

个人资料



-mindwind-



访问： 788600次  
积分： 8208  
等级： **BLOG > 5**  
排名： 第1957名  
  
原创： 147篇 转载： 0篇  
译文： 8篇 评论： 529条

文章分类

踏莎行·术 (64)  
破阵子·道 (25)  
青玉案·编 (65)  
后端分布式系列 (16)

文章存档

2017年03月 (1)  
2017年02月 (5)  
2017年01月 (4)  
2016年12月 (4)  
2016年11月 (5)

展开

阅读排行

深入浅出 RPC - 深入浅出 (132375)  
NIO系列6：流行 NIO Framework (37342)  
面向服务与微服务架构 (30594)  
深入浅出 RPC - 浅出篇 (28526)  
程序员，别了校园入了江湖 (22802)  
Redis 事务 (18169)

## 深入浅出 RPC - 浅出篇

标签： **RPC** 分布式计算

2014-09-22 11:06 28562人阅读 评论(11) 收藏 举报

分类：

踏莎行·术 (63) ▾

版权声明：本文为博主原创文章，未经博主允许不得转载。

目录(?)



- 1. **RPC 是什么**
- 2. **RPC 起源**
- 3. **RPC 结构**
- 4. **RPC 实现**
- 5. 总结

本文来自本人独立博客，为获得更佳阅读体验，请点击 [这里](#)

近几年的项目中，服务化和**微服务**化渐渐成为中大型分布式系统**架构**的主流方式，而 **RPC** 在其中扮演着关键的作用。在平时的日常开发中我们都在隐式或显式的使用 **RPC**，一些刚入行的程序员会感觉 **RPC** 比较神秘，而一些有多年使用 **RPC** 经验的程序员虽然使用经验丰富，但有些对其原理也不甚了了。缺乏对原理层面的理解，往往也会造成开发中的一些误用。

本文分上下两篇《浅出篇》和《深入篇》，其目标就是想尝试深入浅出的分析下 **RPC** 本质，我总是这么认为理解了本质才能更好的应用。

### RPC 是什么？

**RPC** 的全称是 **Remote Procedure Call** 是一种进程间通信方式。它允许程序调用另一个地址空间（通常是共享网络的另一台机器上）的过程或函数，而不用程序员显式编码这个远程调用的细节。即程序员无论是调用本地的还是远程的，本质上编写的调用代码基本相同。

### RPC 起源

**RPC** 这个概念术语在上世纪 80 年代由 **Bruce Jay Nelson** 提出。这里我们追溯下当初开发 **RPC** 的原动机是什么？在 **Nelson** 的论文 "**Implementing Remote Procedure Calls**" 中他提到了几点：

- 1. 简单：**RPC** 概念的语义十分清晰和简单，这样建立分布式计算就更容易。
- 2. 高效：过程调用看起来十分简单而且高效。
- 3. 通用：在单机计算中过程往往是不同算法部分间最重要的通信机制。

通俗一点说，就是一般程序员对于本地的过程调用很熟悉，那么我们把 **RPC** 作成和本地调用完全类似，那么就更容易被接受，使用起来毫无障碍。**Nelson** 的论文发表于 30 年前，其观点今天看来确实高瞻远瞩，今天我们使用的 **RPC** 框架基本就是按这个目标来实现的。

### RPC 结构

**Nelson** 的论文中指出实现 **RPC** 的程序包括 5 个部分：

- 1. User
- 2. User-stub
- 3. **RPCRuntime**

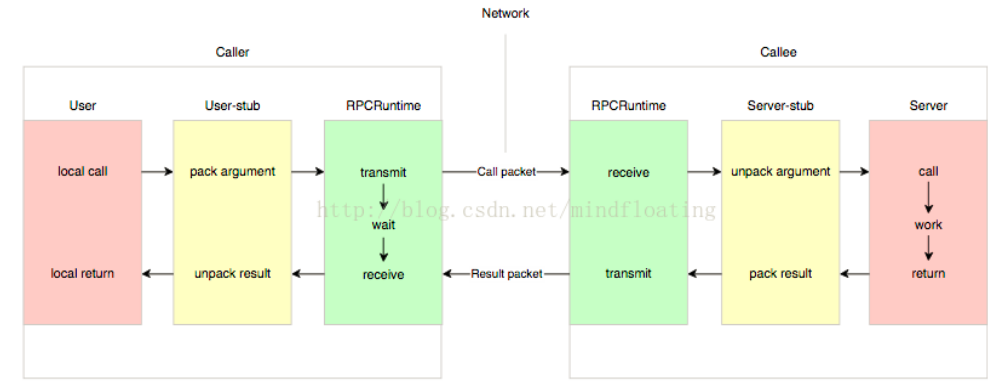
- 程序员，你为什么值这么
- (14823)
- RPC 的概念模型与实现
- (12902)
- IM设计思考：XMPP多用
- (12862)
- tomcat 性能之谜
- (12543)

- 评论排行
- 程序员，别了校园入了江
- (49)
- 程序员，你为什么值这么
- (24)
- 2016 工作、生活与得失
- (22)
- 深入浅出 RPC - 深入篇
- (18)
- 技术宅找女朋友的技术分
- (18)
- 程序员的沟通之痛
- (17)
- 京东咚咚架构演进
- (17)
- 程序员的口头禅：技术上
- (15)
- Java 征途：行者的地图
- (14)
- 成长路上，刀剑如梦
- (14)

- 最新评论
- 技术晋升的评定与博弈
- jiqiao1892: 有人的地方就是江湖，没有绝对公平
- 论年度计划的可行性
- 且听风雨999: @mindfloating: 首席你好，哈哈。我通过一个朋友问到你了。我在一家创业公司，做算法
- RPC 使用中的一些注意点
- 二师兄1986: 拜读，我收入了我的知识图谱
- 论年度计划的可行性
- mindwind: 你在哪家公司，做什么的呢？
- 技术晋升的评定与博弈
- mindwind: 对，处在两端的人很容易分辨，但中间的那部分就有看看运气了
- 技术晋升的评定与博弈
- I-M: 要比别人优秀1.3~1.5而不是一点点，这样自己的成绩就不会有争议了吧
- 程序员的沟通之痛
- 冯尧: 站的角度不同，想法也不一样。
- 技术晋升的评定与博弈
- 韩梦甜: 博主，很不错的分享！
- 技术晋升的评定与博弈
- mimixi666: @gzy11: 搞it这一行的话，这种情况比较少吧
- 技术晋升的评定与博弈
- 蜗牛水里爬: 说的在理，实际中，还是暗箱操作的居多。不提名，你连机会都没有。看似公平，其实没有公平。呵呵，其实加薪...

4. Server-stub
5. Server

这 5 个部分的关系如下图所示



这里 user 就是 client 端，当 user 想发起一个远程调用时，它实际是通过本地调用 user-stub。user-stub 负责将调用的接口、方法和参数通过约定的协议规范进行编码并通过本地的 RPCRuntime 实例传输到远端的实例。远端 RPCRuntime 实例收到请求后交给 server-stub 进行解码后发起本地端调用，调用结果再返回给 user 端。

RPC 实现

Nelson 论文中给出的这个实现结构也成为后来大家参考的标准范本。大约 10 年前，我最早接触分布式计算时使用的 CORBAR 实现结构基本与此类似。CORBAR 为了解决异构平台的 RPC，使用了 IDL（Interface Definition Language）来定义远程接口，并将其映射到特定的平台语言中。后来大部分的跨语言平台 RPC 基本都采用了此类方式，比如我们熟悉的 Web Service（SOAP），近年开源的 Thrift 等。他们大部分都通过 IDL 定义，并提供工具来映射生成不同语言平台的 user-stub 和 server-stub，并通过框架库来提供 RPCRuntime 的支持。不过貌似每个不同的 RPC 框架都定义了各自不同的 IDL 格式，导致程序员的学习成本进一步上升（苦逼啊），Web Service 尝试建立业界标准，无赖标准规范复杂而效率偏低，否则 Thrift 等更高效的 RPC 框架就没有必要出现了。

IDL 是为了跨平台语言实现 RPC 不得已的选择，要解决更广泛的问题自然导致了更复杂的方案。而对于同一平台内的 RPC 而言显然没必要搞个中间语言出来，例如 Java 原生的 RMI，这样对于 java 程序员而言显得更直接简单，降低使用的学习成本。目前市面上提供的 RPC 框架已经可算是五花八门，百家争鸣了。需要根据实际使用场景谨慎选型，需要考虑的选型因素我觉得至少包括下面几点：

1. 性能指标
2. 是否需要跨语言平台
3. 内网开放还是公网开放
4. 开源 RPC 框架本身的质量、社区活跃度

总结

《浅出篇》大概就到这里结束了，《深入篇》会具体深入讲解一个 RPC 框架需要实现哪里基本功能，达到什么目标，并以在 java 平台上去具体实现一个 RPC 框架为例，分析其需要考虑的实现因素。

顶

33

踩

2

- 上一篇
- 谈谈开源技术选型
- 下一篇
- 深入浅出 RPC - 深入篇

我的同类文章

踏莎行·术（63）			
• RPC 使用中的一些注意点	2016-12-25	阅读 555	• HTTPS 互联网世界的安全... 2016-11-20 阅读 1289
• HA 高可用软件系统保养指南	2016-06-26	阅读 2437	• 适合程序员的画图技法 2016-06-05 阅读 6041

• <a href="#">RPC 的概念模型与实现解析</a>	2016-05-22	• <a href="#">LB 负载均衡的层次结构</a>	2016-03-31	阅读 4609
• <a href="#">Raft 为什么是更易理解的...</a>	2016-03-01	• <a href="#">Redis 集群的合纵与连横</a>	2016-01-04	阅读 8265
• <a href="#">MongoDB 初见指南</a>	2016-01-18	• <a href="#">最近碰到的一些 SSL 问题...</a>	2016-02-16	阅读 2560

参考知识库



微服务知识库  
3424 关注 | 179 收录



Java SE知识库  
23490 关注 | 468 收录



Java EE知识库  
15736 关注 | 1265 收录



Java 知识库  
23624 关注 | 1449 收录



算法与数据结构知识库  
13905 关注 | 2320 收录



大型网站架构知识库  
7878 关注 | 708 收录

猜你在找

微信公众平台深度开发Java版 v2.0（第一季）完整版	深入浅出 RPC - 深入篇
微信公众平台深度开发 (Java版)	深入浅出 RPC - 深入篇
微信公众平台企业号开发Java版_2接收消息与响应消息	深入浅出 RPC - 深入篇
java语言从入门到精通2016+项目实训	深入浅出 RPC - 深入篇
Java Swing、JDBC开发桌面级应用	深入浅出 RPC - 深入篇

查看评论

8楼 [qq\\_37100744](#) 2016-12-21 14:52发表



博主的文章让我了解了RPC分布式的基本概念，也算是入门了，有很大的帮助，非常感谢

7楼 [白石江边](#) 2016-05-29 17:49发表



学习了

6楼 [air0955](#) 2016-05-03 20:36发表



之前写的留言被CSDN吞了，ORz。简单说吧，感谢楼主，很易懂的一片文章！

5楼 [air0955](#) 2016-05-03 20:32发表



博主的文章真的做到了深入浅出，特意登陆赞一个！看了之后，终于能从总体去了解认识RPC了，细节还需要自己去挖掘。  
赞美博主~~~~~

4楼 [cgicgicgl](#) 2016-01-16 12:29发表



这篇叫 浅入篇 更好

3楼 [hua504358405](#) 2015-06-04 14:22发表



你好，请问下画的流程图用的是什么工具？

Re: [cgicgicgl](#) 2016-01-16 12:26发表



回复hua504358405: 用excel可以画出你想要的所有图形

2楼 [jsfansw](#) 2015-03-17 16:12发表



你好，最近我在做东西的时候，遇到了一些关于xml rpc方面的困难，不知道你是否能够帮我指点一下。具体的问题是这样的，服务器端已经被别人实现了，我现在是想写一个客户端，能够发送一些命令让服务器执行，但是我不知道服务器端处理这个命令的具体函数是什么。有没有什么方法能够获取到服务器所提供的服务函数？

1楼 [jsfansw](#) 2015-03-17 16:12发表



你好，最近我在做东西的时候，遇到了一些关于xml rpc方面的困难，不知道你是否能够帮我指点一下。具体的问题是这样的，服务器端已经被别人实现了，我现在是想写一个客户端，能够发送一些命令让服务器执行，但是我不知道服务器端处理这个命令的具体函数是什么。有没有什么方法能够获取到服务器所提供的服务函数？

Re: 鹏鹏超人 2015-09-30 10:38发表



回复jsfansw: 通信双方要遵照一个接口规范的，要不怎么通信？或者是服务端提供出调用方式，供你调用。

Re: -mindwind- 2015-03-18 10:33发表



回复jsfansw: 是用的 webservice 么

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

\* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

- 全部主题
- Hadoop    AWS    移动游戏    Java    Android    iOS    Swift    智能硬件    Docker
- OpenStack    VPN    Spark    ERP    IE10    Eclipse    CRM    JavaScript    数据库    Ubuntu    NFC
- WAP    jQuery    BI    HTML5    Spring    Apache    .NET    API    HTML    SDK    IIS    Fedora    XML
- LBS    Unity    Splashtop    UML    components    Windows Mobile    Rails    QEMU    KDE    Cassandra
- CloudStack    FTC    coremail    OPhone    CouchBase    云计算    iOS6    Rackspace    Web App
- SpringSide    Maemo    Compuware    大数据    aptech    Perl    Tornado    Ruby    Hibernate    ThinkPHP
- HBase    Pure    Solr    Angular    Cloud Foundry    Redis    Scala    Django    Bootstrap