实验报告

孔静 2014K8009929022

October 12, 2017

1 实验题目

Socket 应用编程实验

2 实验内容

2.1 基于 socket 的分布式字符统计程序

- Worker 1 和 Worker 2 分别监听端口 12345
- workers.conf 配置文件中存储每个 worker 的 IP 地址
- 需要统计的字符存放在 war_and_peace.txt 文件中

3 实验流程

3.1 基于 socket 的分布式字符统计程序

- Master 通过读取 workers.conf 配置文件，获取每个 worker 的 IP 地址，然后分别建立 socket 连接

```
//Read ip
fp = fopen("workers.conf","r");
if(fp == NULL){
  printf("Could not open file\n");
  return -1;
}
fgets(ip1, 16, fp);
fgets(ip2, 16, fp);
fclose(fp);
printf("Ip got\n");
```

- Master 获取 war_and_peace.txt 文件长度，将统计任务等分到所有的 worker

```c
//Count lines
if (argc == 1)
{
  printf("No file to read\n");
  return -1;
}
fp = fopen(argv[1],"r");
if(fp == NULL){
  printf("Could not open file\n");
  return -1;
}
while(!feof(fp)){
  fgets(message, 200, fp);
  total++;
}
fclose(fp);
printf("File has %d lines\n", total);
```

```c
//Send message
int * msg = (int *)message;
int i, length;
msg[0] = 0;
msg[1] = total / 2;
strcpy(&message[8], argv[1]);
length = 2 + (strlen(argv[1]) - 1) / 4 + 1;
for(i = 0; i < length; i++){
  msg[i] = htonl(msg[i]);
}
send(sock1, message, length * 4, 0);
msg[0] = msg[1];
msg[1] = htonl(total);
send(sock2, message, length * 4, 0);
```

- 每个 worker 收到消息后，进行解析，根据指定统计区间对文件进行统计

```c
//Count letter number
int letter[26], length;
char current;
memset(letter, 0, 26 * sizeof(int));

for(i = 0; i < startline; i++){
  fgets(message, 200, fp);
}

while(startline < endline && !feof(fp)){
  fgets(message, 200, fp);
  length = strlen(message);

  for(i = 0; i < length; i++){
    current = message[i];
    if(current >= 'a' && current <= 'z'){
      letter[current - 'a']++;
    }
    if(current >= 'A' && current <= 'Z'){
      letter[current - 'A']++;
    }
  }

  startline++;
}
fclose(fp);
```

- Worker 统计结束后，将每个字符出现的次数以 4 字节整数形式（网络字节序）返回给 Master，因此传输消息长度为 104 字节

```
//Send message back to server
memset(message, 0, 2000);
for(i = 0; i < 26; i++){
  msg[i] = htonl(letter[i]);
}
write(cs, message, 104);
```

- 编写脚本一键运行

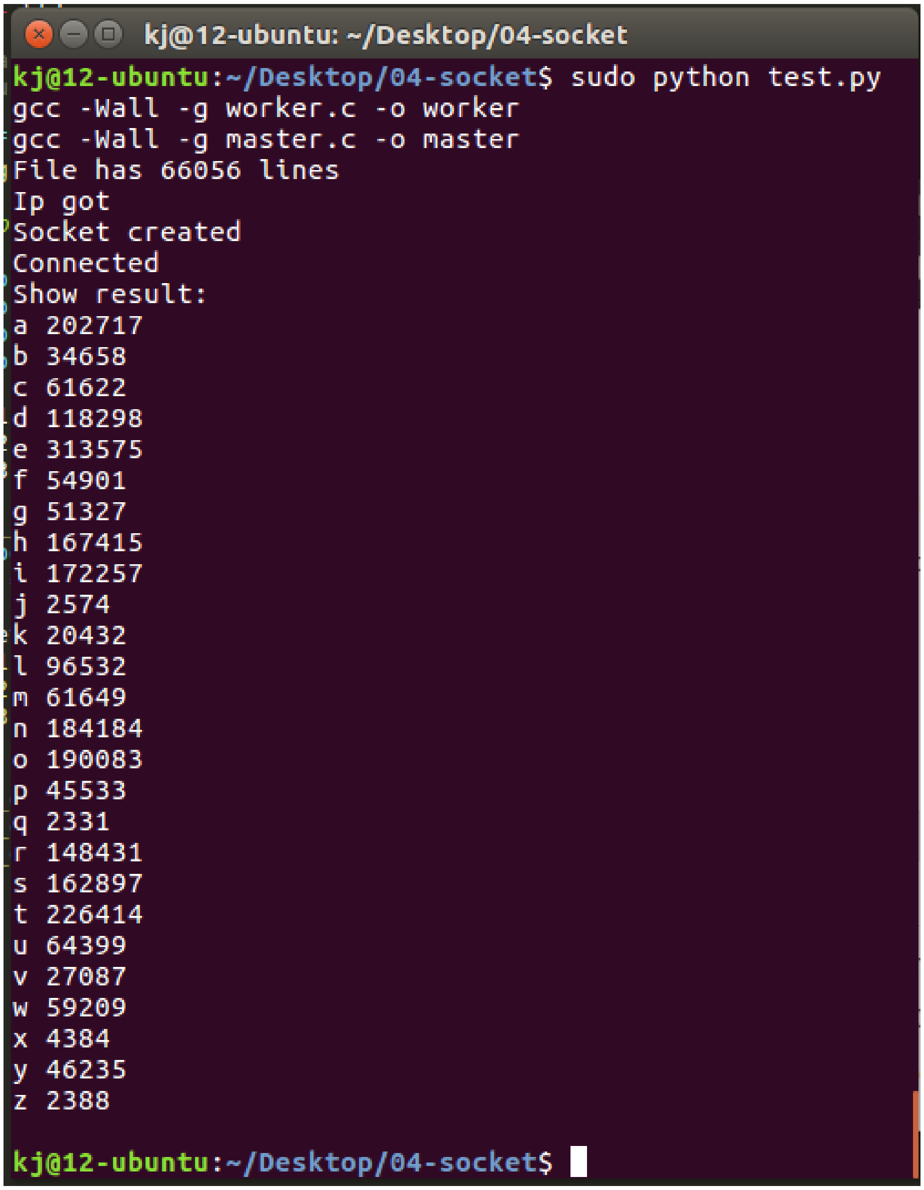
```
os.system("make")
topo = MyTopo()
net = Mininet(topo = topo)

net.start()

h1, h2, h3 = net.get('h1', 'h2', 'h3')
h2.cmd('./worker &')
h3.cmd('./worker &')
print h1.cmd('./master war_and_peace.txt')

net.stop()
os.system("make clean")
```

4 实验结果



kj@12-ubuntu: ~/Desktop/04-socket

```
kj@12-ubuntu:~/Desktop/04-socket$ sudo python test.py
gcc -Wall -g worker.c -o worker
gcc -Wall -g master.c -o master
File has 66056 lines
Ip got
Socket created
Connected
Show result:
a 202717
b 34658
c 61622
d 118298
e 313575
f 54901
g 51327
h 167415
i 172257
j 2574
k 20432
l 96532
m 61649
n 184184
o 190083
p 45533
q 2331
r 148431
s 162897
t 226414
u 64399
v 27087
w 59209
x 4384
y 46235
z 2388

kj@12-ubuntu:~/Desktop/04-socket$ 
```

5 结果分析

5.1 基于 socket 的分布式字符统计程序

对比例图，数据结果正确。