

Dubbo与SpringCloud的Ribbon、Hystrix、Feign的优劣势比较

在微服务架构中，分布式通信、分布式事务、分布式锁等问题是亟待解决的几个重要问题。

Spring Cloud是一套完整的微服务解决方案，基于 Spring Boot 框架。确切的说，Spring Cloud是一个大容器（而不是一个框架），它可以将通过集成一些好的微服务框架，从而简化开发者的代码量。

Dubbo 是阿里开源的分布式通信框架，专注于通信服务治理，类似于Spring Cloud中 Ribbon、Hystrix、Feign等核心组件的功能。

下面，我们着重来聊下这两种处理方式有何不同。

01 协议处理

1) Spring Cloud更加优雅简单

Feign使用Http进行传输。

Feign 集成了Ribbon，并且嵌入了Spring cloud全家桶，通过简单配置，就能在分布式里面实现服务间的调用，类似于Bean 调用。

2) Dubbo方式更灵活

Dubbo协议可选，大部分情况使用Dubbo传输协议，也可以使用http协议。

从协议层选择看，Dubbo是配置化的，更加灵活。

Dubbo 协议更适合小数据高并发场景。

02 性能方面

1) Spring Cloud性能调优

Feign 在高并发场景下，通常需要进行如下性能优化，有明显瓶颈，需要改造。

- 调整服务容器到 Undertow，在负载大的情况下Undertow 的性能有提高；
- 曾有同学表示将HTTPURLConnection 改成 HttpClient /Okhttp，这样可以优化性能，其实单次调用性能HttpClient差很多，HttpClient 因为封装了很多方便开发者处理的方法，性能比HTTPURLConnection 差，改进同时HttpClient 需要设置复用连接池，效果可见一般；
- 开启Gzip；
- Feign中HttpMessageConverters 默认使用jackson2方式进行序列化和反序列化，可以将其改造为ProtoBuf，降低Cpu 损耗并且响应时间也降低。

2) Dubbo性能调优

主要是配置而无需改造。

03 负载均衡

1) Ribbon 的负载均衡策略

- 随机；
- 规则轮询；
- 空闲策略；
- 响应时间策略。

Feign默认使用Ribbon作为负载均衡的组件，Ribbon需要进行全局配置，个性化配置比较麻烦。

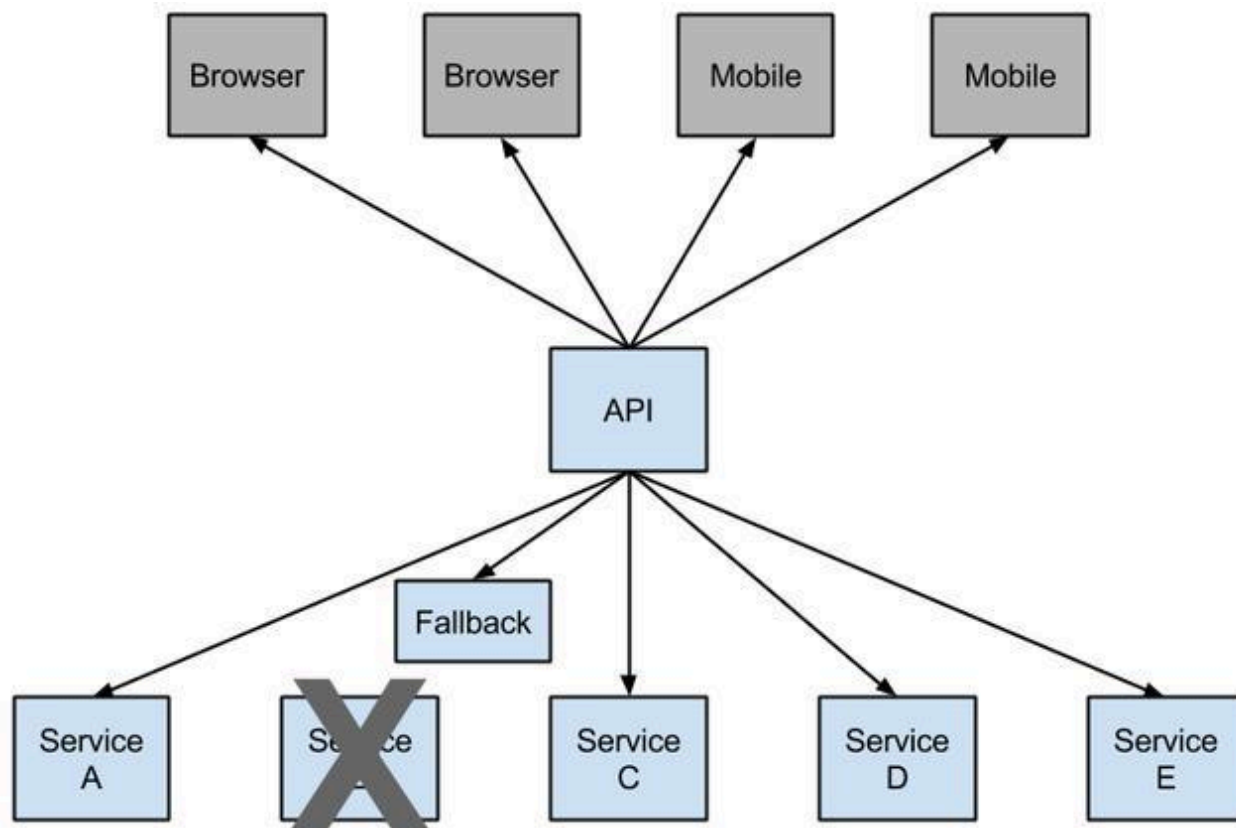
2) Dubbo 的负载均衡策略

- 随机；
- 权重轮询；
- 最少活跃调用数；
- 一致性Hash策略。

Dubbo 可以使用路由策略，然后再进行负载均衡。

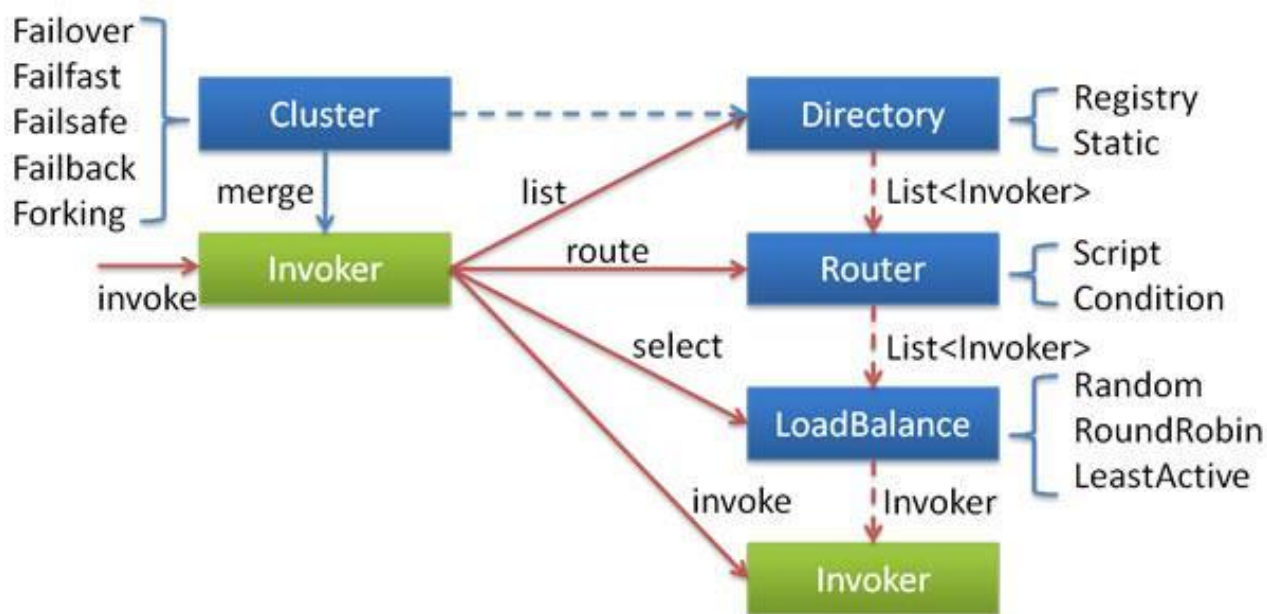
04 容错机制

Spring cloud 的 Hystrix 提供了服务降级，服务熔断，依赖隔离，监控(Hystrix Dashboard)等功能。



图片为引用，来源于官方图片

Dubbo 提供了一整套 FailOver、FailFast、Failsafe、FailBack、Available、Broadcast、Forking 策略，以及Mock



图片为引用，来源于官方图片

05 路由、流量调度、ABtest

1) Ribbon需自己实现，应用不灵活

Ribbon主要通过扩展 AbstractLoadBalancerRule负载均衡的方法来实现，在负载均衡的部分还要进行改造升级。

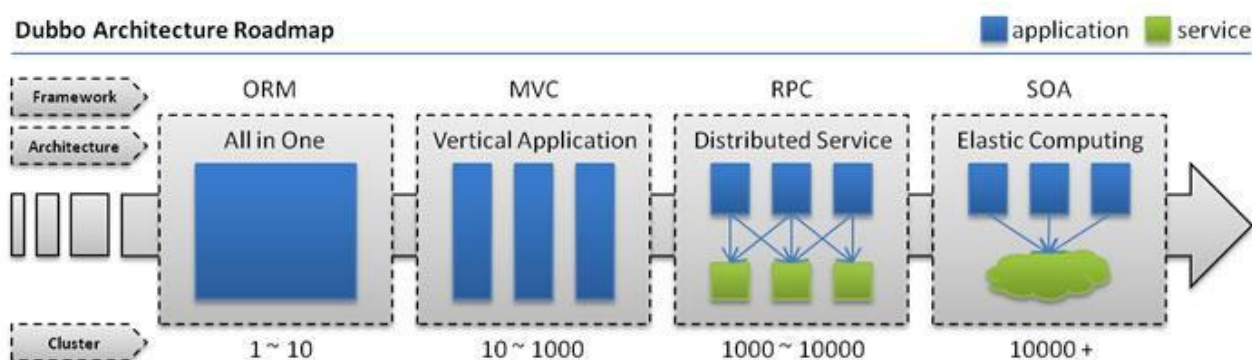
2) Dubbo更加灵活方便

Dubbo通过界面化、校本化配置路由规则，可以实现灰度发布、动态流量调度、容量计算等，方案成熟。

另外，Dubbo 还支持多版本调用。

06 总结

正如Dubbo的介绍(如下图)，业务发展影响着架构的选型，当服务数量不是很大时，使用普通的分布式RPC架构即可，当服务数量增长到一定数据，需要进行服务治理时，就需要考虑使用流式计算架构。



图片为引用，来源于官方图片

Dubbo可以方便的做更精细化的流量调度，服务结构治理的方案成熟，适合生产上使用，虽然Dubbo是尘封后重新开启，但这并不影响其技术价值。

Ribbon Hystrix Feign在服务治理中，配合Spring Cloud做微服务，使用上有很多优势，社区也比较活跃，看将来更新发展。