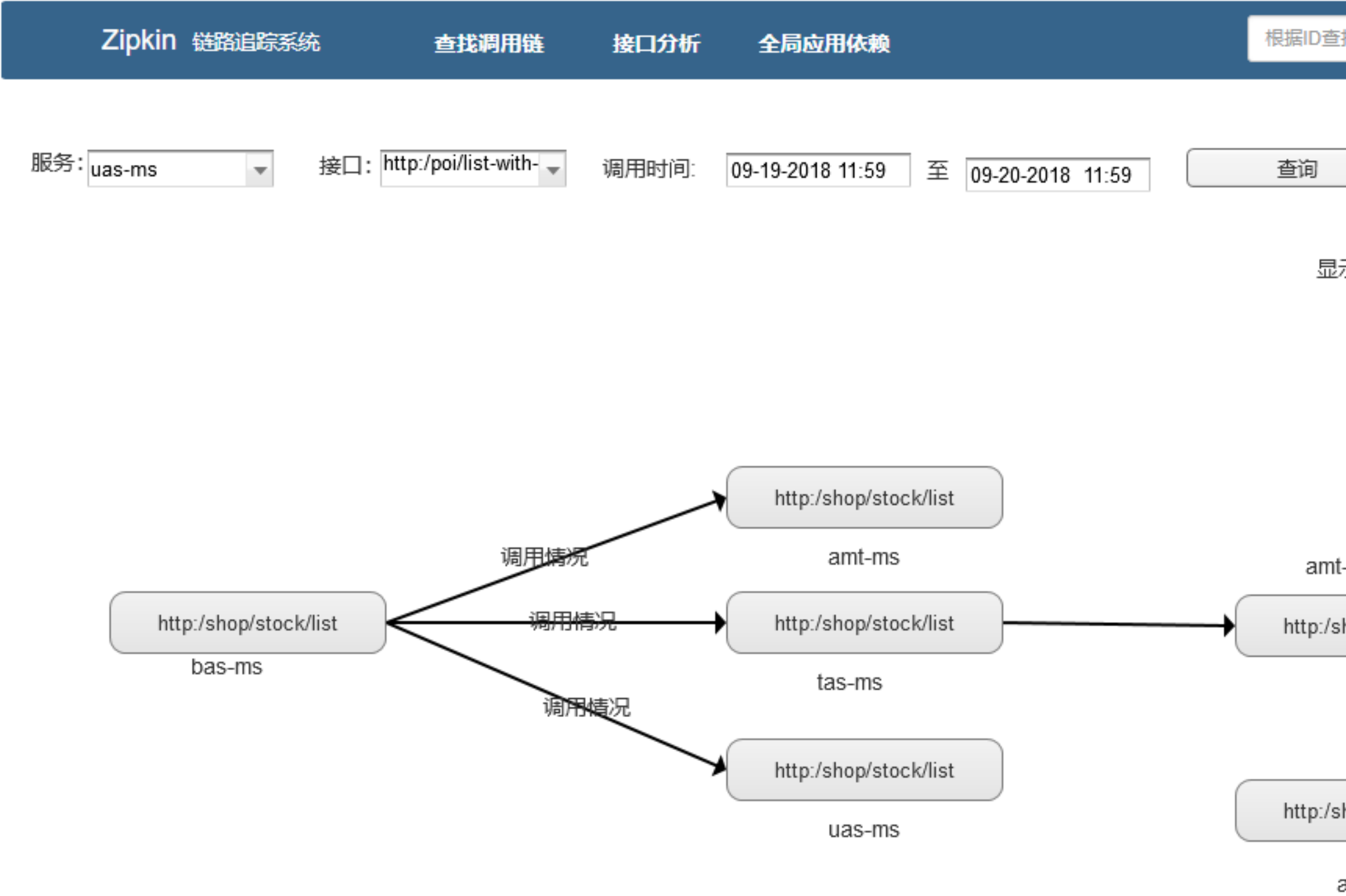
通过拓扑图的方式暂时接口调用情况

该页面，显示的是所有接口的调用次数，平均响应时间，错误次数等信息，可以看做是接口的全局预览。

可以通过 应用名，接口地址进行搜索， 点击列表页右侧的查看按钮，可以查看该接口的调用链

接口调用链

当前接口的颜色要进行区分， 查询时间要进行限制，不可以查询所有数据，后期数量量较大



使用长框的形式展示，每个接口下面，需要注明所属服务。

接口和接口之间的连接线上需要注明调用次数，失败次数

traceld管理

traceld列表

可以通过服务名，接口地址，开始时间，结束时间，响应时间，traceld进行查询， 点击相应的traceld， 可以进入详情页。

zipkin分布式链路追踪

服务名:

接口地址:

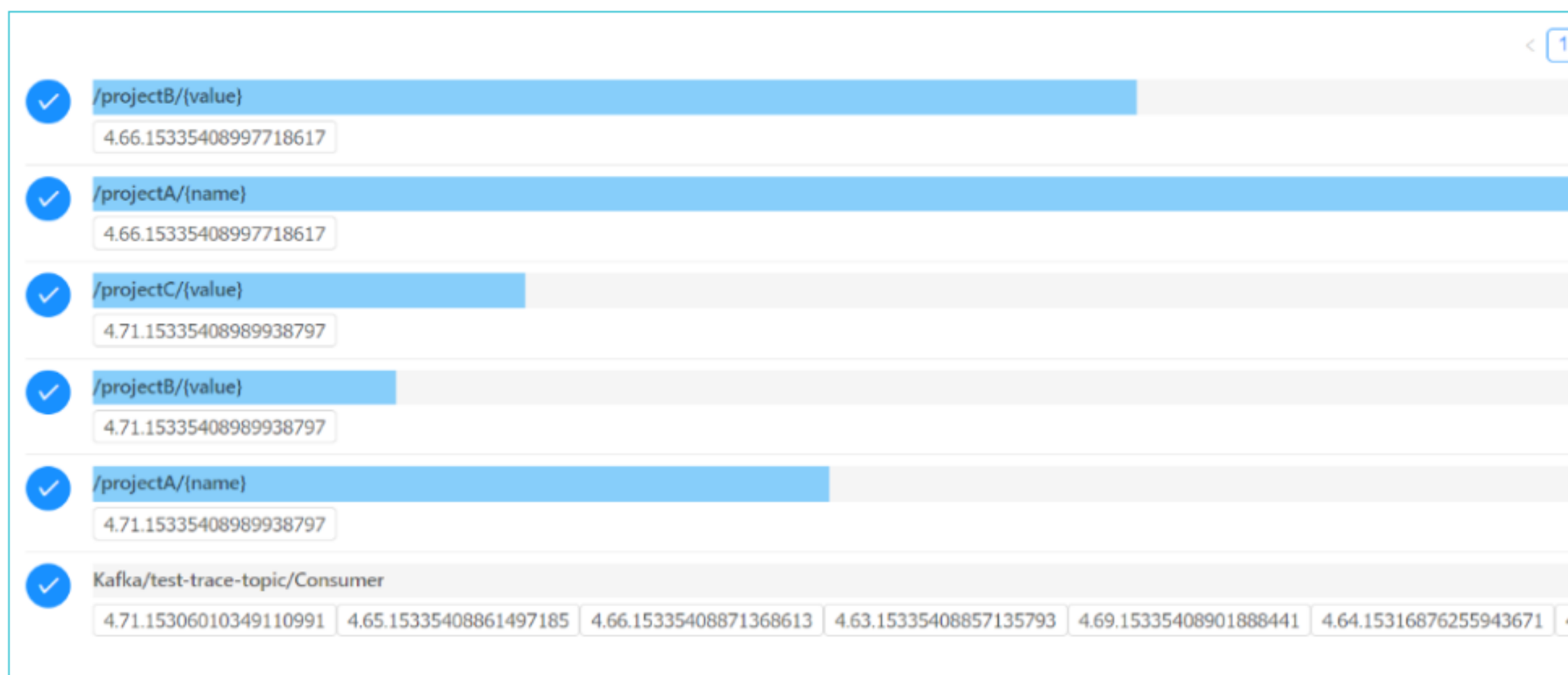
开始时间:

结束时间:

响应时间大于:

traceld:

查询



traceld 详情

页面样式可以参考原生的，显示的就是这个traceId关联的调用链

全局调用链

sharedCode源码交流群， 欢迎喜欢阅读源码的朋友加群， 添加下面的微信， 备注”加群“。