Dubbo和RPC框架

* 1. 从图中你能看到，Dubbo 的架构主要包含四个角色，其中 Consumer 是服务消费者，Provider 是服务提供者，Registry 是注册中心，Monitor 是监控系统。  
     具体的交互流程是 Consumer 一端通过注册中心获取到 Provider 节点后，通过 Dubbo 的客户端 SDK 与 Provider 建立连接，并发起调用。Provider 一端通过 Dubbo 的服务端 SDK 接收到 Consumer 的请求，处理后再把结果返回给 Consumer。  
     **1.2 RPC架构**  
     一个完整的RPC架构里面包含了四个核心的组件，分别是Client，Client Stub，Server以及Server Stub，这个Stub可以理解为存根。
  2. 客户端(Client)，服务的调用方。
  3. 客户端存根(Client Stub)，存放服务端的地址消息，再将客户端的请求参数打包成网络消息，然后通过网络远程发送给服务方。
  4. 服务端(Server)，真正的服务提供者。
  5. 服务端存根(Server Stub)，接收客户端发送过来的消息，将消息解包，并调用本地的方法。

**Netty**

**最流行的 NIO 框架，由 JBOSS 提供的，整合了FTP,SMTP,HTTP协议**

1. API 简单
2. 成熟稳定
3. 社区活跃·
4. 经过大规模验证（互联网、大数据、网络游戏、电信通信）  
   Elasticsearch、Hadoop 子项目 avro项目、阿里开源框架 Dubbo、使用 Netty

### 五种 I/O 模型

五种 I/O 模型：  
阻塞 IO、非阻塞 IO、多路复用 IO、信号驱动 IO、异步 IO，前 4 种是同步 IO，在内核数据 copy 到用户空间时是阻塞的

**IO 多路复用**

* 1. I/O 是指网络 I /O ，多路指多个 TCP 连接，复用指一个或几个线程。  
     简单来说：就是使用一个或者几个线程处理多个 TCP 连接，最大优势是减少系统开销，不必创建过多的线程进程，也不必维护这些线程进程

**TCP和UDP的区别？**

* 1. TCP是面向连接的（在客户端和服务器之间传输数据之前要先建立连接），UDP是无连接的（发送数据之前不需要先建立连接）
  2. TCP提供可靠的服务（通过TCP传输的数据。无差错，不丢失，不重复，且按序到达）；UDP提供面向事务的简单的不可靠的传输。
  3. UDP具有较好的实时性，工作效率比TCP高，适用于对高速传输和实时性比较高的通讯或广播通信。随着网速的提高，UDP使用越来越多。
  4. 没一条TCP连接只能是点到点的，UDP支持一对一，一对多和多对多的交互通信。
  5. TCP对系统资源要去比较多，UDP对系统资源要求比较少
  6. UDP程序结构更加简单
  7. TCP是流模式，UDP是数据报模式
  8. **TCP协议如何保证可靠传输？**  
     TCP通过序列号、检验和、确认应答信号、重发控制、连接管理、窗口控制、流量控制、拥塞控制实现可靠性。  
     参考：<https://www.jianshu.com/p/6aac4b2a9fd7>

## TCP三次握手过程

第一次握手：主机A通过向主机B 发送一个含有同步序列号的标志位的数据段给主机B，向主机B 请求建立连接，通过这个数据段， 主机A告诉主机B 两件事：我想要和你通信；你可以用哪个序列号作为起始数据段来回应我。

第二次握手：主机B 收到主机A的请求后，用一个带有确认应答（ACK）和同步序列号（SYN）标志位的数据段响应主机A，也告诉主机A两件事：我已经收到你的请求了，你可以传输数据了；你要用那个序列号作为起始数据段来回应我

第三次握手：主机A收到这个数据段后，再发送一个确认应答，确认已收到主机B 的数据段："我已收到回复，我现在要开始传输实际数据了，这样3次握手就完成了，主机A和主机B 就可以传输数据了。

## 3次握手的特点

没有应用层的数据 ,SYN这个标志位只有在TCP建立连接时才会被置1 ,握手完成后SYN标志位被置0

## TCP建立连接要进行3次握手，而断开连接要进行4次

第一次： 当主机A完成数据传输后,将控制位FIN置1，提出停止TCP连接的请求 ；

第二次： 主机B收到FIN后对其作出响应，确认这一方向上的TCP连接将关闭,将ACK置1；

第三次： 由B 端再提出反方向的关闭请求,将FIN置1 ；

第四次： 主机A对主机B的请求进行确认，将ACK置1，双方向的关闭结束.。

由TCP的三次握手和四次断开可以看出，TCP使用面向连接的通信方式， 大大提高了数据通信的可靠性，使发送数据端和接收端在数据正式传输前就有了交互， 为数据正式传输打下了可靠的基础。

## 名词解释

1、ACK 是TCP报头的控制位之一，对数据进行确认。确认由目的端发出， 用它来告诉发送端这个序列号之前的数据段都收到了。 比如确认号为X，则表示前X-1个数据段都收到了，只有当ACK=1时,确认号才有效，当ACK=0时，确认号无效，这时会要求重传数据，保证数据的完整性。

2、SYN 同步序列号，TCP建立连接时将这个位置1。

3、FIN 发送端完成发送任务位，当TCP完成数据传输需要断开时,，提出断开连接的一方将这位置1。