

《分布式流媒体》实训项目

C/C++教学体系

TNV DAY15

直播课

目录

建表脚本(storage.sql)

安装/卸载脚本(install.sh/uninstall.sh)

安装测试

建表脚本(storage.sql)



建表脚本(storage.sql)

```
USE tnv_storagedb
DROP TABLE IF EXISTS `t_file_01`;
CREATE TABLE `t_file_01` (
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',
  `appid` varchar(32) DEFAULT NULL,
  `userid` varchar(128) DEFAULT NULL,
  `status` tinyint(4) DEFAULT NULL,
  `file_path` varchar(512) DEFAULT NULL,
  `file_size` bigint(20) 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;
```



建表脚本(storage.sql)

```
DROP TABLE IF EXISTS `t_file_02`;  
CREATE TABLE `t_file_02` (  
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `status` tinyint(4) DEFAULT NULL,  
  `file_path` varchar(512) DEFAULT NULL,  
  `file_size` bigint(20) 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;
```



建表脚本(storage.sql)

```
DROP TABLE IF EXISTS `t_file_03`;
CREATE TABLE `t_file_03` (
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',
  `appid` varchar(32) DEFAULT NULL,
  `userid` varchar(128) DEFAULT NULL,
  `status` tinyint(4) DEFAULT NULL,
  `file_path` varchar(512) DEFAULT NULL,
  `file_size` bigint(20) 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;
```

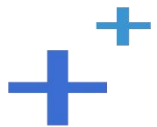


安装/卸载脚本 (install.sh/uninstall.sh)



安装脚本(install.sh)

```
#!/bin/bash
cd 02_tracker
make clean && make all
cd ../03_id
make clean && make all
cd ../04_storage
make clean && make all
cd ../05_client
make clean && make all
cd ../06_http
make clean && make all
exit 0
```



卸载脚本(uninstall.sh)

```
#!/bin/bash
cd 02_tracker
make clean
cd ../03_id
make clean
cd ../04_storage
make clean
cd ../05_client
make clean
cd ../06_http
make clean
cd ../../bin
rm -rf *
exit 0
```



安装测试



创建数据库

- 分别为跟踪服务器、ID服务器和存储服务器创建数据库tnv_trackerdb、tnv_idsdb和tnv_storagedb，并执行sql目录下的建表脚本，初始化表结构
 - `$ cd ~/Projects/TNV/sql`
`$ mysql -uroot -p123456`
 - `mysql> source tracker.sql`
`mysql> source id.sql`
`mysql> source storage.sql`
`mysql> exit`



编译安装应用

- 直接执行src目录下的安装脚本install，完成整个项目的编译、链接和安装
 - \$ cd ~/Projects/TNV/src
 - \$./install

```
#!/bin/bash
```

```
cd 02_tracker  
make clean && make all  
cd ../03_id  
make clean && make all  
cd ../04_storage  
make clean && make all  
cd ../05_client  
make clean && make all  
cd ../06_http  
make clean && make all  
  
exit 0
```



启动跟踪服务器

- 直接执行bin目录下的tracker可执行程序
 - \$ cd ~/Projects/TNV/bin
 - \$./tracker

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./tracker
master_log_open(24): no MASTER_LOG's env value
src/master/template/acl_threads_server.c(1377)->acl_threads_server_main: configure file = ../etc/tracker.cfg
src/master/template/acl_threads_server.c(891)->server_init: can't get SERVICE_LOG's env value, use acl_master.log log
event_limit: max fdsize: 102400
src/event/events_epoll_thr.c(599), event_epoll_alloc_thr, use thread events - epoll
inet_listen: listen 127.0.0.1:21000 ok
log_event_mode(919): use kernel_event
12_server.cpp(64), proc_on_init: cfg_appids: tnvideo, cfg_maddrs: 127.0.0.1, cfg_raddrs: 127.0.0.1:6379, cfg_interval: 120, cfg_mtimeout: 30, cfg_maxconns: 600, cfg_ctimeout: 10, cfg_rtimeout: 10, cfg_ktimeout: 60
acl_threads_server_main(1586), /home/tarena/Projects/TNV.20201111/bin/tracker daemon started, log: acl_master.log
```



启动ID服务器

- 直接执行bin目录下的id可执行程序
 - \$ cd ~/Projects/TNV/bin
 - \$./id

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./id
master_log_open(24): no MASTER_LOG's env value
src/master/template/acl_threads_server.c(1377)->acl_threads_server_main: configure file = ../etc/id.cfg
src/master/template/acl_threads_server.c(891)->server_init: can't get SERVICE_LOG's env value, use acl_master.log log
event_limit: max fdsize: 102400
src/event/events_epoll_thr.c(599), event_epoll_alloc_thr, use thread events - epoll
inet_listen: listen 127.0.0.1:22000 ok
log_event_mode(919): use kernel_event
08_server.cpp(32), proc_on_init: cfg_maddrs: 127.0.0.1, cfg_mtimeout: 30, cfg_maxoffset: 100
acl_threads_server_main(1586), /home/tarena/Projects/TNV.20201111/bin/id daemon started, log: acl_master.log
```



启动存储服务器

- 直接执行bin目录下的storage可执行程序
 - \$ cd ~/Projects/TNV/bin
 - \$./storage

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./storage
master_log_open(24): no MASTER_LOG's env value
src/master/template/acl_threads_server.c(1377)->acl_threads_server_main: configure file = ../etc/storage.cfg
src/master/template/acl_threads_server.c(891)->server_init: can't get SERVICE_LOG's env value, use acl_master.log log
event_limit: max fdsize: 102400
src/event/events_epoll_thr.c(599), event_epoll_alloc_thr, use thread events - epoll
inet_listen: listen 127.0.0.1:23000 ok
log_event_mode(919): use kernel_event
16_server.cpp(97), proc_on_init: cfg_gpname: group001, cfg_spaths: ../data, cfg_taddrs: 127.0.0.1:21000, cfg_iaddrs: 127.0.0.1:22000, cfg_maddrs: 127.0.0.1, cfg_raddrs: 127.0.0.1:6379, cfg_bindport: 23000, cfg_interval: 10, cfg_mtimeout: 30, cfg_maxconns: 600, cfg_ctimeout: 10, cfg_rtimeout: 10, cfg_ktimeout: 60
acl_threads_server_main(1586), /home/tarena/Projects/TNV.20201111/bin/storage daemon started, log: acl_master.log
14_tracker.cpp(101), join: bodylen: 0, command: 100, status: 0
14_tracker.cpp(178), beat: bodylen: 0, command: 100, status: 0
14_tracker.cpp(178), beat: bodylen: 0, command: 100, status: 0
14_tracker.cpp(178), beat: bodylen: 0, command: 100, status: 0
14_tracker.cpp(178), beat: bodylen: 0, command: 100, status: 0
```



启动HTTP服务器

- 直接执行bin目录下的http可执行程序
 - \$ cd ~/Projects/TNV/bin
 - \$./http

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./http
master_log_open(24): no MASTER_LOG's env value
src/master/template/acl_threads_server.c(1377)->acl_threads_server_main: configu
re file = ../etc/http.cfg
src/master/template/acl_threads_server.c(891)->server_init: can't get SERVICE_LO
G's env value, use acl_master.log log
event_limit: max fdsize: 102400
src/event/events_epoll_thr.c(599), event_epoll_alloc_thr, use thread events - ep
oll
inet_listen: listen 127.0.0.1:8080 ok
log_event_mode(919): use kernel_event
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, ma
x connect: 16
connect_manager.cpp(122), init: no default connection set
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:6379, max
connect: 250
connect_manager.cpp(122), init: no default connection set
07_server.cpp(24), proc_on_init: cfg_taddrs: 127.0.0.1:21000, cfg_raddrs: 127.0.
0.1:6379, cfg_maxthrs: 250, cfg_ctimeout: 10, cfg_rtimeout: 10, cfg_rsession: 0
acl_threads_server_main(1586), /home/tarena/Projects/TNV.20201111/bin/http daemo
n started, log: acl_master.log
```



测试从跟踪服务器获取组列表

- 执行如下命令行
 - \$ cd ~/Projects/TNV/bin
 - \$./client 127.0.0.1:21000 groups
- 宕下并重启存储服务器，观察组列表的状态变化

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./client 127.0.0.1:21000 groups
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, max connect: 16
connect_manager.cpp(122), init: no default connection set
02_conn.cpp(612), recvhead: bodylen: 407, command: 100, status: 0
COUNT OF GROUPS: 1
GROUPNAME: group001
COUNT OF STORAGES: 1
COUNT OF ACTIVE STORAGES: 1
VERSION: 1.0
HOSTNAME: TNV
ADDRESS: 127.0.0.1:23000
STARTUP TIME: Sun Nov 15 13:49:58 2020
JOIN TIME: Sun Nov 15 13:49:58 2020
BEAT TIME: Sun Nov 15 13:49:58 2020
STATUS: ACTIVE
tarena@TNV:bin$
```



测试向存储服务器上传文件

- 执行如下命令行
 - `$ cd ~/Projects/TNV/bin`
 - `$ ./client 127.0.0.1:21000 upload tnvideo tnv001`
 - `../upload/001.mp4`
- 检查存储服务器的存储目录和数据库
- 通过diff命令比对文件内容

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./client 127.0.0.1:21000 upload tnvideo tnv001 ../upload/001.mp4
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, ma
x connect: 16
connect_manager.cpp(122), init: no default connection set
02_conn.cpp(612), recvhead: bodylen: 34, command: 100, status: 0
02_conn.cpp(612), recvhead: bodylen: 0, command: 102, status: 0
Upload success: 5fb0c3ed088d19807e46b570003fc4000177
tarena@TNV:bin$
```



测试向存储服务询问文件大小

- 执行如下命令行
 - \$ cd ~/Projects/TNV/bin
 - \$./client 127.0.0.1:21000 filesize tnvideo tnv001 5fb0...0177
- 检查存储服务器的存储目录、数据库和客户机文件系统中的文件大小是否完全一致

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./client 127.0.0.1:21000 filesize tnvideo tnv001 5fb0c3ed088d198
07e46b570003fc4000177
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, ma
x connect: 16
connect_manager.cpp(122), init: no default connection set
02_conn.cpp(612), recvhead: bodylen: 34, command: 100, status: 0
02_conn.cpp(612), recvhead: bodylen: 8, command: 102, status: 0
Get filesize success: 734529655
tarena@TNV:bin$
```



测试从存储服务器下载文件

- 执行如下命令行
 - \$ cd ~/Projects/TNV/bin
 - \$./client 127.0.0.1:21000 download tnvideo tnv001 5fb0...0177 0 0
- 通过diff命令比对文件内容
- 下载文件中的部分内容

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./client 127.0.0.1:21000 download tnvideo tnv001 5fb0c3ed088d198
07e46b570003fc4000177 0 0
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, ma
x connect: 16
connect_manager.cpp(122), init: no default connection set
02_conn.cpp(612), recvhead: bodylen: 34, command: 100, status: 0
02_conn.cpp(612), recvhead: bodylen: 734529655, command: 102, status: 0
Download success: 734529655
tarena@TNV:bin$
```



测试通过HTTP下载存储服务器上的文件

- 执行如下命令行
 - \$ cd ~/Projects/download
 - \$ wget http://127.0.0.1:8080/files/5fb0...0177
- 通过diff命令比对文件内容

```
tarena@TNV: ~/Projects/TNV.20201111/download
tarena@TNV:download$ wget http://127.0.0.1:8080/files/5fb0c3ed088d19807e46b570003fc4000177
--2020-11-15 14:24:37-- http://127.0.0.1:8080/files/5fb0c3ed088d19807e46b570003fc4000177
正在连接 127.0.0.1:8080... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 未指定 [application/octet-stream]
正在保存至：“5fb0c3ed088d19807e46b570003fc4000177.1”

5fb0c3ed088d19807e4 [ <=> ] 700.50M --.-KB/s in 17s

2020-11-15 14:24:54 (42.0 MB/s) - “5fb0c3ed088d19807e46b570003fc4000177.1” 已保存 [734529655]

tarena@TNV:download$
```



测试通过HTTP播放存储服务器上的流媒体

- 在QtPlayer中打开如下URL
 - <http://127.0.0.1:8080/files/5fb0...0177>



测试删除存储服务器上的文件

- 执行如下命令行
 - \$ cd ~/Projects/TNV/bin
 - \$./client 127.0.0.1:21000 delete tnvideo tnv001 5fb0...0177
- 检查存储服务器的存储目录和数据库

```
tarena@TNV: ~/Projects/TNV.20201111/bin
tarena@TNV:bin$ ./client 127.0.0.1:21000 delete tnvideo tnv001 5fb0c3ed088d19807e46b570003fc4000177
connect_manager.cpp(149), set_service_list: add one service: 127.0.0.1:21000, max connect: 16
connect_manager.cpp(122), init: no default connection set
02_conn.cpp(612), recvhead: bodylen: 34, command: 100, status: 0
02_conn.cpp(612), recvhead: bodylen: 0, command: 102, status: 0
Delete success: 5fb0c3ed088d19807e46b570003fc4000177
tarena@TNV:bin$
```



附录：程序清单



TNV/sql/storage.sql

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

```
CREATE TABLE `t_file_01` (  
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `status` tinyint(4) DEFAULT NULL,  
  `file_path` varchar(512) DEFAULT NULL,  
  `file_size` bigint(20) 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/storage.sql

```
CREATE TABLE `t_file_02` (  
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `status` tinyint(4) DEFAULT NULL,  
  `file_path` varchar(512) DEFAULT NULL,  
  `file_size` bigint(20) 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/storage.sql

```
CREATE TABLE `t_file_03` (  
  `id` varchar(256) NOT NULL DEFAULT '' COMMENT '文件ID',  
  `appid` varchar(32) DEFAULT NULL,  
  `userid` varchar(128) DEFAULT NULL,  
  `status` tinyint(4) DEFAULT NULL,  
  `file_path` varchar(512) DEFAULT NULL,  
  `file_size` bigint(20) 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/src/install.sh

```
#!/bin/bash
```

```
cd 02_tracker  
make clean && make all  
cd ../03_id  
make clean && make all  
cd ../04_storage  
make clean && make all  
cd ../05_client  
make clean && make all  
cd ../06_http  
make clean && make all
```

```
exit 0
```

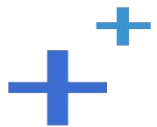


TNV/src/uninstall.sh

```
#!/bin/bash

cd 02_tracker
make clean
cd ../03_id
make clean
cd ../04_storage
make clean
cd ../05_client
make clean
cd ../06_http
make clean
cd ../../bin
rm -rf *

+ exit 0
```



复习课见