一、Dubbo框架背景

1. 架构演变

1.1 单一应用架构

当网站流量很小时，只需一个应用，将所有功能都部署在一起，以减少部署节点和成本。

此时，用于简化增删改查工作量的 数据访问框架(ORM) 是关键。

1.2 垂直应用架构

当访问量逐渐增大，单一应用增加机器带来的加速度越来越小，将应用拆成互不相干的几个应用，以提升效率。此时，用于加速前端页面开发的 Web框架(MVC) 是关键。

1.3 分布式服务架构（RPC）

当垂直应用越来越多，应用之间交互不可避免，将核心业务抽取出来，作为独立的服务，逐渐形成稳定的服务中心，使前端应用能更快速的响应多变的市场需求。

此时，用于提高业务复用及整合的 分布式服务框架(RPC) 是关键。

1.4 流动计算架构

当服务越来越多，容量的评估，小服务资源的浪费等问题逐渐显现，此时需增加一个调度中心基于访问压力实时管理集群容量，提高集群利用率。

此时，用于提高机器利用率的 资源调度和治理中心(SOA) 是关键。

2. PRC框架（Remote Procedure Call Protocol 远程调用）

(1) RPC是一种通过网络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。RPC协议假定某些传输协议的存在，如TCP或UDP，为通信程序之间携带信息数据。在OSI网络通信模型中，RPC跨越了传输层和应用层。RPC使得开发包括网络分布式多程序在内的应用程序更加容易。

(2) R**PC采用客户机/服务器模式**。请求程序就是一个客户机，而服务提供程序就是一个服务器。首先，客户机调用进程发送一个有进程参数的调用信息到服务进程，然后等待应答信息。在服务器端，进程保持睡眠状态直到调用信息到达为止。当一个调用信息到达，服务器获得进程参数，计算结果，发送答复信息，然后等待下一个调用信息，最后，客户端调用进程接收答复信息，获得进程结果，然后调用执行继续进行。

3. RPC主要解决的问题

(1) 通讯问题 : 主要是通过在客户端和服务器之间建立TCP连接，远程过程调用的所有交换的数据都在这个连接里传输。连接可以是按需连接，调用结束后就断掉，也可以是长连接，多个远程过程调用共享同一个连接。

(2) 寻址问题 ： A服务器上的应用怎么告诉底层的RPC框架，如何连接到B服务器（如主机或IP地址）以及特定的端口，方法的名称名称是什么，这样才能完成调用。比如基于Web服务协议栈的RPC，就要提供一个endpoint URI，或者是从UDDI服务上查找。如果是RMI调用的话，还需要一个RMI Registry来注册服务的地址。

(3) 序列化 与 反序列化 ： 当A服务器上的应用发起远程过程调用时，方法的参数需要通过底层的网络协议如TCP传递到B服务器，由于网络协议是基于二进制的，内存中的参数的值要序列化成二进制的形式，也就是序列化（Serialize）或编组（marshal），通过寻址和传输将序列化的二进制发送给B服务器。

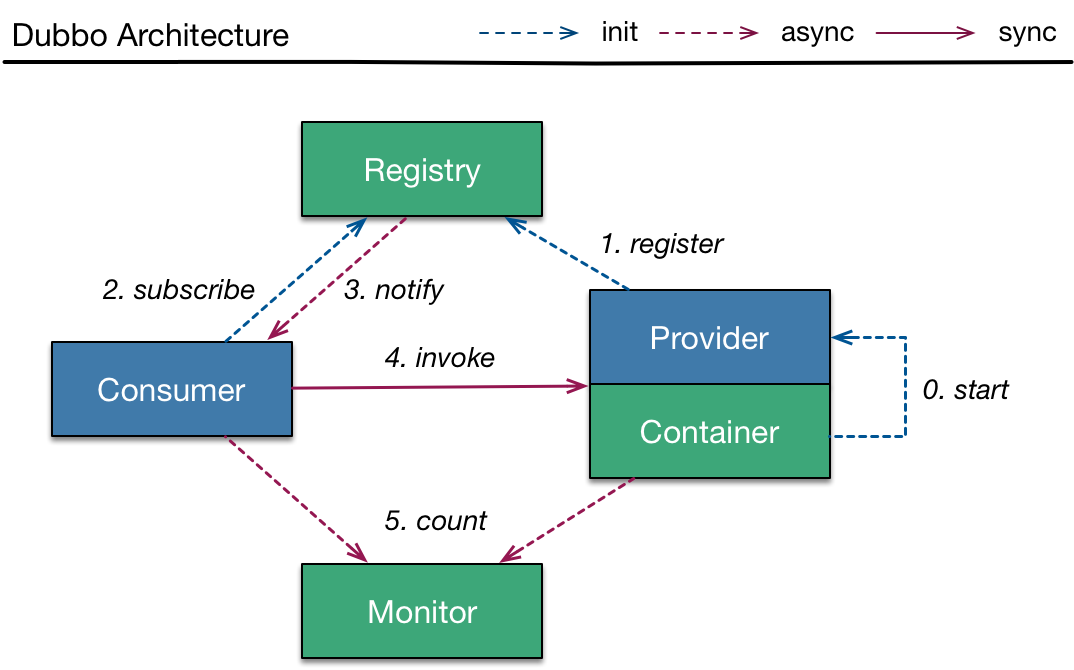
同理，B服务器接收参数要将参数反序列化。B服务器应用调用自己的方法处理后返回的结果也要序列化给A服务器，A服务器接收也要经过反序列化的过程。

二、Dubbo框架概念

1. 定义

分布式服务框架，高性能和透明化的RPC远程服务调用方案，SOA服务治理方案。

2. 工作原理



Provider: 暴露服务的服务提供方。

Consumer: 调用远程服务的服务消费方。

Registry: 服务注册与发现的注册中心。

Monitor: 统计服务的调用次数和调用时间的监控中心。

2.1 调用流程

(1) 服务容器负责启动，加载，运行服务提供者。

(2) 服务提供者在启动时，向注册中心注册自己提供的服务。

(3) 服务消费者在启动时，向注册中心订阅自己所需的服务。

(4) 注册中心返回服务提供者地址列表给消费者，如果有变更，注册中心将基于长连接推送变更数据给消费者。

(5) 服务消费者，从提供者地址列表中，基于软负载均衡算法，选一台提供者进行调用，如果调用失败，再选另一台调用。

(6) 服务消费者和提供者，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控中心

3. Dubbo注册中心

3.1 作用：

(1) 于服务提供方，它需要发布服务，而且由于应用系统的复杂性，服务的数量、类型也不断膨胀；

(2) 对于服务消费方，它最关心如何获取到它所需要的服务，而面对复杂的应用系统，需要管理大量的服务调用。

(3) 对于服务提供方和服务消费方来说，他们还有可能兼具这两种角色，即既需要提供服务，有需要消费服务。

(4) 通过将服务统一管理起来，可以有效地优化内部应用对服务发布/使用的流程和管理。服务注册中心可以通过特定协议来完成服务对外的统一。

3.2 注册中心的类型

(1) Multicast注册中心

(2) Zookeeper注册中心

(3) Redis注册中心

(4) Simple注册中心

4. Dubbo优点

(1) 透明化的远程方法调用 - 像调用本地方法一样调用远程方法；只需简单配置，没有任何API侵入。

(2) 软负载均衡及容错机制，可在内网替代nginx lvs等硬件负载均衡器。

(3) 服务注册中心自动注册 & 配置管理

-不需要写死服务提供者地址，注册中心基于接口名自动查询提供者ip。

使用类似zookeeper等分布式协调服务作为服务注册中心，可以将绝大部分项目配置移入zookeeper集群。

(4) 服务接口监控与治理

-Dubbo-admin与Dubbo-monitor提供了完善的服务接口管理与监控功能，针对不同应用的不同接口，可以进行 多版本，多协议，多注册中心管理。

三、RPC框架和MVC框架的区别

1. MVC框架是直接用程序调用，调用方和被调用方在一个war包内，可扩展性和并发性较差。

2. RPC框架的主要特征是远程调用，服务端将服务发布到注册中心，客户端从注册中心调用服务，Dubbo框架是解决服务端到注册中心，客户端到注册中心的通信问题，以及服务从发布到调用完整过程的寻址问题。