

Report on Project for MI 2.01

Hong Hanh

04 April 2019

1 Achievement

In this project, we implement a tool allowing the execution of word count application following the map-reduce principle and compare the execution time between centralized and map-reduce (2,3 nodes) version. The 1.1 GB dataset is generated with 1000 common words. The functions are executed successfully mentioned below:

- **Map:** Read file, produce count table, integrated into **Daemon**, and send results to **Launch**.
- **Reduce:** Compute the sum from map and send the results.
- **Daemon:** Bind socket for different ports, receive connection for receiving blocks and requesting **Map**'s execution.
- **Split:** Split 1GB file into blocks and sends blocks to **Daemon**.
- **Launch:** Fork and exec **Split**, receive progress from **Split** using pipe, connect to **Daemon**, and send signal to execute **Map**.
- **Centralized:** Count word in a big text file.
- **Generate:** Generate 1.1 GB text data from 1000 most common words.

The execution time of map-reduce version and centralized version are demonstrated in the table below:

		Execution time (s)
Centralized version		435
Map-reduce verion	2 Daemons	235
	3 Daemons	192

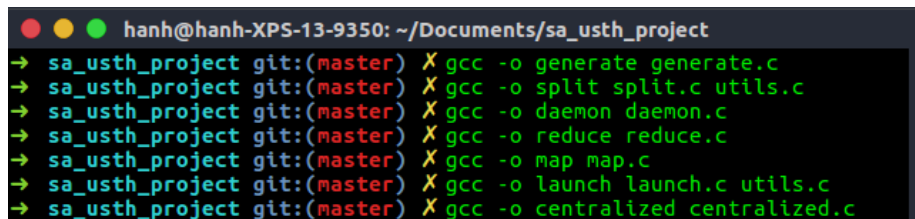
Table 1: Execution time of different versions

As we can see, the results illustrates that the the program with multiple Daemons reduces execution time to approximately a half compared to the centralized version.

2 Instructions on local machine

We instruct the implementation with the centralized version and the map-reduce case of 2 daemons (similar execution with 3 and 4 daemons).

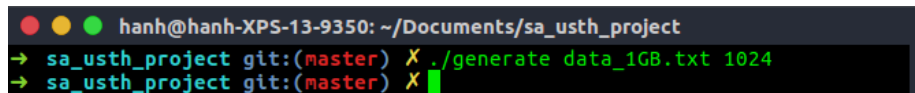
- Build program:
gcc -o generate generate.c
gcc -o split split.c utils.c
gcc -o daemon daemon.c
gcc -o reduce reduce.c
gcc -o map map.c
gcc -o launch launch.c utils.c
gcc -o centralized centralized.c



```
hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X gcc -o generate generate.c
→ sa_usth_project git:(master) X gcc -o split split.c utils.c
→ sa_usth_project git:(master) X gcc -o daemon daemon.c
→ sa_usth_project git:(master) X gcc -o reduce reduce.c
→ sa_usth_project git:(master) X gcc -o map map.c
→ sa_usth_project git:(master) X gcc -o launch launch.c utils.c
→ sa_usth_project git:(master) X gcc -o centralized centralized.c
```

Figure 1: Syntax to build program

- Execute program:
 - Generate 1 GB data:
./generate data_1GB.txt 1024



```
hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./generate data_1GB.txt 1024
→ sa_usth_project git:(master) X █
```

Figure 2: Generate 1GB data

- Centralized version:
./centralized data_1GB.txt “server/centralized_output”

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./centralized data_1GB.txt "server/centralized_output.txt"
Map is started, at (UNIX time: 1554537358) Sat Apr 6 14:55:58 2019
Map done, at (UNIX time: 1554537793) Sat Apr 6 15:03:13 2019
Execution time: 435 s
→ sa_usth_project git:(master) X █

```

Figure 3: Execute counting with centralized version

- Map-reduce version (2 nodes):
 - ./launch 2 "server/reduce_output.txt"
 - ./split 2 data_1GB.txt
 - ./daemon
 - ./daemon

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./launch 2 "server/reduce_output.txt"
Start listen on port 12346
Reduce is executed
New connection, socket fd is 7
Adding to list of sockets as 0
New connection, socket fd is 8
Adding to list of sockets as 1
Client 0 is started, at (Unix time: 1554538550) Sat Apr 6 15:15:50 2019
Client 1 is started, at (Unix time: 1554538554) Sat Apr 6 15:15:54 2019
Client 0 has disconnected.
Client 1 has disconnected.
Reduce done, at (UNIX time: 1554538785) Sat Apr 6 15:19:45 2019
Execution time: 235 s
→ sa_usth_project git:(master) X █

```

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
usth_project git:(master) X ./split 2 data_1GB.txt
Size of file data_1GB.txt: 1073744889 byte
Size of out file approx 536872444 byte
Done split file data_1GB.txt into:
server/block_0.txt
server/block_1.txt
Start listen on port 12345
New connection, socket fd is 5
New connection, socket fd is 6
All client (2) are connected
Start send file server/block_0.txt (536872446 byte) to client 0
Send: 536872446 byte
Send file server/block_0.txt to client 0 done!
Start send file server/block_1.txt (536872443 byte) to client 1
Send: 536872443 byte
Send file server/block_1.txt to client 1 done!
All file are send
→ sa_usth_project git:(master) X █

```

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Connected to Launch: 3
Connected to Split: 4
Success to create file client/data_1.txt on client
Received file client/data_1.txt (536872443 byte)
Map is executed
Map done
→ sa_usth_project git:(master) X █

```

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Connected to Launch: 3
Connected to Split: 4
Success to create file client/data_0.txt on client
Received file client/data_0.txt (536872446 byte)
Map is executed
Map done
→ sa_usth_project git:(master) X █

```

Figure 4: Execute counting with 2 Daemons

```

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./launch 3 "server/reduce_output.txt"
Start listen on port 12346
Reduce is executed
New connection, socket fd is 7
Adding to list of sockets as 0
New connection, socket fd is 8
Adding to list of sockets as 1
New connection, socket fd is 9
Adding to list of sockets as 2
Client 0 is started, at (unix time: 1554539536) Sat Apr  6 15:32:16 2019
Client 1 is started, at (unix time: 1554539536) Sat Apr  6 15:32:16 2019
Client 2 is started, at (unix time: 1554539536) Sat Apr  6 15:32:16 2019
Client 0 has disconnected.
Client 1 has disconnected.
Client 2 has disconnected.
Reduce done, at (UNIX) time: 1554539728) Sat Apr  6 15:35:28 2019
Execution time: 192 s
→ sa_usth_project git:(master) X[]

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Size of file data_1GB.txt: 1073744000 byte
Size of our file server: 1073744000 byte
Done split file data_1GB.txt into:
server/block_0.txt
server/block_1.txt
server/block_2.txt
Start listen on port 12345
New connection, socket fd is 0
New connection, socket fd is 1
New connection, socket fd is 2
All client (3) are connected
Start send file server/block_0.txt (1073744000 byte) to client 0
Send: 1073744000 byte
Send file server/block_0.txt to client 0 done!
Start send file server/block_1.txt (1073744000 byte) to client 1
Send: 1073744000 byte
Send file server/block_1.txt to client 1 done!
Start send file server/block_2.txt (1073744000 byte) to client 2
Send: 1073744000 byte
Send file server/block_2.txt to client 2 done!
→ sa_usth_project git:(master) X[]

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Connected to Launch 3
Connected to Split 4
Success to create file client/data_0.txt on client
Received file client/data_0.txt (1073744000 byte)
Map is executed
Map done
→ sa_usth_project git:(master) X[]

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Connected to Launch 3
Connected to Split 4
Success to create file client/data_1.txt on client
Received file client/data_1.txt (1073744000 byte)
Map is executed
Map done
→ sa_usth_project git:(master) X[]

hanh@hanh-XPS-13-9350: ~/Documents/sa_usth_project
→ sa_usth_project git:(master) X ./daemon
Connected to Launch 3
Connected to Split 4
Success to create file client/data_2.txt on client
Received file client/data_2.txt (1073744000 byte)
Map is executed
Map done
→ sa_usth_project git:(master) X[]

```

Figure 5: Similarly, execute counting with 3 Daemons

3 Workflow

For the map-reduce version, we executed the following programs:

- Execute **Generate** to generate 1.1 GB data.
- Run **Launch** with max number of clients (in our case, 2) and the file containing the final result ("server/reduce_output.txt").
 - Execute **Reduce**
 - Listen to port 12346.
- Run **Split** with max number of clients (i.e 2) and the 1.1 GB dataset.
 - Split file of into 2 file blocks (number of file blocks equals number of clients).
 - Listen to port 12345.
- Execute 2 **Daemons**.
- **Split** sends data to **Daemons**, **Daemons** saves data to local file.
- **Daemon** runs **Map** with local file.
- **Daemon** sends output to **Launch**.
- **Launch** waits the results of all **Daemons** and put the result to **Reduce**
- **Launch** saves **Reduce**'s output to file.

The details are commented on source code.