**背景：**

传统的网站架构为单一应用架构，也就是说把所有的功能都放在一个项目工程里，部署到一台服务器上，比如我们建立一个互联网金融p2p网站，我们创建一个工程叫p2p，这个工程里面包含所有的功能：注册、登录、投资、充值、提现、合同、红包、消息等等。

当访问量越来越大，单一的服务器已不能满足需要，我们需要通过不断添加服务器的方式来应对越来越大的访问量，但是通过不断添加服务器的方式带来的速度提升也越来越小，此时我们就需要拆分我们的应用，拆分成几个独立的不相干的应用，以提升服务的能力。

随着用户的增长和业务的发展，我们拆分的应用会越来越多，应用之间的交互和数据传输已不可避免，此时我们就需要将一些核心的功能抽取出来，形成一个个公共的服务，不同系统直接调用这些公共的服务，提高业务复用。

但是当我们的服务越来越多，资源的不合理分配也会变得比较严重，比如一些访问量很小的服务，占用了大量的IT资源，而另外一些应用的资源严重不足，服务负载一直高居不下，我们需要能基于访问流量的情况实时监控和管理各个服务，提高资源的利用率。

**系统进化理论：（架构演进：单体架构、SOA、微服务）**

集中式系统：就是把所有的程序、功能、模块都集中到一个项目中，部署在一台服务器上，从而对外提供服务。（缺点：复杂性逐渐变高、技术债务逐渐上升、部署速度逐渐变慢、阻碍技术创新、无法按需伸缩）

分布式系统：分布（是指在一定的范围内分散开，其反义词是：集中）

分布式系统就是把所有的程序、功能拆分成不同的子系统，部署在多台不同的服务器上，这些子系统相互协作共同对外提供服务，对用户而言他并不知道后台是多个子系统和多台服务器在提供服务，在使用上和集中式系统一样。

所以集中式系统和分布式系统是相反的两个概念，他们的区别体现在“合”与“分”

**集群和分布式：**

集群：就是将相同程序、功能，部署在两台或多台服务器上，这些服务器对外提供的功能是完全一样的，集群是通过不断横向扩展增加服务器的方式，以提高服务的能力。

分布式：就是将两个或多个程序、功能分别运行在两台或多台主机服务器上，这些服务器对外提供的功能并不一样，它们通过相互协作最终完成某一个服务或功能。

所以简单说，如果两台服务器部署的程序完全一样是集群，不一样就是分布式，分布式中的每一个节点，都可以做集群，而集群并不一定是分布式的。

**分布式架构：**

分布式系统设计的本质就是：如何合理的将一个系统拆分为多个子系统部署到多台不同的服务器上，因此需要解决如下几个问题：

1. 如何合理的拆分系统
2. 拆分后的各个子系统之间如何通信
3. 通信过程如何防止信息拦截和篡改等安全问题
4. 如何适应不断增长的业务需求，使其具有良好的可扩展性
5. 如何保证子系统的可靠性和数据的最终一致性

**Dubbo：**

Dubbo是一个分布式服务治理框架，提供高性能和透明化的RPC远程服务调用及服务治理。Dubbo是阿里巴巴开源产品，国内大量企业都在用dubbo框架。

用途：RPC远程调用和服务治理（透明化的远程方法调用，就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）

PRC远程调用功能的框架：dubbo、springcloud、hessian、thrift、motan、gRPC...

服务治理功能的框架：bubbo、spring cloud、motan...

项目中使用dubbo是采用spring配置的方式集成dubbo，这也是官方推荐的方式，这种方式不会有任何的代码侵入。

**RPC:Remote Procedure Call远程过程调用**

比如有两台服务器A和B，一个应用部署在A服务器上，另一个应用部署在B服务器上，如果A应用想要调用B应用提供的方法，由于他们不在一台机器上，也就是他们不在一个JVM内存空间，无法直接调用，需要通过网络进行调用，那么这个过程就叫RPC。

在同一个JVM下，一个方法可以直接调用另一个方法，如controller调service，service调dao

不同的JVM下，一个项目的方法通过RPC调用另一个项目的方法，也叫跨JVM调用

RPC调用过程如何实现：

基于tcp协议的rpc，运用到的技术有（序列化与反序列化、socket、反射）

基于http协议的RPC,xml、json

Dubbo已经帮我们实现了这一切，但dubbo不仅仅是rpc

**Dubbo的整体架构：**

五大核心部件：

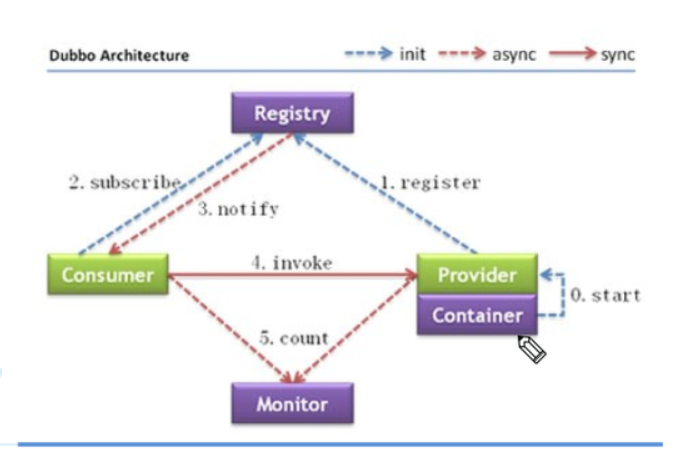
Provider：提供服务的服务提供方（生产者）

Consumer：调用远程服务的服务消费方（消费者）

Registry：服务注册与发现的注册中心（注册中心）

Monitor：统计服务调用次数和调用时间的监控中心（监控中心）

Container：服务运行容器（运行容器，启动运行Provider、Consumer，可理解为 main方法或tomcat容器）



**Dubbo注册中心：（zookeeper、redis与dubbo没必然关系，只是dubbo使用它们作为注册中心）**

1. multicast注册中心（不需要专门下载注册中心软件，在xml中配置即可，常用语开发或测试时使用该注册中心）
2. Zookeeper注册中心：是一个高性能，分布式的，开源分布式应用协调服务软件，dubbo官方推荐使用zookeeper注册中心，是apache下的一个软件类似于tomcat
3. Redis注册中心：是一个高效的基于内存亦可持久化的键值对存储服务器，是非常流行的NoSQL数据库

**Dubbo分布式服务的开发模式：**

传统模式MVC：分层思想，分为三层（业务控制层controller层，业务逻辑层处理service层，数据库操作dao层），将所有功能写在一个项目中。

分布式SOA服务化（面向服务）：分割思想，将一个系统分割成一个个服务，如用户中心，支付中心，订单中心...

Dubbo常用协议(8种）：

1. dubbo://

Dubbo协议采用单一长连接和NIO异步通信，适合于小数据量大并发的服务调用，以及服务消费者机器数远大于服务提供者机器数的情况

Dubbo协议底层默认使用netty，性能非常优秀，官方推荐使用此协议

Dubbo协议不适合传送大数据量的服务，比如传文件、传视屏等，除非请求量低

<dubbo:protocol name=”dubbo” port=”20880” />,dubbo协议默认端口为20880

1. hessian://

Hessian是Caucho开源的一个RPC框架，其通讯效率高于WebService和java自带的序列化

1. rmi://
2. http://
3. Webservice://
4. Thrift://

是facebook捐给apache的一个rpc框架

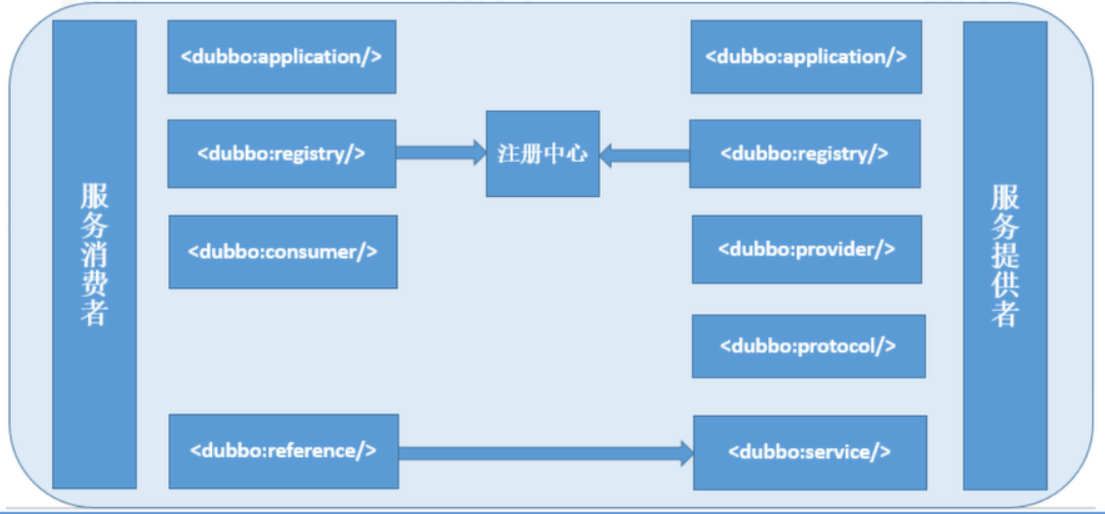
1. memcached://

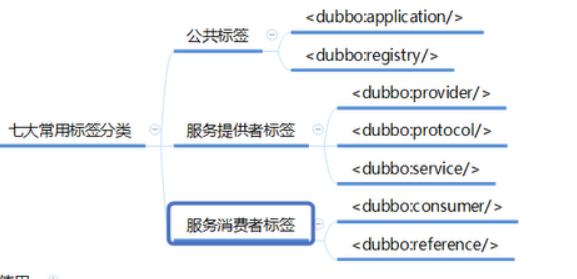
Memcache是一个高效的KV缓存服务器

1. redis://

Redis是一个高效的KV存储服务器

**Dubbo常用标签：**





**Dubbo管控台的部署与配置**（dubbo提供的一个web程序，用于监控服务的调用信息）

**Zookeeper集群：**

Zk集群特点：集群中只要有超过半数的机器是正常工作的，那么整个集群对外就是可用的，也就是说如果有2个zookeeper，那么只要有1个故障了，zookeeper就不能用了，因为1没有过半，所以两个zookeeper不是高可用的，因为不能容忍任何一台发生故障，一般集群配奇数台zookeeper服务器

**服务化建议：**

