

## Dubbo和Spring Cloud有什么区别？

说真的，这两个东西没有可比性，Dubbo最开始是一个可扩展的RPC调用框架，在Dubbo里一次调用涉及到的服务路由、负载均衡、序列化机制、网络传输协议等等都是可以扩展的，具体的性能取决于所选用的组件，同样Spring Cloud也类似，所以我们不能站在性能的角度来对比两个框架。

其次，作为框架，要对比我们也应该对比这个框架的可扩展性，Dubbo的可扩展性是不要比Spring Cloud好的。而Spring Cloud目前的优势是组件比较齐全，比如有服务网关、分布式配置中心、服务跟踪等等，而这些Dubbo暂时还没有，但是在Dubbo生态里是准备加上的。

## Dubbo2.7有些什么新功能？

1. 动态配置中心 ( configcenter )
2. 元数据中心
3. 条件路由支持应用
4. 标签路由

## Dubbo支持哪些通信协议，不同的协议分别适用哪些场景

1. dubbo://，框架的默认协议，采用单一长连接和 NIO 异步通讯，适合于小数据量大并发的服务调用。
2. http://，使用短连接同步传输，使用数据量不确定，并且需要在js或浏览器中使用。

## Http长连接和短连接

Http1.0，不支持长连接

Http1.1，默认为长连接，Connection=keep-alive

但是，说Http长连接并不准确，因为http是应用层协议，tcp才是传输层协议，只有传输层才能说建立连接与关闭连接。

那么，长连接有什么好处，长连接就是可以保持连接，那当我打开一个网页后，实际上我通常要过会才有可能去请求其他东西，其实并不需要这个连接一直保持在这，从这个角度看，似乎使用短连接比较合适。

长连接，实际上是指tcp长连接，那么http可以在这个连接的基础上发送多次http请求与接收多次http响应，实际当我们打开一个网页时，还需要请求很多js，css等资源，这个时候这些请求是可以复用tcp连接的，从而节省了网络资源。

那么，长连接什么时候才关闭？有一个超时时间的，在header进行设置，当这段时间内没有任何http请求则超时后自动关闭。

## Dubbo支持的集群容错方案

1. Failover Cluster：失败自动切换，当出现失败，重试其它服务器。通常用于读操作，但重试会带来更长延迟。
2. Failfast Cluster：快速失败，只发起一次调用，失败立即报错。通常用于非幂等性的写操作，比如新增记录。
3. Failsafe Cluster：失败安全，出现异常时，直接忽略。通常用于写入审计日志等操作。
4. Failback Cluster：失败自动恢复，后台记录失败请求，定时重发。通常用于消息通知操作。
5. Forking Cluster：并行调用多个服务器，只要一个成功即返回。通常用于实时性要求较高的读操作，但需要浪费更多服务资源
6. Broadcast Cluster：广播调用所有提供者，逐个调用，任意一台报错则报错。通常用于通知所有提供者更新缓存或日志等本地资源信息。

## Dubbo支持的负载均衡策略

1. Random LoadBalance：随机
2. RoundRobin LoadBalance：轮询
3. LeastActive LoadBalance：最少活跃调用数
4. ConsistentHash LoadBalance：一致性 Hash，相同参数的请求总是发到同一提供者。

## Dubbo中的服务降级是如何实现的

Dubbo中的服务降级实际上使用的就是Dubbo中的Mock机制，比如当调用某个服务失败后，可以设置mock=fail:return+null表示调用服务失败后返回null。

## 服务提供者能实现失效踢出是什么原理？

服务失效踢出基于 Zookeeper 的临时节点原理。

## Dubbo服务暴露的过程

首先Dubbo会通过DubboNamespanceHandler解析dubbo:service/标签，并且Spring会完成Bean的实例化，随后发布ContextRefreshEvent事件，会调用ServiceBean中的export方法，以这个方法为入口然后进行服务暴露。实际上，我们应该叫服务导出，因为服务导出包括服务注册和服务暴露，导出顺序也是先进行服务注册在进行服务暴露，服务注册就是将服务信息（包括接口名以及对于的协议等信息）注册到注册中心，然后进行服务信息监听器绑定，然后进行服务暴露（其实就是启动tomcat或nettyserver）。

## Dubbo服务引入的过程

首先Dubbo会通过DubboNamespanceHandler解析dubbo:reference/标签，并且Spring会完成Bean的实例化，在引入该Bean的时候，会调用ReferenceBean的getObject方法，以这个方法为入口然后进行服务引入，服务引入的步骤为：先根据服务名从注册中心找到所有的服务提供者并且封装到服务目录中，然后进行服务路由的初始化，然后进行监听器绑定，然后将服务目录封装为Cluster，最后生成代理类。实际上服务引入是相当复杂的，这里只能面试大概的步骤。

## Dubbo服务调用的过程

消费者：

1. Mock机制
2. 服务路由
3. 负载均衡
4. Filter过滤
5. 发送请求

服务者：

1. 接收请求
2. Filter过滤
3. 执行具体的实现类
4. 返回结果