## 整合springboot、dubbo、zookeeper

### SOA

面向服务的架构（SOA）是一个组件模型，它将应用程序的不同功能单元（称为服务）进行拆分，并通过这些服务之间定义良好的接口和契约联系起来。接口是采用中立的方式进行定义的，它应该独立于实现服务的硬件平台、操作系统和编程语言。这使得构建在各种各样的系统中的服务可以以一种统一和通用的方式进行交互。

实现SOA架构时,常用服务.Dubbo 做为服务；WebService 做为服务； Dubbox 做为服务；服务方就是web项目,调用web项目的控制器(使用HttpClient可以调用其他项目的控制器)。

### Zookeeper

ZooKeeper是一个分布式的，开放源码的分布式应用程序协调服务，是Google的Chubby一个开源的实现，它是集群的管理者，监视着集群中各个节点的状态根据节点提交的反馈进行下一步合理操作。最终，将简单易用的接口和性能高效、功能稳定的系统提供给用户，提供的主要功能为文件系统与通知机制。与dubbo整合时，主要用到的是其通知机制。

### Dubbo

### Dubbo阿里巴巴公司开源的一个高性能优秀的服务框架，使得应用可通过高性能的 RPC (Remote Procedure Call———[远程过程调用](https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8/7854346" \t "https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8%E5%8D%8F%E8%AE%AE/_blank)，它是一种通过[网络](https://baike.baidu.com/item/%E7%BD%91%E7%BB%9C/143243" \t "https://baike.baidu.com/item/%E8%BF%9C%E7%A8%8B%E8%BF%87%E7%A8%8B%E8%B0%83%E7%94%A8%E5%8D%8F%E8%AE%AE/_blank)从远程计算机程序上请求服务，而不需要了解底层网络技术的协议)实现服务的输出和输入功能，以及SOA服务治理方案。

服务器

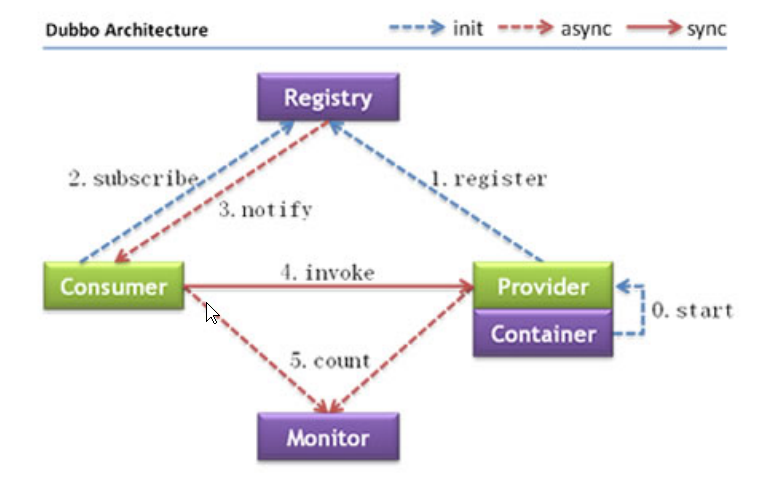
只有提供功能列表,没有功能具体实现

客户端

RPC简单说明

Dubbo是一个分布式、高性能、透明化的RPC服务框架,提供服务自动注册、自动发现等高效服务治理方案。

下图是官方提供的dubbo架构图



3.1 Provider :提供者,服务发布方.

3.2 Consumer:消费者, 调用服务方

3.3 Container:Dubbo容器.依赖于Spring容器(如果没有spring，则无法使用Dubbo).

3.4 Registry: 注册中心.当Container启动时把所有可以提供的服务列表上Registry中进行注册.

3.4.1 作用:告诉Consumer提供了什么服务和服务方在哪里.

3.5 Monitor:监听器

3.6 虚线都是异步访问,实线都是同步访问

3.7 蓝色虚线:在启动时完成的功能

3.8 红色虚线(实线)都是程序运行过程中执行的功能

3.9 所有的角色都是可以在单独的服务器上.所以必须遵守特定的协议.

### 4.运行原理:

4.0 启动容器,相当于在启动Dubbo的Provider

4.1 启动后会去注册中心进行注册.注册所有可以提供的服务列表

4.2 在Consumer启动后会去Registry中获取服务列表和Provider的地址.进行订阅.

4.3 当Provider有修改后,注册中心会把消息推送给Consummer

4.3.1 使用了观察者设计模式(又叫发布/订阅设计模式)

4.4 根据获取到的Provider地址,真实调用Provider中功能.

4.4.1 在Consumer方使用了代理设计模式.创建一个Provider方类的一个代理对象.通过代理对象获取Provider中真实功能,起到保护Provider真实功能的作用.

4.5 Consumer和Provider每隔1分钟向Monitor发送统计信息,统计信息包含,访问次数,频率等.

### 5.Dubbo支持的注册中心

1. Zookeeper

1.1 优点:支持网络集群

1.2 缺点:稳定性受限于Zookeeper

1. Redis

2.1 优点:性能高.

2.2 缺点:对服务器环境要求较高.

1. Multicast

3.1 优点:面中心化,不需要额外安装软件.

3.2 缺点:建议同机房(局域网)内使用

1. Simple

4.1 适用于测试环境.不支持集群.

### 6. Dubbo支持的协议

1. Dubbo

1.1 Dubbo官方推荐的协议.

1.2 本质:使用NIO和线程池进行处理.

1.3 缺点:大文件传输时可能出现文件传输失败问题.

1. RMI

2.1 JDK提供的协议,远程方法调用协议.

2.2 缺点:偶尔连接失败.

2.3 优点:JDK原生,不需要进行额外配置(导入jar)

1. Hession

3.1 优点:基于http协议,http请求支持.

3.2 缺点:需要额外导入jar,并在短连接时性能低

### 常见异常

1. dubbo 也有@Service但在不同的spring版本中也会存在这个注解名，所以我们一般不用这个注解而用xml配置。

如果可以使用@Service注解的话，我们在xml中可以省两个配置步骤，只需要在xml中开启dubbo的扫描包的路径，就可以自动配置需要发布的接口和对应的接口实现类的bean

消费者不存在上面的问题，可以使用自动注解，因为消费者不需要使用@service,他是用的是@Reference

2.com.alibaba的dubbo的2.5.3版本默认依赖spring的2.5.3版本。如果使用高版本的话需要maven在dubbo中

通过Exclusions排除这个spring的老版本(2.\*和3.\*+版本太大通过引入高版本的spring无法覆盖这个spring-2.5.6的jar包)，

3.开发时启动生产者，需要注意生产者服务器最好不要wify热点，可能会导致dubbo注册失败，但是不会报错

4.如果使用Main.main启动生产者是需要在把spring的配置文件放到META-INF/spring的路径下，如果是maven项目，则META-INF应该在resources目录下

5.启动dubbo时，可能出现下面两个异常

一个是 The matching wildcard is strict, but no declaration can be found for element 'dubbo:application'.

另一个是找不到命名空间的问题

引起这两个问题的原因，可能是dubbo配置文件xml的编码格式不正确或存在特殊字符(从官网复制的时候，存在异常)，另一个可能是没有引入dubbo的jar。

对于找不到明明空的问题，网上也有人说是xsd不正确，可以采用引入本地xsd的方式尝试，但我不是这个原因，我是由于在网页上复制的时候，出现了特殊字符。