# 直播portal的入口

|  |  |  |
| --- | --- | --- |
| 接口 | 请求参数 | 返回内容 |
| flv直播流调度接口  /get\_pi? | GET /get\_pi?mid=<直播播放流id> | 302跳转到直播流的请求地址 |
| HLS直播流调度接口  /get\_mpi? | GET /get\_mpi?mid=<直播播放流id> | 302跳转到直播流的请求地址 |
| 直播流创建接口  /create\_stream? | POST /create\_stream?  user\_id=<用户id>&  room\_id=<房间id>&  user\_level=<用户级别，包括普通用户，主播或管理员等>;  需要post上来数据包括互动能力的声明（包括音视频的能力,上传协议,编码能力,编码格式,码率空间）,以及房间属性(包括房间大小等);使用JSON格式 | 直播采集流id,  直播流上传接口的URL，  对应的直播播放流列表  （包括直播播放流id，直播流调度接口的URL，直播流的用户数查询接口URL，  直播流的媒体信息），采集流的采集参数（包括音视频的格式,编码格式,码率等）等信息；JSON格式  5月1日制定格式的细节 |
| 直播流销毁接口  /destroy\_stream? | POST /destroy\_stream?  user\_id=<用户id>&  stream\_id=<直播采集流id> |  |
| 直播流状态查询接口  /get\_all\_stream? | GET /get\_all\_stream?  room\_id=<房间id> | 直播流列表，包括  直播采集流id,  直播流上传接口的URL，  对应的直播播放流列表  （包括直播播放流id，直播流调度接口的URL，直播流的用户数查询接口URL，  直播流的媒体信息），采集流的采集参数（包括音视频的格式,编码格式,码率等）等信息；  JSON格式 |
| 直播流的用户数查询接口  /get\_stream\_online? | GET /get\_stream\_online?stream\_id=<直播播放流id> | 该直播播放流在各个播放平台上的用户在线数 |
|  |  |  |

response

{

"version": "1.0",

“error\_code”:1,

}

创建流

request

POST /create\_stream?user\_id=<用户id>&room\_id=<房间id>

response

{

"version": "1.0",

“error\_code”:0

"cs\_id": <capture\_stream\_id>,

"cs\_para":"<some text data for encoder>",

"upload\_url":"<url>",

"ps\_list":

[

{

"ps\_id":<play\_stream\_id>,

"ps\_info":"<some text data for this ps>",

"ps\_url":

[

{

"type": "flv",

"url": "<url>"

},

{

"type": "ts",

"url": "<url>"

}

]

}

]

}

销毁流

request

POST /destroy\_stream?user\_id=<用户id>&stream\_id=<直播采集流id>

response

{

"version": "1.0",

"result": 0,

}

获取房间的流列表

request

GET /get\_all\_stream?room\_id=<房间id>

Response

{

"version": "1.0",

“error\_code”:0

" stream\_list" :[

{

"cs\_id": <capture\_stream\_id>,

"cs\_para":"<some text data for encoder>",

"user\_id":"<user\_id>"

"upload\_url":"<url>",

"ps\_list":

[

{

"ps\_id":<play\_stream\_id>,

"ps\_info":"<some text data for this ps>",

"ps\_url":

[

{

"type": "flv",

"url": "<url>"

},

{

"type": "ts",

"url": "<url>"

}

]

}

]

}

]

}

# CDN调度和分发



|  |  |  |
| --- | --- | --- |
| 术语 | 说明 | 备注 |
| 分发服务器 | 负责把收集到视频流，推送到各个机房的流媒体服务器 | 机群 |
| L0流媒体服务器 | 使用指定的协议（P2P）向流媒体服务器分发视频流 |  |
| 流媒体服务器 | 负责传送流媒体数据给用户 |  |
| 直播portal | 负责控制流媒体服务器,收集流媒体服务器上的数据, 互动行为的发起和集成调度功能 |  |
| Tracker | 跟踪播放流在各个P2P节点的分布情况 |  |
|  |  |  |

**播放**调度算法需要解决的问题

1. 降低流分发成本；
2. 机器过载保护；

调度算法，需要各个服务器定时上报负载情况。算法如下：

所有机器划分为Group，一个group含有多台来自不同机房或地域的机器；

当一个用户要访问某个流时，调度算法执行如下

* + 1. 在n个Group里面，以流id作为随机种子；
    2. 使用随机数生成算法查找到一个Group；
    3. Group里面有k台机器，根据就近原则，获取h台机器；
    4. 如果是热门的流(用户在线数超过1000)，则从h台机器里面过滤掉负载超过阀值的机器；当有剩余的机器时，使用轮询算法，返回一台机器；
    5. 如果非热门的流，则使用与查找Group类似的算法来查找一个负载不超过阀值的机器，返回该机器；
    6. 如果未找到合适的机器，移除该Group，n=n-1，重复步骤2；
    7. 当n==0时，返回不可服务；（表示需要扩容了）

需要完善的点

热门流的预测，提前分发流

基于业务场景的优化

流媒体服务器分发原则

延迟分发，用户请求传递到某一台流媒体服务器时，如果该流媒体服务器上没有该流，则该流媒体服务器根据直播portal的控制，使用指定的协议同步直播流。

延迟回收，当流媒体服务器上的某个流没有用户访问，可标示为空闲状态；当持续空闲的时间较长时，流媒体服务器可以销毁当前流，停止流同步。

遗留问题

1. 直播流的用户数减少时如何缩小分发服务器规模；
2. 预分发

服务能力估算(独占机器)

直播流的码流为400Kbps，单台服务器（1Gbps）可以同时服务2500个用户；

40台流媒体服务器，可以服务10万用户；

每个group有5台，可以分为8个group，100个房间；

单个group，

可以服务13个房间，每个房间的平均在线为1000人，

分发成本为13\*5\*400Kbps=26Mbps

400台流媒体服务器，可以服务100万用户；

每个group有10台，可以分为40个group，200个房间；

单个group，

可以服务5个房间，每个房间的平均在线为5000人，

分发成本为5\*10\*400Kbps=20Mbps

400台流媒体服务器，可以服务100万用户；

每个group有20台，可以分为20个group，20个房间；

单个group，

可以服务1个房间，每个房间的平均在线为50000人，

分发成本为5\*20\*400Kbps=40Mbps

分块流广播树

分块标准：采用变长分块，分块的长度以时间为单位以FLV的tag为边界，大小不超过100毫秒；以flv tag的时间戳作为划分的标准;每一块打上序列号,序列号的起始值是当前的系统时间戳，后续的块采用起始值的累加作为序号。

共享算法：采用推模式，分为发布者和订阅者；Tracker跟踪各个发布者；Peer启动P2P时，首先向Tracker查询发布者；随后，Peer向发布者订阅分块流；订阅者需要回复订阅成功或失败。Peer跟踪分块的接收时长，超时发生时，Peer会退订，然后向其它订阅者订阅。

分块算法：如果超过1秒以上没有生成一个分块,则需要推送一个空的分块,以避免各个订阅者以为发布者超时了。

需要处理的问题：

1. 避免循环订阅。

每次只返回一个上级, Tracker统一管理上下级关系，来避免循环

直播服务器坏了，如何通知业务服务器