

《分布式流媒体》实训项目

C/C++教学体系

TNV DAY08

直播课



目录

服务器类(server_c)

主函数(main)

构建脚本(Makefile)

配置文件(tracker.cfg)

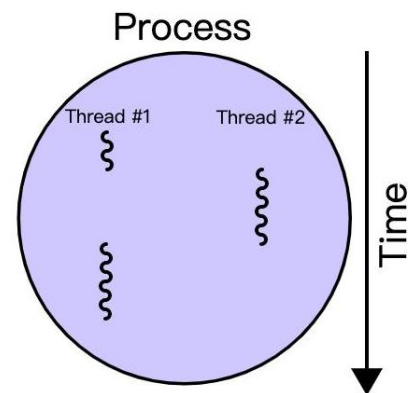
建表脚本(tracker.sql)

服务器类(server_c)



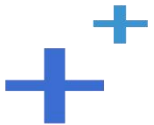
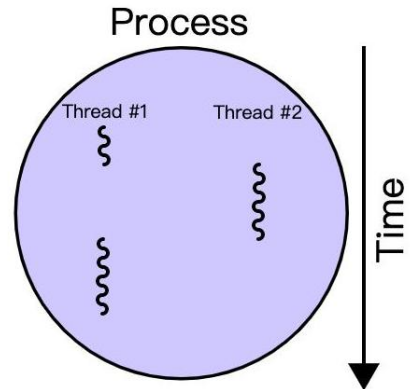
线程级回调方法

- 线程获得连接时被调用: `thread_on_accept`
 - 返回true, 连接将被用于后续通信, 否则函数返回后即关闭连接
 - 打印日志
- 线程连接可读时被调用: `thread_on_read`
 - 返回true, 保持长连接, 否则函数返回后即关闭连接
 - 接收包头
 - 业务处理



线程级回调方法

- 线程读写超时时被调用: `thread_on_timeout`
 - 返回true, 继续等待下一次读写, 否则函数返回后即关闭连接
 - 打印日志
 - 返回true以保持连接
- 线程连接关闭时被调用: `thread_on_close`
 - 打印日志

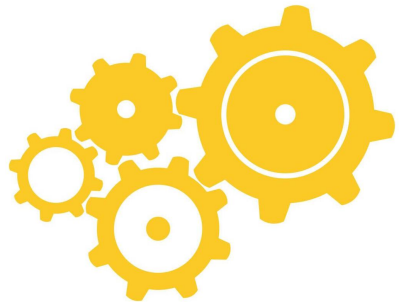


主函数(main)

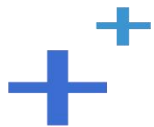


主函数(main)

- 初始化ACL库
 - `acl::acl_cpp_init();`
 - `acl::log::stdout_open(true);`
- 创建并运行服务器
 - `server_c& server = acl::singleton2<server_c>::get_instance();`
 - `server.set_cfg_str(cfg_str);`
 - `server.set_cfg_int(cfg_int);`
 - `server.run_alone("127.0.0.1:21000", "../etc/tracker.cfg");`

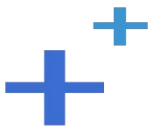


构建脚本(Makefile)



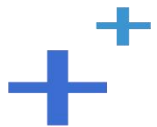
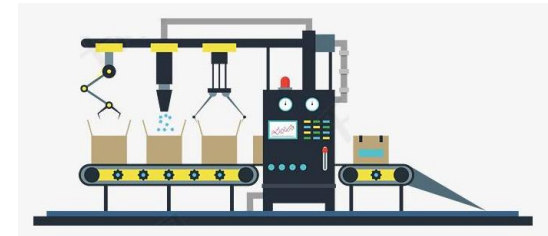
构建脚本(Makefile)

```
PROJ = ../bin/tracker
OBJS = $(patsubst %.cpp, %.o, $(wildcard ../01_common/*.cpp *.cpp))
CC = g++
LINK = g++
RM = rm -rf
CFLAGS = -c -Wall -I/usr/include/acl-lib/acl_cpp `mysql_config --cflags` -I../01_common
LIBS = -pthread -lacl_all `mysql_config --libs`
```

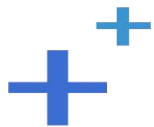


构建脚本(Makefile)

```
all: $(PROJ)
$(PROJ): $(OBJS)
    $(LINK) $^ $(LIBS) -o $@
.cpp.o:
    $(CC) $(CFLAGS) $^ -o $@
clean:
    $(RM) $(PROJ) $(OBJS)
```

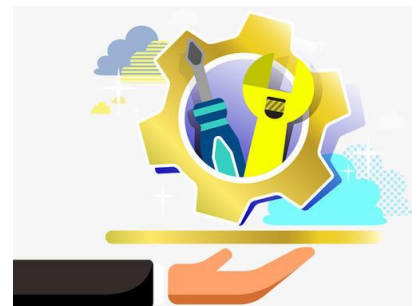


配置文件(tracker.cfg)

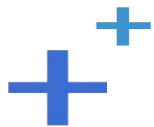


配置文件(tracker.cfg)

- 应用ID表: `tnv_apps_id = tnvideo`
- MySQL地址表: `mysql_addrs = 127.0.0.1`
- Redis地址表: `redis_addrs = 127.0.0.1:6379`
- 存储服务器状态检测间隔秒数: `check_active_interval = 120`
- MySQL读写超时: `mysql_rw_timeout = 30`
- Redis连接池最大连接数: `redis_max_conn_num = 600`
- Redis连接超时: `redis_conn_timeout = 10`
- Redis读写超时: `redis_rw_timeout = 10`
- Redis键超时: `redis_key_timeout = 60`



建表脚本(tracker.sql)



建表脚本(tracker.sql)

```
USE tnv_trackerdb
DROP TABLE IF EXISTS `t_groups_info`;
CREATE TABLE `t_groups_info` (
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `group_name` varchar(32) DEFAULT NULL,
  `create_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,
  `update_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
INSERT INTO `t_groups_info` (`group_name`) VALUES ('group001');
```

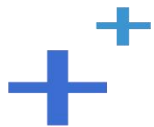


建表脚本(tracker.sql)

```
DROP TABLE IF EXISTS `t_router`;  
CREATE TABLE `t_router` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `group_name` varchar(32) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `update_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON  
  UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



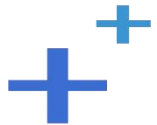
附录：程序清单



TNV/src/02_tracker/12_server.cpp

```
// 线程获得连接时被调用
// 返回true, 连接将被用于后续通信, 否则
// 函数返回后即关闭连接
bool server_c::thread_on_accept(acl::socket_stream* conn) {
    logger("connect, from: %s", conn->get_peer());
    return true;
}

// 与线程绑定的连接可读时被调用
// 返回true, 保持长连接, 否则
// 函数返回后即关闭连接
bool server_c::thread_on_read(acl::socket_stream* conn) {
    // 接收包头
    char head[HEADLEN];
    if (conn->read(head, HEADLEN) < 0) {
```

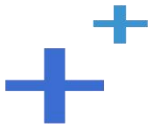


TNV/src/02_tracker/12_server.cpp

```
        if (conn->eof())
            logger("connection has been closed, from: %s",
                  conn->get_peer());
        else
            logger_error("read fail: %s, from: %s",
                          acl::last_serror(), conn->get_peer());
        return false;
    }
```

```
    // 业务处理
    service_c service;
    return service.business(conn, head);
}
```

```
// 线程读写连接超时时被调用
```



TNV/src/02_tracker/12_server.cpp

// 返回true, 继续等待下一次读写, 否则

// 函数返回后即关闭连接

```
bool server_c::thread_on_timeout(acl::socket_stream* conn) {  
    logger("read timeout, from: %s", conn->get_peer());  
    return true;  
}
```

// 与线程绑定的连接关闭时被调用

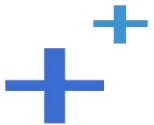
```
void server_c::thread_on_close(acl::socket_stream* conn) {  
    logger("client disconnect, from: %s", conn->get_peer());  
}
```



TNV/src/02_tracker/13_main.cpp

```
// 跟踪服务器
// 定义主函数
//
#include "01_globals.h"
#include "11_server.h"

int main(void) {
    // 初始化ACL库
    acl::acl_cpp_init();
    acl::log::stdout_open(true);
    // 创建并运行服务器
    server_c& server = acl::singleton2<server_c>::get_instance();
    server.set_cfg_str(cfg_str);
    server.set_cfg_int(cfg_int);
    server.run_alone("127.0.0.1:21000", "../etc/tracker.cfg");
    return 0;
}
```



TNV/src/02_tracker/Makefile

```
PROJ    = ../../bin/tracker
OBJS    = $(patsubst %.cpp, %.o, $(wildcard ../01_common/*.cpp *.cpp))
CC      = g++
LINK    = g++
RM      = rm -rf
CFLAGS  = -c -Wall -I/usr/include/acl-lib/acl_cpp `mysql_config --cflags` -I../01_common
LIBS    = -pthread -lacl_all `mysql_config --libs`
```

```
all: $(PROJ)
```

```
$(PROJ): $(OBJS)
        $(LINK) $^ $(LIBS) -o $@
```

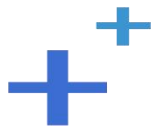
```
.cpp.o:
        $(CC) $(CFLAGS) $^ -o $@
```

```
clean:
        $(RM) $(PROJ) $(OBJS)
```



TNV/etc/tracker.cfg

```
service tracker {  
    # 应用ID表  
    tnv_apps_id = tnvideo  
    # MySQL地址表  
    mysql_addrs = 127.0.0.1  
    # Redis地址表  
    redis_addrs = 127.0.0.1:6379  
    # 存储服务器状态检查间隔秒数  
    check_active_interval = 120  
    # MySQL读写超时  
    mysql_rw_timeout = 30  
    # Redis连接池最大连接数  
    redis_max_conn_num = 600  
    # Redis连接超时  
    redis_conn_timeout = 10  
    # Redis读写超时  
    redis_rw_timeout = 10  
    # Redis键超时  
    redis_key_timeout = 60  
}
```



TNV/sql/tracker.sql

```
DROP DATABASE IF EXISTS tnv_trackerdb;  
CREATE DATABASE tnv_trackerdb;  
USE tnv_trackerdb;
```

```
CREATE TABLE `t_groups_info` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `group_name` varchar(32) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `update_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



TNV/sql/tracker.sql

```
INSERT INTO `t_groups_info` (`group_name`) VALUES ('group001');
```

```
CREATE TABLE `t_router` (  
  `id` int(11) unsigned NOT NULL AUTO_INCREMENT,  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `group_name` varchar(32) DEFAULT NULL,  
  `create_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP,  
  `update_time` timestamp NULL DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```



复习课见