

# 为什么选择 SpringCloud? 不选择 Dubbo

Dubbo 属于 RPC 框架

SpringCloud 属于微服务全家桶框架 帮助 在微服务架构中遇到的难题都会提供对应框架 来进行解决 openfeign 才是属于 RPC 框架。

- 1.分布式服务注册中心（服务治理） Eureka、Zookeeper、Consule、Nacos、Redis、数据库等;
- 2.分布式配置中心 SpringCloud Config、携程阿波罗、Nacos Config;
- 分布式事务解决方案(MQ 最终一致性/LCN(已经淘汰)/ Seata（阿里背书）)
- 分布式任务调度平台(xxl-job、elastic job、阿里巴巴 Scheduler)
- 5.分布式日志采集系统 ELK+Kafka
- 6.分布式服务追踪与调用链 Zipkin、skywalking 等。
- 7.分布式锁(Redis（Redisson）/Zookeeper（Curator）实现分布式锁)
- 8.服务的接口保护(hystrix/sentinel)

## 服务注册中心框架哪些

Eureka、Zookeeper、Consule、Nacos、Redis、数据库

Nacos 首先 2.0 Ap 和 CP 模式

## 谈谈服务注册中心设计原理？

- 1.服务注册：服务注册中心 Server 端提供 服务注册接口（http 协议），当生产者项目启动时，会将它的接口地址注册到注册中心上（生产者启动时调用注册中心接口 传递服务名称和接口地址 到注册中心 存放在 map 集合中）  
Nacos 整合：配置文件填写 注册中心地址
- 2.服务发现：整合负载均衡器 消费者 根据服务的名称从 注册中心获取该集群地址 让后在通过负载均衡器 本地计算 选出一个接口地址 实现本地 rpc 远程调用  
服务剔除： 注册中心每间隔一段时间 例如 5s 检查 注册中心地址 是否有续约，如果没有续约的话 则会将该地址直接从我们的 注册中心移除。 我们生产者每隔一段时间 发送心跳给我们注册中心 告诉 当前地址还是存活的状态。如果注册中心 在一段时间内 没有收到该地址发送过来的心跳，则认为该地址已经宕机 直接从 map 集合中移除。  
Debug 模式停止 springboot 项目（正常模式停止 springboot 项目）  
（非正常模式停止 springboot 项目）

正常停止 springboot 项目：走容器销毁 回调方法，在回调方法里面  
主动发送请求到达注册中心 将该服务对应的地址从 map 集合中移除。  
非正常停止 springboot 项目：不会走回调方法 主动通知注册中心删除该地址。  
可以通过服务剔除来进行实现。  
3.服务治理：解决我们在 RPC 远程调用的过程 地址管理问题。

## 消费者获取到一个宕机地址？如何解决？

回答：

为什么消费者会获取到一个宕机的地址呢：

- 1.非正常停止 springboot 项目：不会走回调方法 不会主动通知注册中心删除该地址。  
可以通过服务剔除来进行实现。
- 2.注册中心每间隔一段时间 例如 5s 检查 注册中心地址 是否有续约  
，如果没有续约的话 则会将该地址直接从我们的 注册中心移除。 我们生产者
- 3.每隔一段时间 发送心跳给我们注册中心 告诉 当前地址还是存活的状态。
- 4.如果注册中心 在一段时间内 没有收到该地址发送过来的心跳，则认为该地址已经宕机 直接从 map 集合中移除。  
通过负载均衡算法 直接故障转移获取下一个地址。