

5.1 IM即时通讯文件服务器设计

1 file_server分析

1.1 在线文件传输

1.2 离线文件传输

2 对应的命令

CID_OTHER_FILE_SERVER_IP_REQ

CID_OTHER_FILE_SERVER_IP_RSP

CID_FILE_REQUEST

CID_OTHER_FILE_TRANSFER_REQ

CID_OTHER_FILE_TRANSFER_RSP

CID_FILE_RESPONSE

CID_FILE_NOTIFY

CID_FILE_LOGIN_REQ

CID_FILE_LOGIN_RES

CID_FILE_STATE

CID_FILE_PULL_DATA_REQ

CID_FILE_PULL_DATA_RSP

DB_PROXY::hasOfflineFile 是否有离线文件

DB_PROXY::addOfflineFile 加入离线文件

DB_PROXY::delOfflineFile 删除离线文件

零声学院 <https://0voice.ke.qq.com>

讲师 Darren老师 QQ326873713

班主任 柚子老师 QQ2690491738

2022年6月28日

1 file_server分析

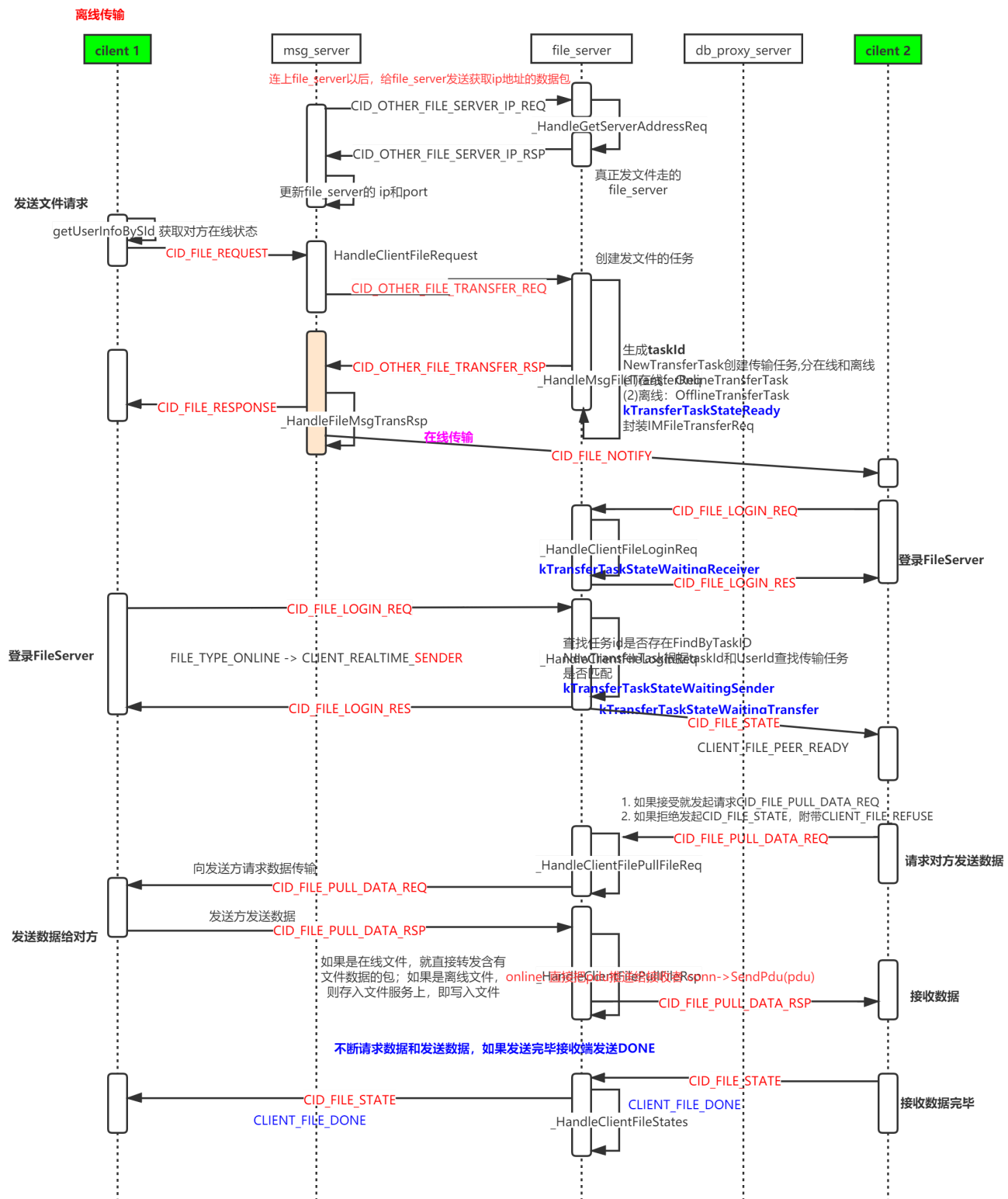
起源于IM.File.proto文件

文件传输分2类：

1. 在线文件传输；
2. 离线文件传输。

两种方式有一定的区别。依赖msg_server、file_server、db_proxy_server。

1.1 在线文件传输



在线文件传输原图路径: <https://www.processon.com/view/link/614f1062e0b34d69dd7a37bf>

1. 向msg_server使用CID_FILE_REQUEST命令请求文件传输, 封装IMFileReq结构体, 包含

from_user_id文件来自谁, to_user_id发给谁, file_name文件名, file_size文件大小, trans_mode在线/离线模式

2. msg_server收到CID_FILE_REQUEST请求后, 调用HandleClientFileRequest进行处理, 然后封装成CID_OTHER_FILE_TRANSFER_REQ向file_server请求创建文件传输事务;
3. file_server收到CID_OTHER_FILE_TRANSFER_REQ后使用_HandleMsgFileTransferReq函数进行处理, 主要处理 (在线和离线使用都使用同一函数, 这里先讲在线的情况):
 - a. 使用uuid创建task_id
 - b. 根据trans_mode, task_id, from_id, to_id, file_name, file_size调用NewTransferTask创建文件传输处理任务
 - c. 将result_code, from_user_id, to_user_id, file_name, file_size, task_id, trans_mode等信息封装到IMFileTransferReq使用CID_OTHER_FILE_TRANSFER_RSP回复msg_server
4. msg_server收到CID_OTHER_FILE_TRANSFER_RSP回复后使用_HandleFileMsgTransRsp进行响应, 主要做2个工作:
 - a. 使用CID_FILE_RESPONSE通知自己是否可以发送文件, 包含result_code, from_user_id, to_user_id, file_name, task_id, ip_addr_list (file_server的ip和port), trans_mode (传输模式在线或离线)。如果允许发送 客户端发送方调用需先连接file_server, 连接成功后客户端调用onConnectDone发送CID_FILE_LOGIN_REQ登录file_server的请求, 此时 file_role角色封装在对应的请求体IMFileLoginReq里面, 对发送者为CLIENT_REALTIME_SENDER。
 - b. 使用CID_FILE_NOTIFY通知对方有文件推送, 客户端接收方收到CID_FILE_NOTIFY后也需要先连接file_server, 连接成功后客户端调用onConnectDone发送CID_FILE_LOGIN_REQ登录file_server的请求, 此时 file_role角色封装在对应的请求体IMFileLoginReq里面, 对于接收方为CLIENT_REALTIME_RECVER。然后再使用CID_FILE_STATE命令决定是否同意接收文件后:
 - i. 如果拒绝则设置为CLIENT_FILE_REFUSE拒绝状态, 然后发送给服务器, 服务器收到命令后则使用_HandleClientFileStates进行处理, 并通知发送方具体的命令。
 - ii. 如果同意对方传输则发送 CID_FILE_PULL_DATA_REQ 请求拉取文件。服务器收到该命令后使用_HandleClientFilePullFileReq进行响应, 并转发给发送端让其开始发送数据。
5. 客户端发送方收到CID_FILE_PULL_DATA_REQ后调用_filePullDataReqResponse函数开始往file_server发送数据, 此时使用CID_FILE_PULL_DATA_RSP命令进行发送。
6. file_server收到CID_FILE_PULL_DATA_RSP后转发给客户端接收方, 调用_filePullDataRspResponse进行接收数据:
 - a. 接收数据未完成, 则继续调用CID_FILE_PULL_DATA_REQ
 - b. 接收数据完成, 则使用CID_FILE_STATE, 并设置状态为CLIENT_FILE_DONE。
 - c. 不管是否接收完, 服务器负责将接收方的回复信息转发给发送方。

总结: 通过msg_server获取file_server的地址, 并通知接收方有人要传输文件给他, 然后发送方和接收方都登录到file_server, 后续的状态和文件传输都转由file_server来处理, 不再需要msg_server。

- 在线文件传输没有做数据库的写入操作;
- 没有中间缓存, 服务器都是收到发送方的数据后原封不动转发给接收方;

1.2 离线文件传输

和在线传输的最大区别是：

- (1) file_server先以文件的方式缓存离线文件；
- (2) 离线推送请求需要以CID_FILE_ADD_OFFLINE_REQ的命令在，db_proxy_server调用DB_PROXY::addOfflineFile写入到文件；
- (3) 接收方上线后去服务器查询是否有离线文件CID_FILE_HAS_OFFLINE_REQ，db_proxy_server调用DB_PROXY::hasOfflineFile获取离线文件信息，最终也是要用CID_FILE_LOGIN_REQ登录file_server，进行文件文件的拉取。

具体流程作为作业。

2 对应的命令

CID_OTHER_FILE_SERVER_IP_REQ

向fileserver请求 file_server 对客户端的ip和端口

```
1 ▾ message IMFileServerIPReq{  
2     //cmd id: 0x0717  
3 }
```

Bash | 复制代码

CID_OTHER_FILE_SERVER_IP_RSP

file服务器回应地址

FileMsgServerConn

FileMsgServerConn::_HandleGetServerAddressReq

SQL | 复制代码

```

1  message IMFileServerIPRsp{
2      //cmd id:      0x0718
3      repeated IM.BaseDefine.IpAddr ip_addr_list = 1;
4  }

```

SQL | 复制代码

```

1  message IpAddr{
2      required string ip = 1;
3      required uint32 port = 2;
4  }

```

CID_FILE_REQUEST

发送方请求发送文件

IMFileReq

SQL | 复制代码

```

1  message IMFileReq{
2      //cmd id:      0x0506
3      required uint32 from_user_id = 1;
4      required uint32 to_user_id = 2;
5      required string file_name = 3;    // 文件名
6      required uint32 file_size = 4;    // 文件大小
7      required IM.BaseDefine.TransferFileType trans_mode = 5; // 1在线, 0离线
8  }

```

CID_OTHER_FILE_TRANSFER_REQ

pb/IM.Server.proto

msg_server向file_server请求创建传输任务，此时注意到结构体里面并没有task_id。

IMFileTransferReq

```

1  message IMFileTransferReq{
2      //cmd id:      0x0715
3      required uint32 from_user_id = 1;
4      required uint32 to_user_id = 2;
5      required string file_name = 3; // 文件名
6      required uint32 file_size = 4; // 文件大小
7      required IM.BaseDefine.TransferFileType trans_mode = 5; // 1在线, 0离线
8      optional bytes attach_data = 20;
9  }

```

CID_OTHER_FILE_TRANSFER_RSP

file_server回复msg_server，并相对请求时多携带了task_id

IMFileTransferRsp

```

1  message IMFileTransferRsp{
2      //cmd id:      0x0716
3      required uint32 result_code = 1;
4      required uint32 from_user_id = 2;
5      required uint32 to_user_id = 3;
6      optional string file_name = 4;
7      optional uint32 file_size = 5;
8      optional string task_id = 6;
9      optional IM.BaseDefine.TransferFileType trans_mode = 7;
10     optional bytes attach_data = 20;
11 }

```

CID_FILE_RESPONSE

发送方请求发送文件后，msg_server回应其请求，此时附带了task_id和对应file_server的ip和port地址。接下来发送方需要登录file_server。

IMFileRsp

```

1  message IMFileRsp{
2      //cmd id: 0x0507
3      required uint32 result_code = 1;          //1: 失败 0:成功
4      required uint32 from_user_id = 2;
5      required uint32 to_user_id = 3;
6      required string file_name = 4;
7      required string task_id = 5;
8      repeated IM.BaseDefine.IpAddr ip_addr_list = 6;
9      required IM.BaseDefine.TransferFileType trans_mode = 7; // 1在线, 0离线
10 }

```

CID_FILE_NOTIFY

在线文件传输时，msg_server将请求在线文件传输发送给接收方，里面包含了对应的file_server的ip和port地址，以及其他的必要信息。接收方收到后也是要登录file_server

IMFileNotify

```

1  message IMFileNotify{
2      //cmd id: 0x0508
3      required uint32 from_user_id = 1;
4      required uint32 to_user_id = 2;
5      required string file_name = 3;
6      required uint32 file_size = 4;
7      required string task_id = 5;
8      repeated IM.BaseDefine.IpAddr ip_addr_list = 6;
9      required IM.BaseDefine.TransferFileType trans_mode = 7;
10     required uint32 offline_ready = 8;          //1:True 0:False
11 }

```

CID_FILE_LOGIN_REQ

真正开始传输文件时先要登录file_server，使用user和task_id，并注明自己的角色

IMFileLoginReq


```

1  message IMFileLoginReq{
2      //cmd id: 0x0501
3      required uint32 user_id = 1;
4      required string task_id = 2;
5      required IM.BaseDefine.ClientFileRole file_role= 3; //1在线发送者,2在线接收者,3离线上传,4离线下载
6  }

```

ClientFileRole

```

1  enum ClientFileRole{
2      CLIENT_REALTIME_SENDER = 0x01; //在线发送者
3      CLIENT_REALTIME_RECVER = 0x02; //在线接收者
4      CLIENT_OFFLINE_UPLOAD = 0x03; //离线上传
5      CLIENT_OFFLINE_DOWNLOAD = 0x04; //离线下载
6  }

```

CID_FILE_LOGIN_RES

回应客户端登录情况

IMFileLoginRsp

```

1  message IMFileLoginRsp{
2      //cmd id: 0x0502
3      required uint32 result_code = 1; //0:succesced 1:failed
4      required string task_id = 2;
5  }

```

CID_FILE_STATE

服务器向客户端请求在线文件发送

IMFileState

```

1  message IMFileState{
2      //cmd id: 0x0503
3      required IM.BaseDefine.ClientFileState state = 1;
4      required string task_id = 2;
5      required uint32 user_id = 3;
6  }

```

ClientFileState

```

1  enum ClientFileState{
2      CLIENT_FILE_PEER_READY = 0x00;
3      CLIENT_FILE_CANCEL     = 0x01;
4      CLIENT_FILE_REFUSE     = 0x02;
5      CLIENT_FILE_DONE       = 0x03;
6  }

```

CID_FILE_PULL_DATA_REQ

CID_FILE_PULL_DATA_REQ 服务器向客户端请求上传离线数据

IMFilePullDataReq

```

1  message IMFilePullDataReq{
2      //cmd id: 0x0504
3      required string task_id = 1;
4      required uint32 user_id = 2;
5      required IM.BaseDefine.TransferFileType trans_mode = 3;
6      required uint32 offset = 4;
7      required uint32 data_size = 5;
8  }

```

CID_FILE_PULL_DATA_RSP

客户端向服务器传输数据

IMFilePullDataRsp

```

1  message IMFilePullDataRsp{
2      //cmd id: 0x0505
3      required uint32 result_code = 1;
4      required string task_id = 2;
5      required uint32 user_id = 3;
6      required uint32 offset = 4; // 偏移位置
7      required bytes file_data = 5; // 文件数据
8  }

```

DB_PROXY::hasOfflineFile是否有离线文件

请求命令: CID_FILE_HAS_OFFLINE_REQ

回复命令: CID_FILE_HAS_OFFLINE_RES

1. 请求结构

IMFileHasOfflineReq

```

1  message IMFileHasOfflineReq{
2      //cmd id: 0x0509
3      required uint32 user_id = 1;
4      optional bytes attach_data = 20;
5  }

```

2. 回应结构

IMFileHasOfflineRsp

```

1  message IMFileHasOfflineRsp{
2      //cmd id: 0x050a
3      required uint32 user_id = 1;
4      repeated IM.BaseDefine.OfflineFileInfo offline_file_list = 2;
5      repeated IM.BaseDefine.IpAddr ip_addr_list = 3;
6      optional bytes attach_data = 20;
7  }

```

OfflineFileInfo相应的信息存储到MySQL数据库

```

1  message OfflineFileInfo{
2      required uint32 from_user_id = 1;
3      required string task_id = 2;
4      required string file_name = 3;
5      required uint32 file_size = 4;
6  }

```

IpAddr

```

1  message IpAddr{
2      required string ip = 1;
3      required uint32 port = 2;
4  }

```

3. 处理逻辑

- DB_PROXY::hasOfflineFile
- 调用CFileModel::getOfflineFile(uint32_t userId, list<IM::BaseDefine::OfflineFileInfo>& lsOffline)获取离线文件信息，离线文件可以有多个。实质是查询"select * from IMTransmitFile where **toId**="+int2string(userId) + " and status=0 order by created";通过**toId** 查询是否属于自己的离线文件。
- 将结果封装到IMFileHasOfflineRsp回复请求者

DB_PROXY::addOfflineFile 加入离线文件

请求命令：CID_FILE_ADD_OFFLINE_REQ

回复命令：无

1. 请求结构

IMFileAddOfflineReq

```

1  message IMFileAddOfflineReq{
2      //cmd id: 0x050b
3      required uint32 from_user_id = 1;
4      required uint32 to_user_id = 2;
5      required string task_id = 3;
6      required string file_name = 4;
7      required uint32 file_size = 5;
8  }

```

2. 回应结构

无

3. 处理逻辑

- DB_PROXY::addOfflineFile
- 具体调用CFileModel::addOfflineFile(uint32_t fromId, uint32_t told, string& taskId, string& fileName, uint32_t fileSize), 使用"insert into IMTransmitFile (`fromId`,`told`,`fileName`,`size`,`taskId`,`status`,`created`,`updated`) values(?,?,?,?,?,?,?,?);"

DB_PROXY::delOfflineFile 删除离线文件

请求命令: CID_FILE_DEL_OFFLINE_REQ

回复命令: 无

1. 请求结构

IMFileDelOfflineReq

```

1  message IMFileDelOfflineReq{
2      //cmd id: 0x050c
3      required uint32 from_user_id = 1;
4      required uint32 to_user_id = 2;
5      required string task_id = 3;
6  }

```

2. 回应结构

3. 处理逻辑

- DB_PROXY::delOfflineFile入口
- 具体调用CFileModel::delOfflineFile(uint32_t fromId, uint32_t told, string& taskId), 使用"delete from IMTransmitFile where fromId=" + int2string(fromId) + " and told="+int2string(told) + " and taskId='" + taskId + "'";

