1. 介绍

什么是redisPlatform？它主要利用redis服务器框架作为平台，让开发者专注于业务逻辑的开发，使用C/C++编写高效率的模块，并编译成.so的动态库，加载到redis服务器平台中，让使用者通过相关命令发送到redis，redis再调用so服务器模块来处理，而这种so服务模块可以被加载到多个redis服务器中，redis也可以加载多个so服务模块，根据发送过来的命令来调用相应的so模块进行处理，并将处理结果返回给使用者。

因为很多时候大部分事情都是一样的，不一样的只是业务逻辑，而这个RedisPlatform就是为此而生，可能有人有疑问，可以使用它的lua做业务逻辑，但是它的lua做了很多限制，使得无法做一些事情，例如定时回调，获取系统时间等等。

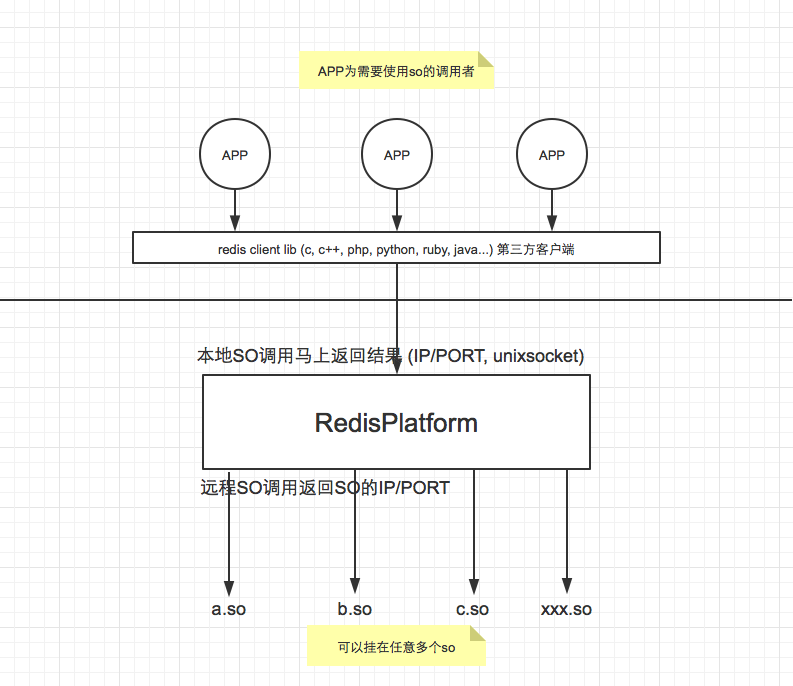
与cgi最大的区别：

1. 速度比cgi快， cgi进程需要启动解析器、加载配置（如业务相关配置）、连接其它服务器（如数据库服务器）、逻辑处理等。而这些在redis加载的so模块中会先初始化好，无需每次都要做这些过程。
2. cgi对每个请求都需要启动一个子进程来执行任务，非常耗资源，而这个在redisplatform中是不需要的，它使用先进的异步IO，并且使用ring buffer方式分发任务。不会因为请求过多，造成机器问题，服务器问题等等。
3. 可以利用redis的众多优势，提供很多便利，例如哨兵，提供强大的容灾。
4. 如果是纯内存操作，那么将可以达到其他set，get操作一样的速度。

……

1. 架构

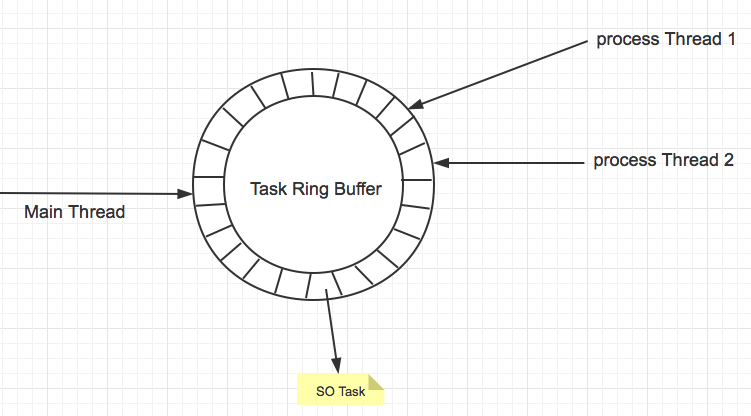
说了这么多，先看看它的先进架构：



1. 功能

这个redisPlatform提供的功能如下：

1. 本地so调用。 如果要使用的服务在这个RedisPlatform中，那么redis处理完就会返回相应的结构，访问这个redisplatform的方式跟访问redis的方式一样。（已实现）
2. 远程so调用。 如果要访问的服务不在这个RedisPlatform中，那么redis返回so对应的redisplatform的IP/PORT，需要客户端配合进行远程访问。（已实现）
3. 多进程模型：为防止阻塞性任务，例如db的访问等等，需要用多线程模型。而这个多线程模型采用先进的ring buffer（from id engine）模式实现，线程数量可以在配置文件中配置。（TODO）



当前还是用单进程的方式运行，只能应付不阻塞的业务操作，

准备采用的多进程模式（TODO）

1. 支持bitmsg（TODO）
2. 支持多个SO服务， 根据发送过来的key来导向相应的SO服务，这个key自己可以定义为命令的方式（已实现）
3. SO支持定时器回调，可以指定SO多久回调一次，然后调用SO相应的函数进行操作。（已实现）
4. 支持消息加密系统（TODO）
5. 动态更新SO，它将会全部更新一次配置文件配置的SO模块。（已实现）
6. 支持客户端同步异步操作，这个主要依赖客户端的实现，目前hredis支持。（已实现）
7. 自动化生成模板（目前只能手动拷贝，准备用python实现,TODO）
8. 使用
9. 编写C/C++ SO 功能服务模块

CPP模块请参见项目模板cpptest.h, cpptest.cpp的写法

C模块请参见项目模板test.h, test.c的写法

简单来说只需要两点即可。

1. 实现OnInit()函数，这个函数会被redis调用初始化SO模块。
2. 声明一个rso 结构体变量，即rsao变量。

rso结构体说明：

hasTimer字段指定定时器的回调时间，单位毫秒, 0为不使用启动定时器。

timerID字段是redis赋予给它的定时器ID

onExec函数回调指定了请求发送过来时的回调函数，可以为空，空不回调。

onTimer函数回调指定了定时器到时间时回调的函数，可以为空，空不回调。

1. 编译成 SO 模块

参见makefile文件

1. 在配置文件rso.conf加上自己的SO模块路径，已经对应的key（CMD）

#本地so配置

loadrso 0x12345 ../solib/test.so

#远程so配置

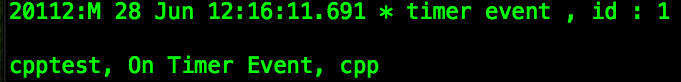
loadremoterso 0x12346 192.168.0.1:6379

1. 使用redis的客户端发送命令测试

启动服务器：

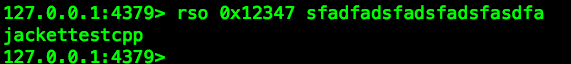
./redis-server ../redis.conf

定时器测试结果如图：



我们发送rso命令给redis，看看它的返回，目前我们在cpptest.cpp中返回jackettestcpp字符串，现在验证一下：

启动客户端并发送命令：下图是客户端收到的结果

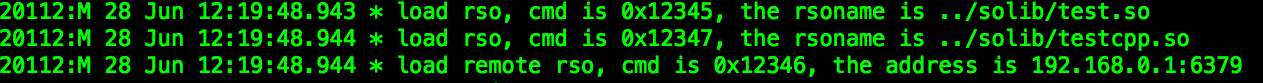


1. 动态更新SO

我们只需要发送命令 reloadrso 命令即可:



观察服务器重新加载结果:



测试成功。

1. 压力测试数据（速度）(TODO)
2. 代码地址：稍后公布
3. 建议：请发送到jacketzhong@gmail.com

（例如这个功能对你是否有用，需要增加什么功能）