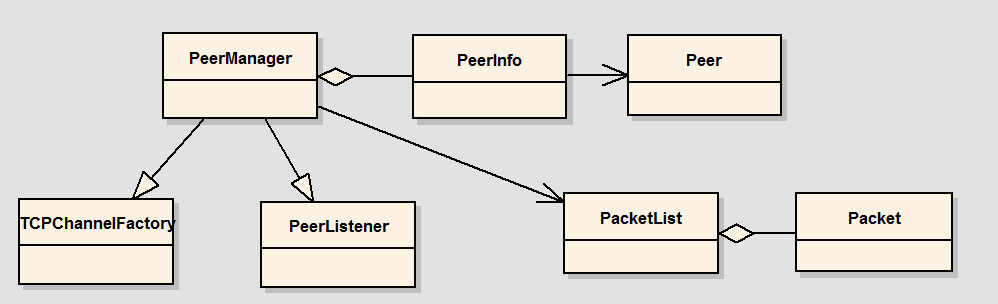
Tracker主要负责peer上报数据的管理以及配合完成peer的调度，主要由PeerManager、UserManager和SourceService模块完成

# 1 PeerManager模块



tracker暂时定义PEER\_MAX\_SIZE 200000个用户，

## 1.1主要成员变量

BaseArray<PeerInfo> m\_peers；//peer列表

PacketList m\_packs://接收到的数据包；

## 1.2主要成员函数

init：新建PEER\_MAX\_SIZE个用户peer，每个peer注册，这样peer中能回调PeerManger中的各种on，建立接收peer连接的TCPserver端；

handle\_root：处理收发线程，调用IOReactor也即是ProReactor的handle\_root，处理发送列表m\_packs中数据，进行发送handle\_send；处理handle\_timer，实际是两分钟查一次，清除三分钟不活动的peer；

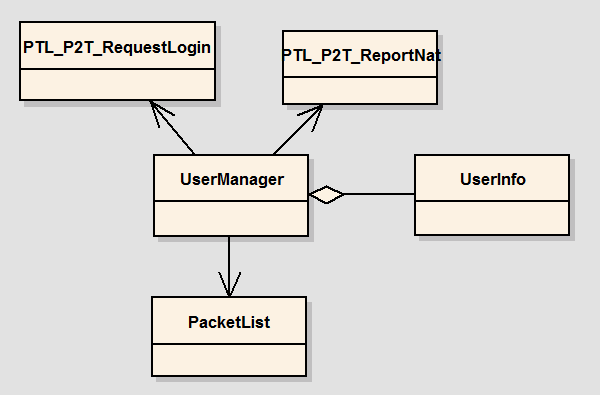
handle\_send：命令是PACK\_CMD\_DATA调用相应的peer（索引是packet中的sid）发送；

get\_peer：从peer列表分配没使用的peer；

on(PeerListener::Connected）：peer连接上会调用的回调；

on(PeerListener::Data ,Peer\* peer,MemBlock \*block）：接收到peer的数据会调用的回调，会把数据交给UserManager去处理；

# 2 UserManger模块



## 2.1主要成员变量

UserInfo \*m\_userl;

PacketList m\_packs;

int m\_nat\_type\_num[7]；//nat类型计数

## 2.2主要成员函数

init：新建PEER\_MAX\_SIZE个UserInfo，与PeerManger中peer数对应，初始化SourceService；

handle\_root：命令处理包线程，从包队列中取包，解析handle\_command；

handle\_command：根据包的命令的不同进行不同的处理，包的命令见协议；

get\_nat\_type：获得各种nat统计类型的计数值；

get\_server\_list；获取类型为UT\_SERVER，UT\_CENTER，UT\_SUPER用户列表；

handle\_timer：定时处理任务：

1）每两分钟重新统计各种nat统计计数值；

2）三分钟不活动的用户做logout处理；

3）连续四次空列表（共享文件列表为空）的用户做logout处理（共享文件列表为空）；

4）10分钟统计一次StatManager超时处理；

5）每30s处理一次SourceService的超时处理；

on\_ptl\_request\_login：处理登陆请求，根据每个用户的mac生成个uid，保存peer的信息，向DBManager传送包（应该会有写入数据库的操作），把peer在tracker中的sessionID以及公网ip、trackerID等信息反馈给用户，把发送包推给PeerManager处理；

on\_ptl\_report\_nat：保存peer上传nat的信息；

on\_ptl\_report\_downloadlist\_maxnum：保存peer上传的最大下载列表项的数据；

on\_ptl\_report\_sharefile：将peer上传的共享文件信息保存到SourceService中；

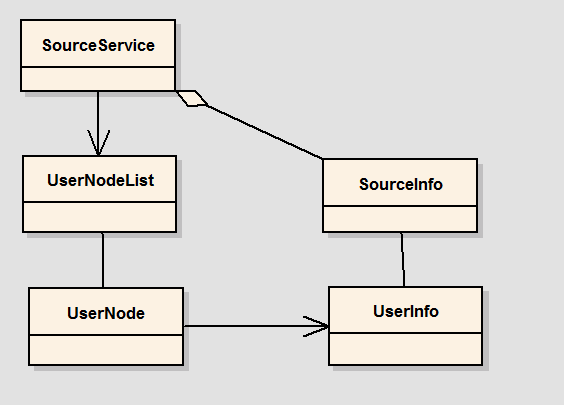
on\_ptl\_report\_removefile：将peer删除的共享文件从SourceService中删除；

on\_ptl\_report\_startdownloadfile：统计信息；

on\_ptl\_report\_startdownloadlist：统计信息，vip的用户调用AutoFileManager的on\_vipfile\_start，可能会分些比较的好源；

on\_ptl\_request\_filesource：处理peer请求文件源的需求（最多返回60个源）；

# 3. SourceService模块



管理各用户的文件源信息，方便用户调度，每一个UserInfo对应是一个用户，其中保存的有一个用户共享的资源。信息保存两层，一是以文件名hash名保存在SourceMap，每个second成员保存的是SourcInfo；二是每个SourceInfo中保存了user、center、server列表的map，每个map是以ip为索引，保存的是UserInfo信息。

UserNodeList：把用户的信息串成链表，节点信息存的是UserInfo；

SourceInfo：在SourceService中以hash会上存在map中，存了很多用户表，包括server，center和super，以及存储UserMap，map是multimap（很多用户可能共用同一个公网ip），以ip为主键，存的的是UserInfo信息；

## 3.1主要成员变量

SourceMap m\_sourcel; //用户列表的map，以ip作key

UserNodeList m\_free\_peernodel; //free列表

## 3.2主要成员函数

source\_add：source添加，把信息添加到m\_sourcel中；

source\_find：每搜到一个源,就将它移到最后,总是从前往后找,保持最均分配，返回已经分享并且能连接的用户信息；

get\_msg\_file\_info：获取指定file的信息，能查到所有正在分享该文件的用户；

## 3.3重要数据结构

1）UserInfo

用户信息统计，记录用户的各种信息，有IP、port、上传下载量、mac、nat类型等

map<hash\_t,uint32> sourceMap：记录该用户所有的文件的hash

2）SourceInfo

某一个文件源的信息记录

uint32 server\_num; //

uint32 center\_num; //

uint32 super\_num; //

uint32 urlflag\_num; //记录允许http连接的个数

uint32 vip\_num; //VIP源计数，0表示不是VIP节目，非0表示VIP节目

rlist\_head\_t server\_head; //peer\_type\_server列

rlist\_head\_t center\_head; //中心服务器源，后备源

rlist\_head\_t super\_head;

typedef std::multimap<uint32,UserInfo\*> UserMap

UserMap user\_map; //用户列表map，以ip为索引，记录是所有拥有此源用户；

# 4. AutoFileManager模块

## 4.1重要成员变量

typedef list<UserInfo\*> UserList;

typedef list<SourceInfo\*> SourceList;

typedef map<hash\_t,string> VipfileList;

UserList m\_supers; //超级用户列表

SourceList m\_vipsources; //VIP节目源

VipfileList m\_vipfiles; //vip文件列表

## 4.2重要成员函数

add\_super/del\_super：添加/移除超级用户；

add\_vipsource/del\_vipsource：添加/移除vip节目；

on\_vipfile\_start：成为VIP节目条件，第一个VIP用户开始下载此文件开始就为VIP节目，调用handle\_super\_source处理；

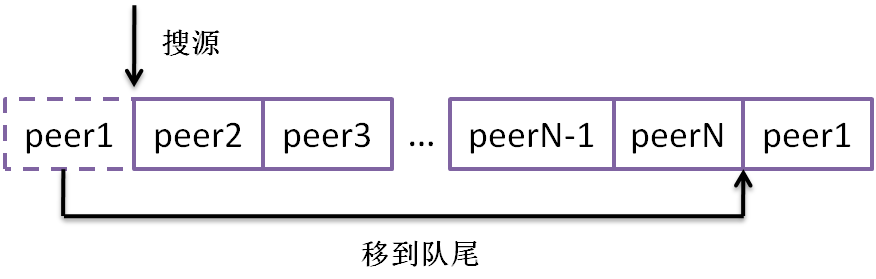
on\_vipfile\_stop：停止条件:不再有共享的vip源，则SoureService在delete是触发；

handle\_super\_source；

# 5.策略/算法

## 5.1取源策略

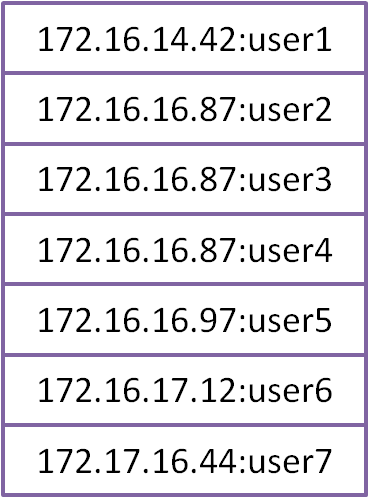
super/server/center端请求tracker server请求文件源，请求的是分享该文件的用户，tracker server端采用顺序的方式，从前向向取，搜到一个源就放到最后，这样保证取源的均匀。



peer端（client）端请求源时策略，取相同ip和相同ip之后的peer，举例如下：

1）172.16.16.87用户请求分配源，则先定位到user5，同时向后和向前查找，向前从user4开始，向后从user5开始，依次会取得的结果是user4，user5，user3，user6...；

2）172.16.16.89用户请求分配源，会分配user4，user5，user3，user6...



# 6.LOG系统

## 6.1 Log系统概要

Log系统的主要由LogManager生成，记录log系统的触发主要是在DBManager完成的，主要由DBC\_USER\_LOGIN、DBC\_USER\_LOGIN\_OUT、DBC\_START\_FILE、DBC\_DOWNLOADFILE\_INFO、DBC\_DOWNLOAD\_INFO、DBC\_REALTIME\_INFO；

其中：

1）DBC\_REALTIME\_INFO和DBC\_DOWNLOAD\_INFO是由StatManager中定时产生的；

2）DBC\_USER\_LOGIN是在收到peer的request\_login时放入DBManager的；

3）DBC\_START\_FILE是在StatManager的on\_req\_file中放入DBManager，在UserManager的收到上报开始下载文件和开始下载文件列表时都会触发StatManager的on\_req\_file；

4）DBC\_DOWNLOADFILE\_INFO；在UserManager的on\_ptl\_report\_downloadfilespeed和on\_ptl\_report\_downloadfileinfo中放入DBManager的；

## 6.2 LOG系统内容

### 6.2.1 LOGIN管理

peer登陆到tracke会记录在login文件中，文件命名"login\_year\_month\_day.log"，例如“login\_2016.06.29.log”，格式如下：

[%s]: uid, user\_type, prog\_ver, os\_ver, ip

//时间，用户uid，用户类型，程序的版本，os的版本，ip地址

uid是tracker为用户根据用户mac，id等信息采用编码而成的，格式类似"vttv-%d-%s-%d"，依次是连接的trackerid，mac的字码，随机值；

user\_type代表用户的类型，有以下：

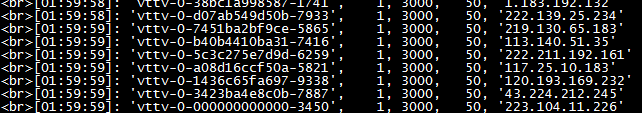
UT\_CLIENT=1//客户端

,UT\_SERVER=2 //一般的super node

,UT\_HTTP=3 //这个用不到

,UT\_CENTER=4 //目前预定义指分发源，或者直播源

,UT\_SUPER=5



### 6.2.2 LOGINOUT管理

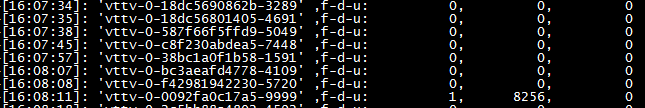
peer退出时，tracker记录的log，文件命名"loginout\_year\_month\_day.log"，例如“loginout\_2016.06.29.log”，格式如下

[%s]: uid, file\_num, download\_flow, upload\_flow

file\_num：请求的文件数

download\_flow：下载量MB；

upload\_flow：上传量MB；



记录的是该用户的文件数、所有下载和上传的数据量。

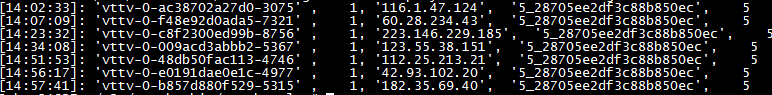
### 6.2.3 STARTFILE管理

记录是播放文件列表或文件开始的log, 文件命名" start\_file\_year\_month\_day.log"或" start\_playlist\_year\_month\_day.log"，例如“start\_file\_2016.06.29.log”或“start\_playlist\_2016.06.29.log”，格式如下

[%s]:uid,user\_type,ip,str\_hash,hash\_type

str\_hash：文件的hash

hash\_type：hash的类型，HT\_URL2用户文件，HT\_URLDL用于播放列表



记录的用户开始上报开始下载列表或开始下载文件，实际中只有列表log在更新。

### 6.2.4 DOWNLOADFILE管理

各peer的下载信息，

[%s]:flag, uid, ip, str\_hash, "

" hash\_type(filesize, speed\_KB, downseconds, cacheTimes, dragTimes, cacheSeconds), "

" c-s/f(TCP\_TURN,TCP\_TURN,UDP\_CONN,UDP\_TURN,UDP\_THRU),"

" s-KB(tcp, udp),"

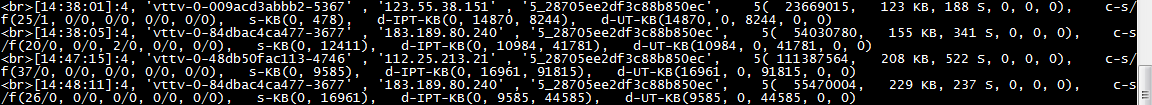
" d-IPT-KB(tcp, udp,http),"

" d-UT-KB(client,server,http,center,super)

s-KB：分享多少数据

d-IPT-KB：下载了多少数据；

d-UT-KB：下载了多少数据；



数据应该是累积的！！！！

### 6.2.5 DOWNLOAD管理

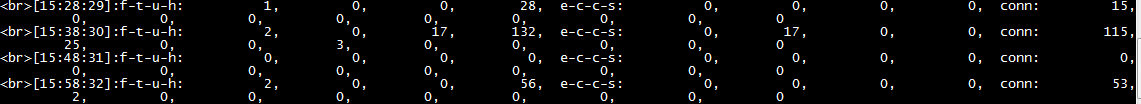
下载信息记录log，文件命名" downinfo\_year\_month\_day.log"，例如“downinfo \_2016.06.29.log”，格式如下

[%s]: 1-request\_files, 2-tcp\_download\_flow, 3-udp\_download\_flow, 4-http\_download\_flow,"

" 5-edge\_download\_flow, 6-client\_download\_flow, 7-center\_download\_flow, 8-super\_download\_flow,"

" 9-tcp\_connect\_success, 10-tcp\_connect\_fail, 11-tcp\_accept\_success, 12-tcp\_accept\_fail, 13-udp\_connect\_success, 14-udp\_connect\_fail,"

" 15-udp\_receive\_success, 16-udp\_receive\_fail, 17-udp\_nat\_success, 18-udp\_nat\_fail



这些信息是10min更新一次，每次都reset。

### 6.2.6 REALTIME管理

tracker根据peer上报的信息记录各peer数量及nat类型统计，文件命名" realtimeinfo\_year\_month\_day.log"，例如“realtimeinfo \_2016.06.29.log”，格式如下

[%s]: online\_users, edge\_servers, center\_servers, super\_servers,"

" online\_progs, online\_sources,"

" nat\_0, nat\_1, nat\_2, nat\_3, nat\_4, nat\_5

