

馨缘·201708

服务之璋RPC

内容概要

▶ 服务的演进

目前市场上服务器开发由小到大的演进过程，构建高可用服务的常用设计

▶ 走进RPC

1. TCP/IP

2. HTTP

3. 常用的RPC thrift,lavar,dubbo系,HSF(traceId)等

▶ Thrift/dubbo

内容概要

▶ 服务的演进

目前市场上服务器开发由小到大的演进过程，构建高可用服务的常用设计

▶ 走进RPC

1. TCP/IP

2. HTTP

3. 常用的RPC thrift,lavar,dubbo系,HSF(traceId)等

▶ Thrift/dubbo

内容概要

- ▶ 1. Thrift架构
 - 2. Thrift服务器类型
 - 3. Thrift传输协议对比
 - 4. dubbo架构
 - 5. dubbo与Thrift比较
- ▶ Hello Thrift
 - 1. 环境搭建
 - 2. IDL语言简介
 - 3. 构建客户端和服务端，实现通信

内容概要

- ▶ Hello Dubbo
 - 1. 环境搭建
 - 2. 构建生产者，消费者，实现通信
- ▶ Spring与Thrift整合
- ▶ Protobuf介绍
- ▶ Protobuf与Thrift对比
- ▶ 总结

▶ 小型服务

特点：

1. 工程简单 往往只有一个项目 甚至前端后台都在一个工程里面，部署在单台服务器上。
2. 用户量很小，不存在并发和数据存储压力
3. 开发人员较少，代码维护成本较低

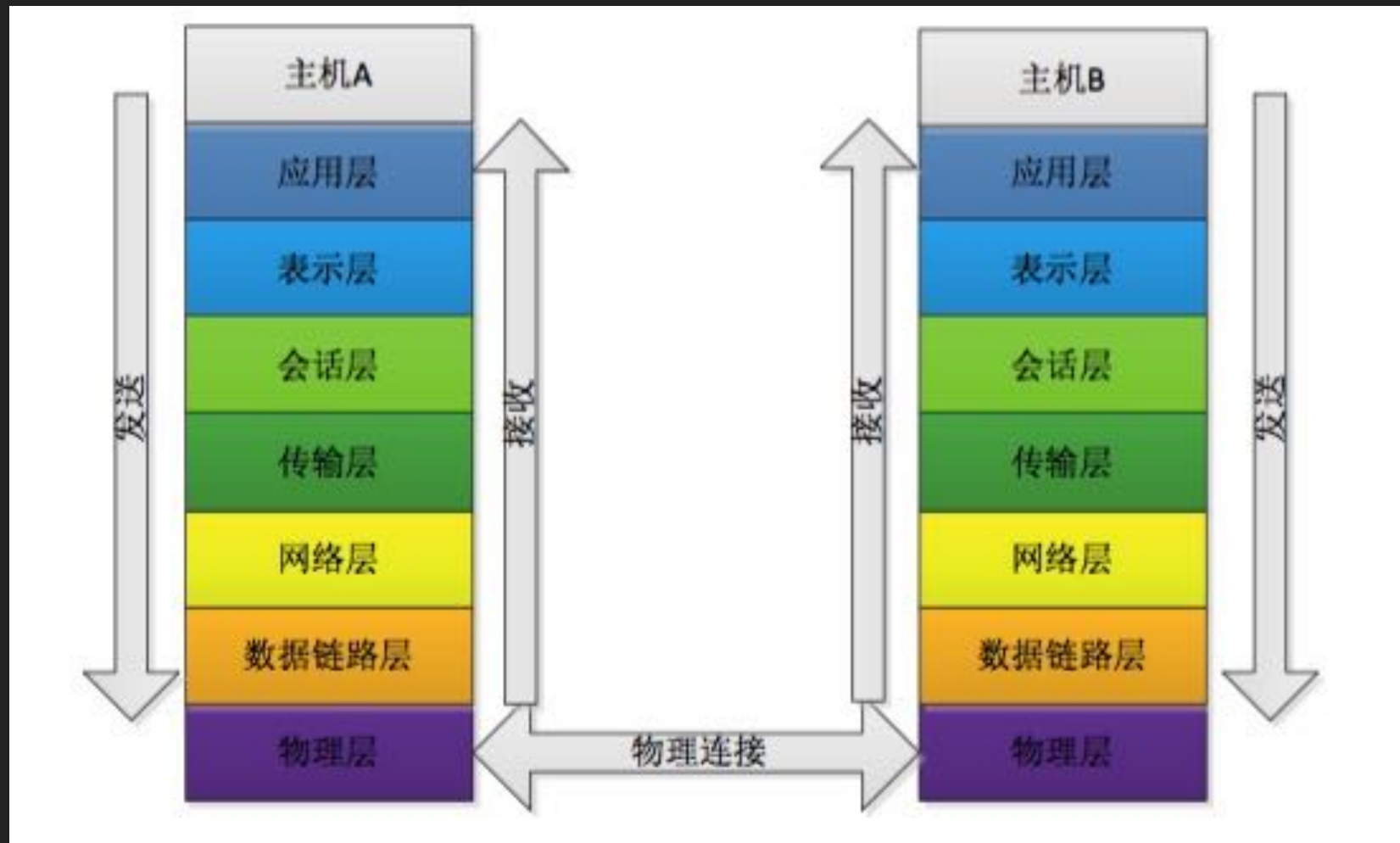
▶ 中型服务

1. 有了一定的用户，服务器开发人员增多，代码维护成本开始增加
2. 服务可用性要求增高
3. 做水平切分，将服务分割成多个层级

▶ 大型项目

1. 用户量较大，并发压力较大，服务复杂
1. 项目垂直切分，将一个单独的项目拆分为若干个模块。每个模块提供自己的个性化服务
2. 服务提供者将服务发布到配置中心，服务消费者订阅服务。

▶ OSI(Open System Interconnection mode)



▶ 七层模型

各层的具体描述如下：

第七层：应用层 定义了用于在网络中进行通信和数据传输的接口 - 用户程式；提供标准服务，比如虚拟终端、文件以及任务的传输 和处理；

第六层：表示层 掩盖不同系统间的数据格式的不同性；指定独立结构的数据传输格式；数据的编码和解码；加密和解密；压缩和 解压缩

第五层：会话层 管理用户会话和对话；控制用户间逻辑连接的建立和挂断；报告上一层发生的错误

第四层：传输层 管理网络中端到端的信息传送；通过错误纠正和流控制机制提供可靠且有序的数据包传送；提供面向无连接的数据包的传送；

第三层：网络层 定义网络设备间如何传输数据；根据唯一的网络设备地址路由数据包；提供流和拥塞控制以防止网络资源的损耗

第二层：数据链路层 定义操作通信连接的程序；封装数据包为数据帧；监测和纠正数据包传输错误

第一层：物理层 定义通过网络设备发送数据的物理方式；作为网络媒介和设备间的接口；定义光学、电气以及机械特性。

▶ HTTP

处于网络通信协议顶层的交互协议。HTTP定义了自己的协议规则，比如请求头，请求体，请求方法等

HTTP定义了多种状态码 如常见的200，404，500等

HTTP安全可靠，基于TCP

HTTP开发简单，普遍支持

▶ HTTP与RPC

HTTP有冗余数据传输

HTTP安全问题

HTTP不支持长连接

▶ 参考<http://friping.iteye.com/blog/531061>

▶ Thrift,lavar,dubbo,tair

Thrift fb开发的一套跨语言的RPC框架，支持多种序列化方式并且提供多种服务协议

Lavar:百度内部RPC框架，在Thrift基础上实现HA

Dubbo系:阿里开源框架，主要针对JAVA，有完善的文档，使用Spring配置风格，有友好的治理界面

tair:腾讯最近开源

▶ 参考<http://friping.iteye.com/blog/531061>

- ▶ IDL接口定义语言
- ▶ 跨语言
- ▶ 传输协议比较
- ▶ 服务类型比较
- ▶ spring整合
- ▶ Spring Thrift HA(zk)

THANK

YOU