《设计文档》

选题：YA-RPC框架

开发者：刘大川

学号：202121081035

班级：分布式系统2班

# 1 引言

## 编写目的

RPC（远程过程调用）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。在当前分布式微服务治理盛行的背景下，RPC解决了服务之间的调用问题，使得远程调用时，要能够像本地调用一样方便，让调用者感知不到远程调用的逻辑。

本项目自制简易RPC框架是为了增加对RPC远程过程调用的理解，实践分布式系统课程理论知识，提升个人的编程能力与探索积累分布式系统的开发经验。

## 背景

本项目是一个简易的基于Java语言开发的自制RPC框架—YA-RPC（Yet-Another RPC Framework）

本项目的主要功能是:

1. 支持 At-least-once 语义
2. 能够实现如下API：
   1. 远程调用 float sum(float a, float b)
   2. 远程调用 string uppercase(str)

## 定义

**RPC：**远程过程调用[(Remote procedure call)](https://links.jianshu.com/go?to=https://en.wikipedia.org/wiki/Remote_procedure_call" \t "_blank)，它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思想。

**ZooKeeper：**是一个[分布式](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F/19276232" \t "https://baike.baidu.com/item/zookeeper/_blank)的，开放源码的[分布式应用程序](https://baike.baidu.com/item/%E5%88%86%E5%B8%83%E5%BC%8F%E5%BA%94%E7%94%A8%E7%A8%8B%E5%BA%8F/9854429" \t "https://baike.baidu.com/item/zookeeper/_blank)协调服务。它是一个为分布式应用提供一致性服务的软件，提供的功能包括：配置维护、域名服务、分布式同步、组服务等。在本项目中作为注册中心。

**Java RMI：**远程方法调用([Remote Method Invocation](https://links.jianshu.com/go?to=https://en.wikipedia.org/wiki/Distributed_object_communication" \t "_blank))，一种用于实现远程过程调用(RPC)的Java API， 能直接传输序列化后的Java对象和[分布式垃圾收集](https://links.jianshu.com/go?to=https://en.wikipedia.org/wiki/Distributed_Garbage_Collection" \t "_blank)。它的实现依赖于[Java虚拟机](https://links.jianshu.com/go?to=https://en.wikipedia.org/wiki/Java_Virtual_Machine" \t "_blank)(JVM)，因此它仅支持从一个JVM到另一个JVM的调用。

## 参考资料

1. 《深入浅出 RPC - 深入篇》CSDN：<https://blog.csdn.net/mindfloating/article/details/39474123>
2. 《ZooKeeper入门看这篇就够了》CSDN：<https://blog.csdn.net/java_66666/article/details/81015302>
3. 《RMI实现RPC》CSDN：<https://blog.csdn.net/u014565127/article/details/113583536>

# 2 总体设计

## 2.1 需求规定

本框架需要满足功能：

1. 支持 At-least-once 语义
2. 能够实现如下API：
   1. 远程调用 float sum(float a, float b)
   2. 远程调用 string uppercase(str)

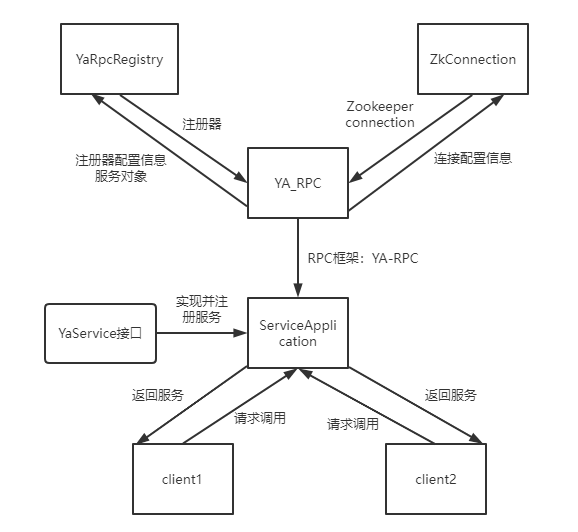
## 2.2 运行环境

本框架开发运行环境为：

硬件：AMD Ryzen 5 3600 6-Core Processor 3.59 GHz,16GB,GTX1030.

软件：idea，JDK1.8，ZooKeeper3.5.5

## 2.3 软件架构



## 2.4 模块设计

|  |  |  |
| --- | --- | --- |
| 编号 | 模块名称 | 详细功能分配 |
| 01 | 框架模块 | 1. YA\_RPC：提供YA-RPC框架对外使用的接口，为注册器和连接器提供初始化信息。 2. YaRpcRegistry：用于服务注册及获取服务对象。 3. ZkConnection：ZooKeeper连接器。 |
| 02 | 服务器模块 | 1. api：YaService接口，其中方法有sum()与uppercase()。 2. ServiceApplication：调用RPC，完成服务注册，等待client访问。 |
| 03 | 客户端模块 | 利用RPC框架，获取服务器已注册的服务并使用。 |

# 3 接口设计

## 3.1 registerService

| **字段名** | **类型** | **必填项** |
| --- | --- | --- |
| serviceInterface | Class<? extends Remote> | 是 |
| serviceObject | Remote | 是 |

**Params:**

**Return:** null

## 3.2 registerService

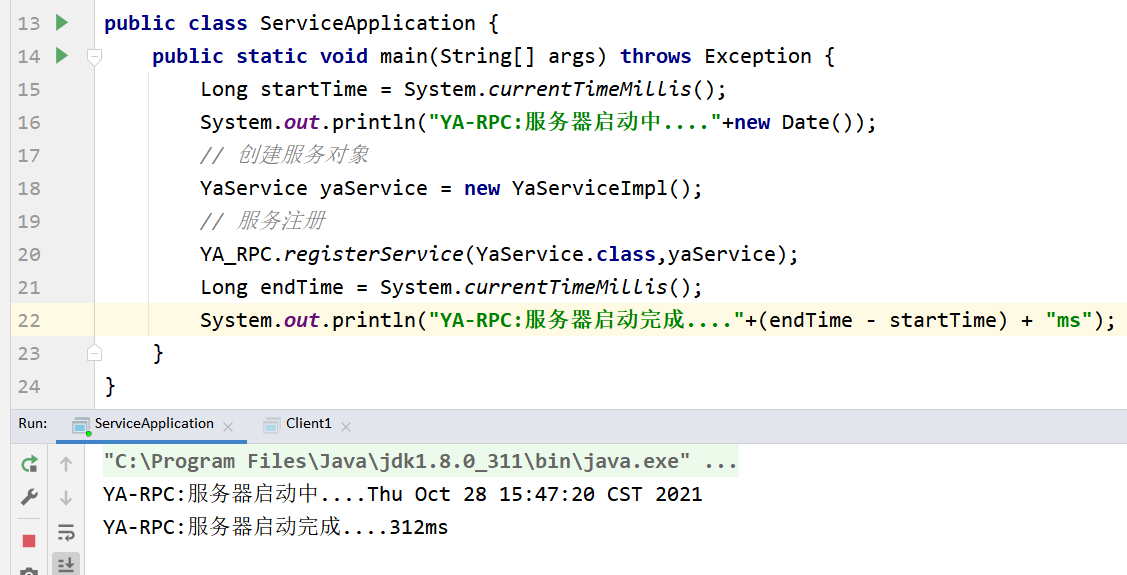
| **字段名** | **类型** | **必填项** |
| --- | --- | --- |
| serviceInterface | Class<T> | 是 |

**Params:**

**Return:** <T extends Remote> T

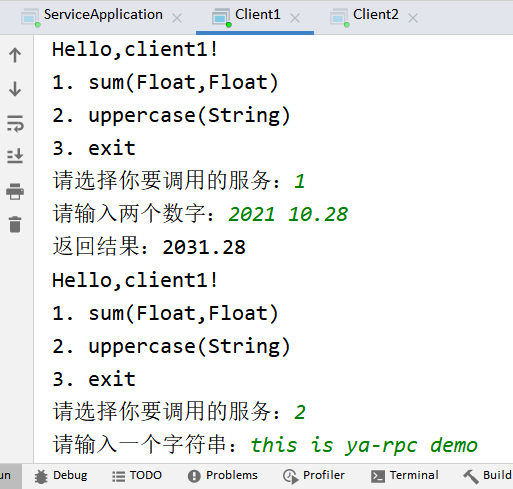
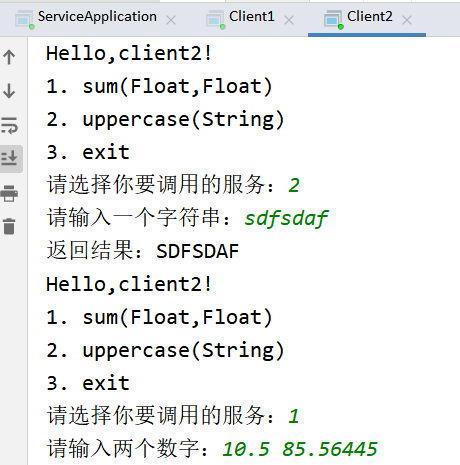
# 4 测试案例

## 4.1 服务器启动

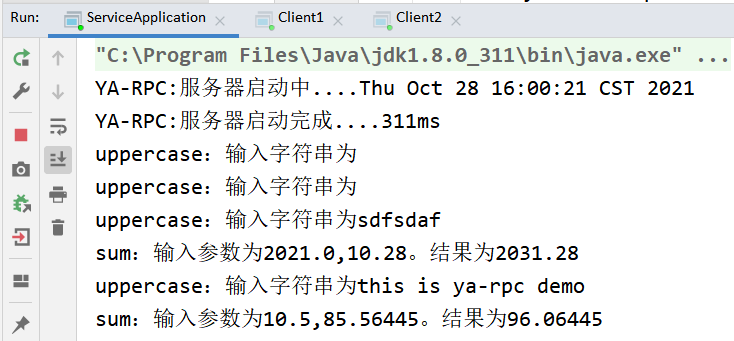


## 4.2 客户端启动

Client1与Client2：

服务器状态：



## 4.3 ZooKeeper状态

