mprpc分布式网络通信框架项目

> **请勿随意拷贝，引用必指明出处，违者必究！**
>
> **施磊老师腾讯课堂地址：** https://fixbug.ke.qq.com
>
> **业界优秀的RPC框架：baidu的brpc，google的grpc**
>
> **项目基于muduo高性能网络库+Protobuf开发，所以命名项目为mprpc**

目录

技术栈

- 集群和分布式概念以及原理
- RPC远程过程调用原理以及实现
- Protobuf数据序列化和反序列化协议
- ZooKeeper分布式一致性协调服务应用以及编程
- muduo网络库编程
- conf配置文件读取
- 异步日志
- CMake构建项目集成编译环境
- github管理项目

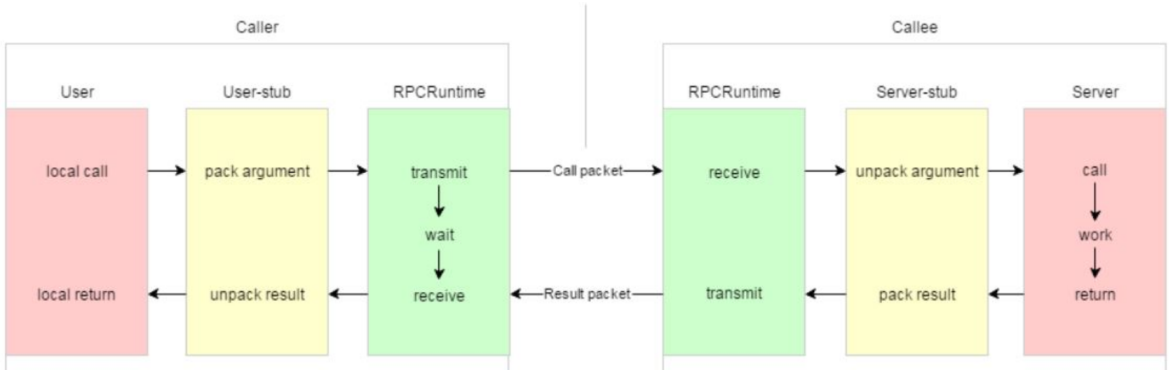
集群和分布式

集群：每一台服务器独立运行一个工程的所有模块。

分布式：一个工程拆分了很多模块，每一个模块独立部署运行在一个服务器主机上，所有服务器协同工作共同提供服务，每一台服务器称作分布式的一个节点，根据节点的并发要求，对一个节点可以再做节点模块集群部署。

RPC通信原理

RPC（Remote Procedure Call Protocol）远程过程调用协议。



黄色部分：设计rpc方法参数的打包和解析，也就是数据的序列化和反序列化，使用Protobuf。

绿色部分：网络部分，包括寻找rpc服务主机，发起rpc调用请求和响应rpc调用结果，使用muduo网络库和zookeeper服务配置中心（专门做服务发现）。

mprpc框架主要包含以上两个部分的内容。

环境配置使用

项目代码工程目录

bin：可执行文件
build：项目编译文件
lib：项目库文件
src：源文件
test：测试代码
example：框架代码使用范例
CMakeLists.txt：顶层的cmake文件
README.md：项目自述文件
autobuild.sh：一键编译脚本

vscode远程开发Linux项目

muduo网络库编程示例

CMake构建项目集成编译环境

请参考：面向校招系列-C++项目_集群聊天服务器项目

> **Linux环境下搭建muduo网络库**

muduo库的安装需要依赖boost库，具体安装配置步骤参考我的博客：
https://blog.csdn.net/QIANGWEIYUAN/article/details/89023980

> **网络I/O模型介绍**

accept + read/write

不是并发服务器

accept + fork - process-pre-connection

适合并发连接数不大，计算任务工作量大于fork的开销

accept + thread thread-pre-connection

比方案2的开销小了一点，但是并发造成线程堆积过多

muduo的设计：reactors in threads - one loop per thread

方案的特点是one loop per thread，有一个main reactor（I/O）负载accept连接，然后把连接分发到某个sub reactor（Worker），该连接的所用操作都在那个sub reactor所处的线程中完成，多个连接可能被分派到多个线程中，以充分利用CPU。

如果有过多的耗费CPU I/O的计算任务，可以创建新的线程专门处理耗时的计算任务。

reactors in process - one loop pre process

nginx服务器的网络模块设计，基于进程设计，采用多个Reactors充当I/O进程和工作进程，通过一把accept锁，完美解决多个Reactors的“惊群现象”。

Protobuf安装配置

protobuf（protocol buffer）是google 的一种数据交换的格式，它独立于平台语言。
google 提供了protobuf多种语言的实现：java、c#、c++、go 和 python，每一种实现都包含了相应语言的编译器以及库文件。
由于它是一种二进制的格式，比使用 xml（20倍）、json（10倍）进行数据交换快许多。可以把它用于分布式应用之间的数据通信或者异构环境下的数据交换。作为一种效率和兼容性都很优秀的二进制数据传输格式，可以用于诸如网络传输、配置文件、数据存储等诸多领域。

> **ubuntu protobuf环境搭建**

见**项目资料下载地址**或者**自己在github源代码下载**地址：https://github.com/google/protobuf
源码包中的src/README.md，有详细的安装说明，安装过程如下：

1、解压压缩包：unzip protobuf-master.zip
2、进入解压后的文件夹：cd protobuf-master
3、安装所需工具：sudo apt-get install autoconf automake libtool curl make g++ unzip
4、自动生成configure配置文件：./autogen.sh
5、配置环境：./configure
6、编译源代码(时间比较长)：make

sudo yum
其中c++应该yum install gcc-c++
第四步前要执行 libtoolize

7、安装：sudo make install
8、刷新动态库：sudo ldconfig

要求：熟悉Protobuf的proto配置文件内容，以及protoc的编译命令。

```
syntax = "proto3";

package fixbug;

option cc_generic_services = true;

message LoginRequest {
    string name = 1;
    string pwd = 2;
}

message RegRequest {
    string name = 1;
    string pwd = 2;
    int32 age = 3;
    enum SEX {
        MAN = 0;
        WOMAN = 1;
    }
    SEX sex = 4;
    string phone = 5;
}

message Response {
    int32 errorno = 1;
    string errormsg = 2;
    bool result = 3;
}

// 定义RPC服务接口类和服务方法
service UserServiceRpc{
    rpc login(LoginRequest) returns (Response);
    rpc reg(RegRequest) returns (Response);
}
```

mprpc框架设计

见项目代码交互图片

Zookeeper分布式协调服务

Zookeeper是在分布式环境中应用非常广泛，它的优秀功能很多，比如分布式环境中全局命名服务，服务注册中心，全局分布式锁等等。

参考链接：https://www.cnblogs.com/xinyonghu/p/11031729.html

> **zk的数据是怎么组织的 - znode节点**

> **zk的watcher机制**

见**项目资料下载地址**下载 zookeeper-3.4.10.tar.gz，解压后

1、zookeeper-3.4.10$ cd conf
zookeeper-3.4.10/conf$ mv zoo_sample.cfg zoo.cfg

2、进入bin目录，启动zkServer， ./zkServer.sh start

3、可以通过netstat查看zkServer的端口，在bin目录启动zkClient.sh链接zkServer，熟悉zookeeper怎么组织节点

zk的原生开发API（c/c++接口）

进入上面解压目录src/c下面，zookeeper已经提供了原生的C/C++和Java API开发接口，需要通过源码编译生成，过程如下：

~/package/zookeeper-3.4.10/src/c$ sudo ./configure
~/package/zookeeper-3.4.10/src/c$ sudo make
~/package/zookeeper-3.4.10/src/c$ sudo make install

主要关注zookeeper怎么管理节点，zk-c API怎么创建节点，获取节点，删除节点以及watcher机制的API编程。

znode节点存储格式

zk客户端常用命令

ls、get、create、set、delete

原生ZkClient API存在的问题

Zookeeper原生提供了C和Java的客户端编程接口，但是使用起来相对复杂，几个弱点：

1.不会自动发送心跳消息 <==== **错误，源码上会在1/3的Timeout时间发送ping心跳消息**
2.设置监听watcher只能是一次性的，每次触发后需要重复设置
3.znode节点只存储简单的byte字节数组，如果存储对象，需要自己转换对象生成字节数组

项目资料下载

见项目资料文件夹