# Dubbo

dubbo,是一个分布式服务框架，以及SOA（SOA：面向服务的体系结构。它将应用程序的不同功能单元成为服务）治理方案。主要功能有：高性能NIO（nio即非阻塞，通常情况下服务方需要非阻塞方式提高效率，）通讯以及多协议集成，服务动态寻址与路由，软负载均衡和容错，依赖分析和降级等。

1. dubbo使用场景，当拆分应用进行服务化，提高开发效率，调优性能，节省关键资源时。
2. dubbo采用微核+插件体系，方便二次开发，适应性强。
3. dubbo运行在java1.5以上版本，缺省依赖javassist,netty.spring等包，但不是必须依赖。除了jdk外可以不依赖任何第三方库运行。
4. dubbo通过长连接减少握手，通过nio及线程池在单链接上并发拼包处理信息。利用二进制流压缩数据。
5. dubbo主要针对内部调用所以在安全上较差，dubbo通过token令牌防止用户绕过注册中心直连，在注册中心上管理授权。Dubbo还能通过黑白名单来控制服务所允许的调用方。
6. dubbo不支持(DTP)分布式事务。

Dubbo的核心功能：

**远程通讯**：提供对多种基于长连接的NIO（new IO）框架抽象封装，包括多种线程模式，序列化，以及“请求-响应”模式的信息交换。

**集群容错**：提供基于接口方法的透明远程调用，包括多协议支持，以及软负载均衡，失败容错，地址路由，动态配置等集群支持。

**自动发现：**基于注册中心目录服务，使服务消费方能动态的查找服务提供方，使地址透明，使服务提供方可以平滑增加或减少机器。

Dubbo作用

1. 透明化的远程调用，就像本地方法一样，调用远程方法，只要简单配置即可，注意问题：接口类的包名和类名在服务端和客户端必须完全一致。而且如果需要用到bean序列化，则要序列化的bean在服务端和客户端的包名和类名也完全一致。
2. 软负载均衡以及容错机制。
3. 服务自动注册和发现，不需要写死服务方地址，在注册中心注册该服务即可，客户端请求到注册中心，注册中心根据接口查找服务端ip地址等信息，所以能平滑的增减服务端机器。

Dubbo节点

Provider：暴露的服务方，即 “服务提供方”；

Consumer：调用远程方，即 “服务消费者”；

Registry: 服务注册与发现的中心目录服务，即 “服务注册中心”；

Monitor：统计服务的调用次数和调用时间的日志服务，即 “服务检测中心”；

Container：服务运行容器；

节点之间调用关系

1. 服务运行容器container，负责启动，加载，运行服务提供者provider。
2. 服务提供者provider启动时，向注册中心Registry注册自己提供的服务
3. 服务消费者counsumer在启动时，向注册中心Registry订阅自己所需服务。
4. 注册中心Registry返回服务提供者provider地址列表给消费者consumer，如果有变更，注册中心Registry将基于长连接推送变更消息给服务消费者consumer。
5. 服务消费者consumer从服务提供者provider地址列表中，基于软负载均衡算法，选一台服务提供者provider进行调用，如果调用失败，则另选一台。
6. 服务消费者consumer和提供者provider，在内存中累计调用次数和调用时间，定时每分钟发送一次统计数据到监测中心monitor。

Dubbo的连通性  
 1，注册中心Registry负责服务地址的注册和查找，相当于服务目录，服务提供者provider和消费者consumer只在启动时与注册中心Registry交互，不进行请求转发，压力较小.(不要忘了消息推送)

2，监测中心monitor负责统计个服务的调用次数，调用时间等，统计现在内存中汇总后每分钟一次发送到检测中心monitor服务器，并以报表展示。

3，服务提供者provider向注册中心Registry注册其服务，并汇报调用时间到监控中心，（调用时间不包括网络时间）

4，服务消费者consumer从注册中心获取服务提供者provider的地址列表，根据负载均衡算法直接进行调用服务提供者provider，同时汇报调用时间到监控中心monitor，（该事件包括网络传输时间）

1. 注册中心Registry，服务提供者provider，服务消费者从consumer三者之间均为长连接，但监控中心不是。
2. 注册中心Registry通过长连接感知服务提供者provider的存在，服务提供者provider宕机后，注册中心Registry将立即推送通知信息给服务消费者consumer。
3. 注册中心Registry和监测中心monitor全部宕机后也不会影响服务的消费者consumer和服务提供者provider，消费者本地缓存了服务提供者的地址列表，但是此时监控中心monitor无法统计调用信息。新增和减少的服务提供者provider也无法同步到服务消费者consumer。
4. 注册中心Registry和监测中心monitor都是可选的，即使没有服务提供者provider和服务消费者consumer之间也可以直接调用。

Dubbo健壮性

1. 监测中心monitor宕机后不会影响服务提供者provider和服务消费者consumer之间调用，但是会损失采样数据。
2. 数据库宕机之后，注册中心Registry仍能通过缓存提供服务列表查询，但不能注册新服务。有疑问：这个数据库是dubbo本身自带的数据库吗？
3. 注册中心Registry对等集群，任意一台宕机后，将自动切换到另一台。（注册中心单独部署集群，只是为了安全考虑，资源方面要求应该不是很高）。
4. 注册中心Registry宕机之后，服务消费者consumer仍能通过本地缓存的服务提供者provider地址列表，进行通讯
5. 服务提供者provider，存在多台的情况下，任意一台宕机都不会影响服务消费者consumer调用。
6. 服务提供者provider如果全部宕机，则服务消费者consumer无法再调用，并开始无限制的重连，等待服务提供者恢复。

Dubbo的伸缩性

1. 注册中心Registry为对等集群，可动态增加机器部署实例，所有客户端（是否包括服务提供者provider和服务消费者consumer）将自动发现新的注册中心。即注册中心Registry可以随时新增，但是注册中心应该有公共域名，否则客户端如何自动发现
2. 如果服务提供者无状态，可以动态新增机器部署实例，注册中心Registry将会推送新的服务提供者provider信息给服务消费者consumer。

配置项描述

<dubbo:service/>服务配置用于暴露一个服务，定义服务的元，注：一个服务可以在多个注册中心暴露，一个服务还可以用多个协议暴露。

<dubbo:reference/>引用配置，用于创建一个远程服务代理，一个引用可以指向多个注册中心registry

<dubbo:protocol/>协议配置，用于配置提供服务的协议信息，协议由服务提供方provider指定，消费方consumer被动接受。

<dubbo:application/>应用配置，用于配置当前应用信息，不管是服务提供者provider还是服务消费者consumer都需要配置，但是两者配置的名称不能相同，该名称可用于计算依赖关系

<dubbo:module/>模块配置，用于配置当前模块信息

<dubbo:registry/>注册中心regsitry的配置，

<dubbo:monitor/>监控中心monitro的配置项。

<dubbo:provider/>服务提供方provider的配置。当protocolconfig和serviceconfig某属性没有配置时采用此缺省值，可选。（尚未用到，如何使用方法待定）

<dubbo:consumer/>服务消费者consumer的配置，当referenceconfig某属性没有配置时，采用此缺省值（尚未用到，具体使用方法待确定）

<dubbo:method/>方法配置，用于serviceconfig和referenceconfig指定方法级的配置信息

<dubbo:argument>用于指定方法参数配置（暂时未用到，具体使用方法待定）

配置案例：



上图中以timeout为例，展示配置生效的顺序，其他的配置retries,loadbalance,actives等类似。

1， 顺序为：方法级优先，接口配置次之，全局配置最后

注：如果级别相同，则消费方consumer的配置优先，服务提供方provider次之。

1. 服务提供方provider的配置经过url经由注册中心registry传递给服务消费方consumer。
2. 建议：有服务提供方provider提供超时设置，服务提供方更清楚该方法的执行时间，而且当服务消费者consumer调用多个服务提供者provider时也就不用关心超时时间设置的问题了。
3. Referenceconfig的非服务标识配置，在consumerconfig，serviceconfig，providerconfig都可以缺省配置

注：dubbo的配置方法主要有三种，xml文件配置，通过dubbo.properties进行属性配置，通过注解进行配置，通过在程序中调用API进行配置。

**具体配置信息见**[**http://dubbo.io/User+Guide-zh.htm**](http://dubbo.io/User+Guide-zh.htm)**。**

**注：服务消费者consumer可以通过url直连服务提供者provider。**

**Dubbo的check默认为false**

名词解释：

nio 即 new io从java1.4版本引入的一个新的IO API，可以替代标准java IO API。

ORM 即 数据访问框架，对象关系映射。

MVC 即web框架

RPC 即分布式服务框架Remote Procedure Call Protocol，

SOA 面向服务体系结构

RMI ：即Remote Method Invocation，远程方法调用。