### dubbo的功能列表

1、启动时检查。启动时会去检查所依赖的服务是否可用。  
2、集群容错。提供几种容错策略【可扩展】：  
（1）快速失败，只要有错误，就立即返回失败  
（2）失效转移，第一次调用失败，则重新选取一个节点重试。  
（3） 失败完全，调用失败时，忽略异常。  
（4）失败自动恢复，后台记录失败请求，定时重发。  
（5）并行调用多个服务器，只要一个成功即返回。  
（6）广播调用所有提供者，逐个调用，任意一台报错则报错。  
3、负载均衡【可扩展】，在集群负载均衡时，Dubbo 提供了多种均衡策略，缺省为 random 随机调用  
（1）Random LoadBalance 随机，按权重设置随机概率。  
（2）RoundRobin LoadBalance 轮循，按公约后的权重设置轮循比率。  
（3）LeastActive LoadBalance 最少活跃调用数，相同活跃数的随机，活跃数指调用前后计数差。  
（4）ConsistentHash LoadBalance 一致性 Hash，相同参数的请求总是发到同一提供者。  
4、直连服务提供者。测试环境时，可以绕过注册中心，直接连接服务提供者。  
5、只消费服务，不发布服务  
6、只注册服务到注册中心，而不消费该注册中心的服务。  
7、提供多种协议的暴露方式【可扩展】。  
8、多注册中心。一个服务可以注册到多个注册中心。【可扩展】  
9、服务分组。  
10、多版本。为了服务的兼容性。  
11、结果缓存。可以配置将服务调用的结果缓存在服务消费端。提高性能。  
12、泛化调用。泛化接口调用方式主要用于客户端没有 API 接口及模型类元的情况，参数及返回值中的所有 POJO 均用 Map 表示，通常用于框架集成，比如：实现一个通用的服务测试框架，可通过 GenericService 调用所有服务实现。  
13、回声测试。服务的健康性检查。  
14、服务调用时，提供上下文。上下文中存放的是当前调用过程中所需的环境信息。所有配置信息都将转换为 URL 的参数。RpcContext 是一个 ThreadLocal 的临时状态记录器，当接收到 RPC 请求，或发起 RPC 请求时，RpcContext 的状态都会变化。比如：A 调 B，B 再调 C，则 B 机器上，在 B 调 C 之前，RpcContext 记录的是 A 调 B 的信息，在 B 调 C 之后，RpcContext 记录的是 B 调 C 的信息。

15、同步调用，异步调用。  
16、参数回调。参数回调方式与调用本地 callback 或 listener 相同，只需要在 Spring 的配置文件中声明哪个参数是 callback 类型即可。Dubbo 将基于长连接生成反向代理，这样就可以从服务器端调用客户端逻辑。  
17、事件通知。在调用之前、调用之后、出现异常时，会触发 oninvoke、onreturn、onthrow 三个事件，可以配置当事件发生时，通知哪个类的哪个方法  
18、服务降级。  
19、服务延迟发布。  
20、并发控制。针对单个服务。【与qps相关】  
21、连接控制。针对服务器的连接数。  
22、服务鉴权  
23、路由规则。相当于白名单配置。