欢迎大家来到第五阶段课程

《分布式流媒体》实训项目

TNV DAY10

复习课

预习内容

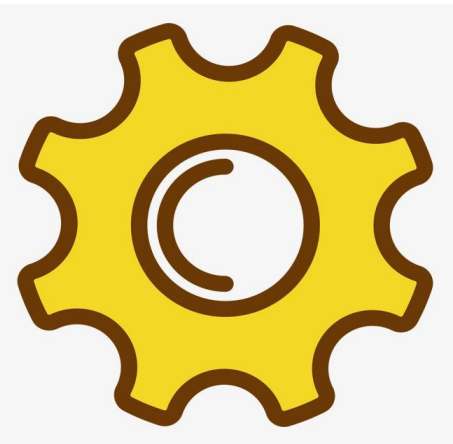
客户机（10）

客户机（10）

客户机类(client_c)的方法

知识讲解

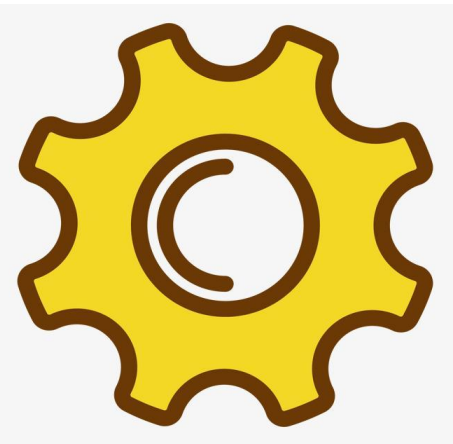
- 从存储服务器下载文件：download
  - 从跟踪服务器获取存储服务器地址列表
  - 顺序抽取一台存储服务器地址，获取相应的连接池
    - 失败：创建针对该存储服务器的连接池并再次获取
      - 失败：尝试下一台存储服务器
  - 从针对该存储服务器的连接池中获取一个空闲连接
    - 失败：尝试下一台存储服务器
  - 通过该连接向存储服务器下载文件
    - 套接字通信错误
      - 关闭连接并尝试下一个连接
    - 否则
      - 成功：释放连接
      - 失败：关闭连接
  - 返回处理结果



客户机类(client_c)的方法

知识讲解

- 删除存储服务器上的文件：del
  - 从跟踪服务器获取存储服务器地址列表
  - 顺序抽取一台存储服务器地址，获取相应的连接池
    - 失败：创建针对该存储服务器的连接池并再次获取
      - 失败：尝试下一台存储服务器
  - 从针对该存储服务器的连接池中获取一个空闲连接
    - 失败：尝试下一台存储服务器
  - 通过该连接删除存储服务器上的文件
    - 套接字通信错误
      - 关闭连接并尝试下一个连接
    - 否则
      - 成功：释放连接
      - 失败：关闭连接
  - 返回处理结果

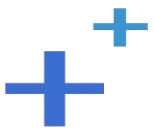


附录：程序清单

TNV/src/05_client/08_client.cpp

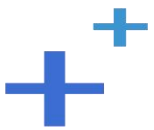
知识讲解

```
//从存储服务器下载文件
int client_c::download(char const* appid, char const* userid,
        char const* fileid, long long offset, long long size,
        char** filedata, long long* filesize) {
        //检查参数
        if (!appid || !strlen(appid)) {
                logger_error("appid is null");
                return ERROR;
        }
        if (!userid || !strlen(userid)) {
                logger_error("userid is null");
                return ERROR;
        }
        if (!fileid || !strlen(fileid)) {
                logger_error("fileid is null");
```



TNV/src/05_client/08_client.cpp

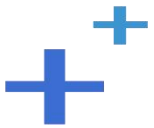

```
        return ERROR;
}

//从跟踪服务器获取存储服务器地址列表
int result;
std::string ssaddrs;
if ((result = saddrs(appid, userid, fileid, ssaddrs)) != OK)
        return ERROR;
std::vector<std::string> vsaddrs;
split(ssaddrs.c_str(), vsaddrs);
if (vsaddrs.empty()) {
        logger_error("storage addresses is empty");
        return ERROR;
}
```



TNV/src/05_client/08_client.cpp

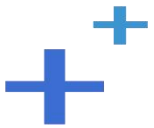
知识讲解

```
result = ERROR;

for (std::vector<std::string>::const_iterator saddr =
        vsaddrs.begin(); saddr != vsaddrs.end(); ++saddr) {
        //获取存储服务器连接池
        pool_c* spool = (pool_c*)s_mngr->get(saddr->c_str());
        if (!spool) {
                s_mngr->set(saddr->c_str(), s_scount);
                if (!(spool = (pool_c*)s_mngr->get(saddr->c_str()))) {
                        logger_warn(
                                "storage connection pool is null, saddr: %s",
                                saddr->c_str());
                        continue;
                }
        }
```



TNV/src/05_client/08_client.cpp

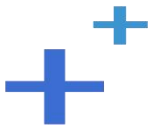

```
for (int sockerrs = 0; sockerrs < MAX_SOCKERRS; ++sockerrs) {
        //获取存储服务器连接
        conn_c* sconn = (conn_c*)spool->peek();
        if (!sconn) {
                logger_warn("storage connection is null, saddr: %s",
                        saddr->c_str());
                break;
        }

        //从存储服务器下载文件
        result = sconn->download(
                appid, userid, fileid, offset, size, filedata, filesize);

        if (result == SOCKET_ERROR) {
                logger_warn("download file fail: %s", sconn->errdesc());
```



TNV/src/05_client/08_client.cpp

```
                spool->put(sconn, false);
            }
            else {
                if (result == OK)
                    spool->put(sconn, true);
                else {
                    logger_error("download file fail: %s",
                            sconn->errdesc());
                    spool->put(sconn, false);
                }
                return result;
            }
        }
    }
```

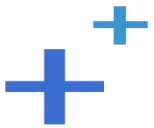
知识讲解



TNV/src/05_client/08_client.cpp

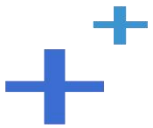
知识讲解

```
        return result;
}

//删除存储服务器上的文件
int client_c::del(char const* appid, char const* userid,
        char const* fileid) {
        //检查参数
        if (!appid || !strlen(appid)) {
                logger_error("appid is null");
                return ERROR;
        }
        if (!userid || !strlen(userid)) {
                logger_error("userid is null");
                return ERROR;
        }
```



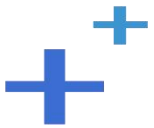
TNV/src/05_client/08_client.cpp


```
if (!fileid || !strlen(fileid)) {
        logger_error("fileid is null");
        return ERROR;
}

//从跟踪服务器获取存储服务器地址列表
int result;
std::string ssaddrs;
if ((result = saddrs(appid, userid, fileid, ssaddrs)) != OK)
        return ERROR;
std::vector<std::string> vsaddrs;
split(ssaddrs.c_str(), vsaddrs);
if (vsaddrs.empty()) {
        logger_error("storage addresses is empty");
        return ERROR;
```



TNV/src/05_client/08_client.cpp

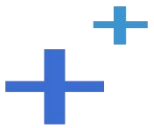

```
    }

    result = ERROR;

    for (std::vector<std::string>::const_iterator saddr =
            vsaddrs.begin(); saddr != vsaddrs.end(); ++saddr) {
            //获取存储服务器连接池
            pool_c* spool = (pool_c*)s_mngr->get(saddr->c_str());
            if (!spool) {
                    s_mngr->set(saddr->c_str(), s_scount);
                    if (!(spool = (pool_c*)s_mngr->get(saddr->c_str()))) {
                            logger_warn(
                                    "storage connection pool is null, saddr: %s",
                                    saddr->c_str());
                            continue;
```



TNV/src/05_client/08_client.cpp

知识讲解

```
            }
        }

        for (int sockerrs = 0; sockerrs < MAX_SOCKERRS; ++sockerrs) {
            //获取存储服务器连接
            conn_c* sconn = (conn_c*)spool->peek();
            if (!sconn) {
                logger_warn("storage connection is null, saddr: %s",
                        saddr->c_str());
                break;
            }

            //删除存储服务器上的文件
            result = sconn->del(appid, userid, fileid);

            if (result == SOCKET_ERROR) {
```



TNV/src/05_client/08_client.cpp

```
                    logger_warn("delete file fail: %s", sconn->errdesc());
                    spool->put(sconn, false);
                }
                else {
                    if (result == OK)
                        spool->put(sconn, true);
                    else {
                        logger_error("delete file fail: %s",
                                sconn->errdesc());
                        spool->put(sconn, false);
                    }
                    return result;
                }
            }
        }

        return result;
    }
```

知识讲解

下节课见