

Lab2: socket programming

鲁君一 517261910033

Lab2: socket programming

1. 文件组织结构
2. 运行方法
3. 运行结果
 - 3.1. 单线程服务器
 - 3.2. 多线程服务器
4. 测试

1. 文件组织结构

- `src` 文件夹下存放源代码
 - 客户端 `client.cc`,
 - 单线程服务器 `server.cc`,
 - 多线程服务器 `server_pthread.cc`
- `client_computer` 下有五个文件夹, 从 `c1` 到 `c5` 分别代表五个客户端本地文件夹.
每个客户端本地文件夹下存放客户端程序 `clientapp`, 以及将从服务器接收到的文件 `file.txt`
- `server_computer` 下存放将要发送的文件 `file.txt`, 以及用于生成 `file.txt` 的 shell 脚本 `generate.sh`
- `compile.sh` 用于生成可执行文件
- `delete.py` 用于批量删除服务器接受到的文件, 方便测试
- `csmodel.py` 为使用 `mininet` python API 编写的网络结构, 该网络中一个交换机与 6 个主机相连接, 将 `host1` 当作服务器, 其他为客户端, 在客户端上计算文件下载时间.

2. 运行方法

在终端打开 `mysocket` 文件夹, 输入并运行

```
sudo python3 csmodel.py
```

注: 可以先运行 `delete.py` 删除之前测试时遗留下来的文件 `file.txt` 来更好的检验程序

3. 运行结果

3.1. 单线程服务器

```
问题 输出 调试控制台 终端
abigail@ubuntu:~/Desktop/mysocket$ bash compile.sh
abigail@ubuntu:~/Desktop/mysocket$ cd server computer/
abigail@ubuntu:~/Desktop/mysocket/server_computer$ ./server

Hi,Iam running server.Listening to socket: No.3...
Successfully connect with one client with socket NO.4...
file_block_length = 61440
file_block_length = 61440
file_block_length = 61440
file_block_length = 61440
file_block_length = 61440
file_block_length = 61440
file_block_length = 57344
File: file.txt Transfer Finished!

Hi,Iam running server.Listening to socket: No.3...
[]

big il@ubuntu:~/Desktop/mysocket$ cd client computer/
big il@ubuntu:~/Desktop/mysocket/client_computer$ cd c1
big il@ubuntu:~/Desktop/mysocket/client_computer/c1$ ./clientapp
Recieve File: file.txt From Server[127.0.0.1] Finished! Downloading
time: 0.000368

big il@ubuntu:~/Desktop/mysocket/client_computer/c1$ []
```

注: 请修改 `client.cc` 中19行的 `SERVER_ADDR` 为'127.0.0.1'并重新编译,再运行上述代码

3.2.多线程服务器

```
abigail@ubuntu:~/Desktop/mysocket$ sudo python3 csmodel.py
[sudo] password for abigail:
*** Creating network
*** Adding controller
*** Adding hosts:
h1 h2 h3 h4 h5 h6
*** Adding switches:
s1
*** Adding links:
(10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h1, s1) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h2, s1) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h3, s1) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h4, s1) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h5, s1) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (h6, s1)
*** Configuring hosts
h1 (cfs 33333/100000us) h2 (cfs 33333/100000us) h3 (cfs 33333/100000us) h4 (cfs 33333/100000us) h5 (cfs 33333/100000us) h6 (cfs 33333/100000us)
*** Starting controller
c0
*** Starting 1 switches
s1 ... (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss) (10.00Mbit 5ms delay 0.00000% loss)
Dumping host connections
h1 h1-eth0:s1-eth1
h2 h2-eth0:s1-eth2
h3 h3-eth0:s1-eth3
h4 h4-eth0:s1-eth4
h5 h5-eth0:s1-eth5
h6 h6-eth0:s1-eth6
Recieve Data From Server 10.0.0.1 Failed!
Recieve File: file.txt From Server[10.0.0.1] Finished! Downloading time: 0.001749
Recieve File: file.txt From Server[10.0.0.1] Finished! Downloading time: 0.003405
Recieve File: file.txt From Server[10.0.0.1] Finished! Downloading time: 0.004018
Recieve File: file.txt From Server[10.0.0.1] Finished! Downloading time: 0.004522
Recieve File: file.txt From Server[10.0.0.1] Finished! Downloading time: 0.00422
*** Stopping 1 controllers
c0
*** Stopping 6 links
.....
*** Stopping 1 switches
s1
*** Stopping 6 hosts
h1 h2 h3 h4 h5 h6
*** Done
Mission complete, bye!
abigail@ubuntu:~/Desktop/mysocket$ []
```

注: 尽管出现"Received Data From Server Failed",但是接收文件并没有失败,可以运行 `delete.py` 后, 再运行 `csmodel`, 查看 `c1` 到 `c5` 的本地文件夹来检验.

4.测试

通过修改 `csmodel.py` 中client的数量, 以及修改 `generate.sh` 中生成文件的大小,我们可以获得以下结论:

- 客户端的数量越多,下载时间越长