服务器并发处理框架文档

CareDear Service Team

August 4, 2014

Contents

Abstract

本文档介绍一种处理并发的服务器后台运行程序框架，暂叫成CDS(CareDear Service)

1 简介

目前服务器程序框架取自memcached([1])，并计划将当中数据库处理逻辑简化并替换成自己需要的内容。

1.1 依赖关系

代码基于如下软件版本开发:

依赖包	说明
libevents 2.0.21-stable	event通知库
memcached 1.4.20	程序框架开发基于版本

同时还需要如下软件包/库：

依赖包	说明
MySQL-dev	C语言访问MySQL的函数借口
LibXML2	解析XML配置文件
LibCrypto	用来解密token key

1.2 编译代码

当前程序库支持binary/.so/.a三种不同编译结果。为简化执行程序部署步骤，默认情况下程序是静态库编译，所以文件比较大。

编译方法：

```
$ make cds          可运行二进制文件cds
$ make cs           动态库文件libcds.so
$ make ca           静态库文件cds.a
$ make testcode     编译测试demo代码，倚赖.so库
$ make all          默认编译结果，生成上述所有文件
```

若想编译成动态库方式，则在编译时指定STATICLIB数值为false：

```
$ make STATICLIB=false                  动态链接可执行程序
$ make STATICLIB=false DEBUG=false      动态链接可执行程序，且是release版本
```

1.3 运行服务器状态查看

从21k-code机器上使用alias的ksxmpp登录，然后在登录上的机器上再使用ksapp3来登录运行程序log存放的位置/home/tokenverify

1.4 最简单服务端调用

框架代码初步设计思想是负责并发请求的接受以及数据的读写。数据处理逻辑由各个调用者来完成。

test/server_demo.c演示如下场景：

- Client发送数值请求
- 服务端利用框架接收和返回处理结果。
- 服务端代码主要是处理Client发来的数据，demo代码是简单double.

代码：

```
1  服务端自己提供处理数据的回调函数，框架只管数据获取和返回
2  int my_handler(int size, void *req, int *len, void *resp){
3      printf("i will process %s\n", param);
4      int i = atoi(param);
5      if(i  > 0)
6      {
7          i = (i * 2);
8          sprintf(resp, "H->%d", i);
9          *len = strlen(resp);
10     }
11     return 0;
12 }
13
14 int main(int argc, char **argv){
15     struct addition_config cfg;
16     cfg.ac_cfgfile = "config.xml";
17     cfg.ac_handler = my_handler;
18     cds_init(&cfg, argc, argv);
19
20     return 0;
21 }
```

2 Token验证模块

2.1 简介

该模块代码置于token_auth_service/目录下。主要功能是校验token的合法性，及更新数据库中的用户token过期时间信息。

2.2 消息交互

3 家庭圈服务模块

3.1 简介

代码位置位于circle_mgr_service/目录

3.2 消息交互

4 网盘服务模块

5 使用范例

5.1 配置文件

程序运行中的参数配置目前定义在/etc/cds_cfg.xml文件中。

xmpp的校验程序(auth)以及token有效期检查等程序都公用这个配置文件，其配置信息如下：

```
<config>
    <tokenserver>
      <ip>127.0.0.1</ip>
      <port>11234</port>
    </tokenserver>

    <sqlserver>
      <ip>10.128.0.27</ip>
      <port>0</port>
      <user>ucen</user>
      <password>heqi</password>
      <databasename>ucen</databasename>
    </sqlserver>
</config>
```

这些配置信息用于控制程序运行时使用的参数：

- tokenserver段是配置auth程序需要连接的token服务端IP和端口值
- sqlserver段是设置token服务器需要访问的MySQL机器的IP地址，端口，用户名，密码以及数据库名称。

在实际使用过程中，我们可以修改配置文件来调整程序运行参数，达到不用重新编译程序的目的。

5.2 Token验证

5.3 处理错误码

服务端的处理结果使用int（4-byte)通知客户端。

其错误码定义：

错误码代号	说明
CDS_OK(0)	处理正确的结果
CDS_ERR_REQ_TOOLONG(1)	客户端请求数据超出范围
CDS_ERR_NOMEMORY(2)	内存不足
CDS_ERR_REQ_INVALID(3)	客户端请求的数据格式无效
CDS_ERR_UNMATCH_USER_INFO(4)	客户端请求的用户信息和token的不一致
CDS_ERR_USER_TOKEN_EXPIRED(5)	用户的token已经过有效期
CDS_ERR_SQL_DISCONNECTED(6)	服务器端的MySQL访问出错
CDS_ERR_NO_RESOURCE(7)	系统无可用资源(如无法正确生成IPC对象等)

定义的代码文件是cds_public.h

5.4 客户端示例

客户端代码简单地通过TCP连接server相应端口。

示例代码test/client_demo.c

6 数据库操作

7 TODO List

- 清除无用代码
- 去掉所有编译Warning
- 增加及优化库函数的原型定义

References

[1] memcached: http://memcached.org, a high-performance cache system.

[2] Libevent: http://libevent.org, an open source event notification library.