Student Name: SHIYANG CHEN
Login: SHIYANGC1
Student ID: 931880

Assignment 1 of Cluster and Cloud Computing

Assignment Title	**COMP90024 Cluster and Cloud Computing Assignment 1**
Subject Number	**COMP90024**
Subject Name	**Cluster and Cloud Computing**
Student Name	**Shiyang Chen**
Lecturer	**Richard Sinnott**
Due Date	**15/04/2020**

The invoking of application:

This application uses Python language for programming to achieve parallel computing. For detailed information of the python code, please refer to the attached zip file.
The beginning of the program obtains the start time firstly. Then through the parallel command, the program can obtain the rank and the set parallel size. The program splits the whole file among the processors where the row number(module) is matched with the current size. By the way of the paralleling, each row of the file is processed according to the difference of the remainder split by processors.
Then the program imports the json library to read the twitter file and process it on the machine.
After that it gets the parameters of the hashtag list and language list, uses Slurm options to open the specified file, such as bigTwitter.json, and gets the hashtags and languages by traversing the content. About the details on language codes for twitter, I got it from https://developer.twitter.com/en/docs/twitter-for-websites/twitter-forwebsites-supported-languages/overview.
Moreover, the program will skip the content in the file that does not meet the processing method, and perform a summary when the data length reaches 1000 to reduce memory usage.
In addition, it gathers the results of each processor through the reduce function. Furthermore, a new MPI operation is created to add the two values, which means integrating the results of the two processors at one time until the results of all processors are finally integrated in the reduction process.
Then it selects top ten of hashtags and languages which are used by the twitter users by the importing of counter library to output the search results.

Slurms:

The sbatch function is used to submit three Slurm scripts to the queue. And the number of nodes (CPU) and cores was set in the Slurm. With the command of “sh + Slurm file’s name”, we can see the results, which include top ten hashtags and languages and the time used by the processing.

Slurm 1: 1node and 1 core

```
#!/bin/bash
#SBATCH --account=COMP90024
#SBATCH --partition=cloud
#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --time=0-00:12:00
#SBATCH --job-name=Twitter_job1
#SBATCH -o slurm_job1.out
# Use this email address:
echo "----------- 1-node-1-core -----------"
echo ""
# Load required modules
module load Python/3.4.3-goolf-2015a
# Launch multiple process python code
time mpiexec python test.py bigTwitter.json
```

Slurm 2: 1node and 8 cores

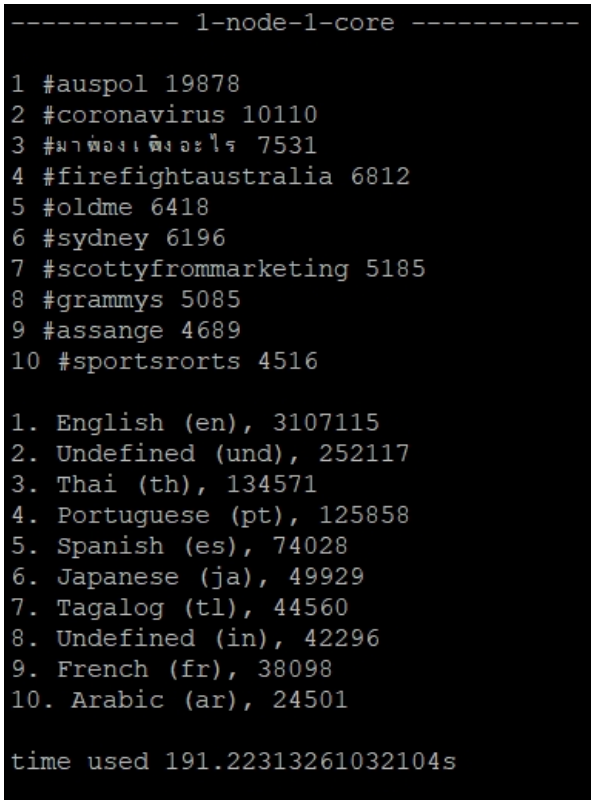
```
#!/bin/bash
#SBATCH --account=COMP90024
#SBATCH --partition=cloud
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --time=0-00:12:00
#SBATCH --job-name=Twitter_job2
#SBATCH -o slurm_job2.out
# Use this email address:
echo "----------- 1-node-8-core -----------"
echo ""
# Load required modules
module load Python/3.4.3-goolf-2015a
# Launch multiple process python code
time mpiexec python test.py bigTwitter.json
```

Slurm 3: 2 nodes and 8 cores

```
#!/bin/bash
#SBATCH --account=COMP90024
#SBATCH --partition=cloud
#SBATCH --nodes=2
#SBATCH --ntasks=8
#SBATCH --cpus-per-task=4
#SBATCH --time=0-00:12:00
#SBATCH --job-name=Twitter_job3
#SBATCH -o slurm_job3.out
# Use this email address:
echo "----------- 2-node-8-core -----------"
echo ""
# Load required modules
module load Python/3.4.3-goolf-2015a
# Launch multiple process python code
time mpiexec python test.py bigTwitter.json
```

The performance on different numbers of nodes and cores:

1 node and 1 core:



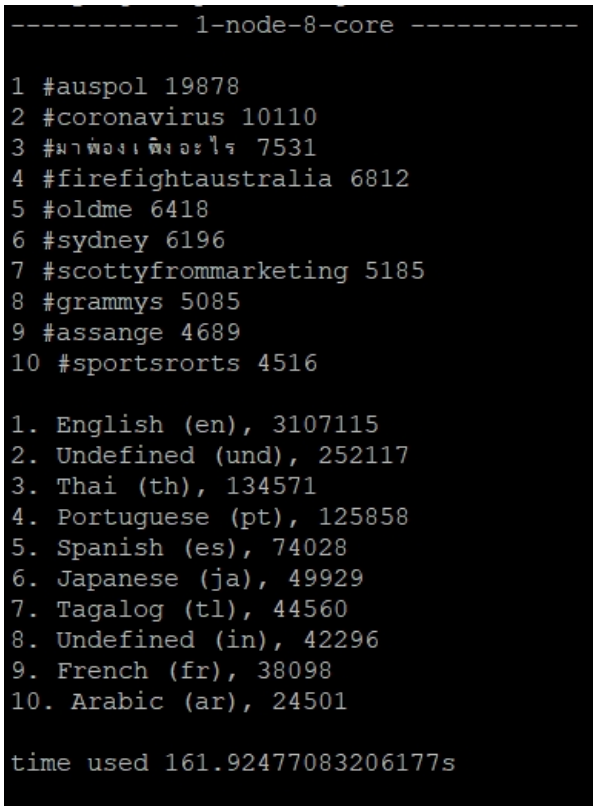
```
----------- 1-node-1-core -----------

1 #auspol 19878
2 #coronavirus 10110
3 #มาห่องเติงอะไร 7531
4 #firefightaustralia 6812
5 #oldme 6418
6 #sydney 6196
7 #scottyfrommarketing 5185
8 #grammys 5085
9 #assange 4689
10 #sportsrorts 4516

1. English (en), 3107115
2. Undefined (und), 252117
3. Thai (th), 134571
4. Portuguese (pt), 125858
5. Spanish (es), 74028
6. Japanese (ja), 49929
7. Tagalog (tl), 44560
8. Undefined (in), 42296
9. French (fr), 38098
10. Arabic (ar), 24501

time used 191.22313261032104s
```

1 node and 8 cores:



```
----------- 1-node-8-core -----------

1 #auspol 19878
2 #coronavirus 10110
3 #มาห่องเติงอะไร 7531
4 #firefightaustralia 6812
5 #oldme 6418
6 #sydney 6196
7 #scottyfrommarketing 5185
8 #grammys 5085
9 #assange 4689
10 #sportsrorts 4516

1. English (en), 3107115
2. Undefined (und), 252117
3. Thai (th), 134571
4. Portuguese (pt), 125858
5. Spanish (es), 74028
6. Japanese (ja), 49929
7. Tagalog (tl), 44560
8. Undefined (in), 42296
9. French (fr), 38098
10. Arabic (ar), 24501

time used 161.92477083206177s
```

2 nodes and 8 cores:

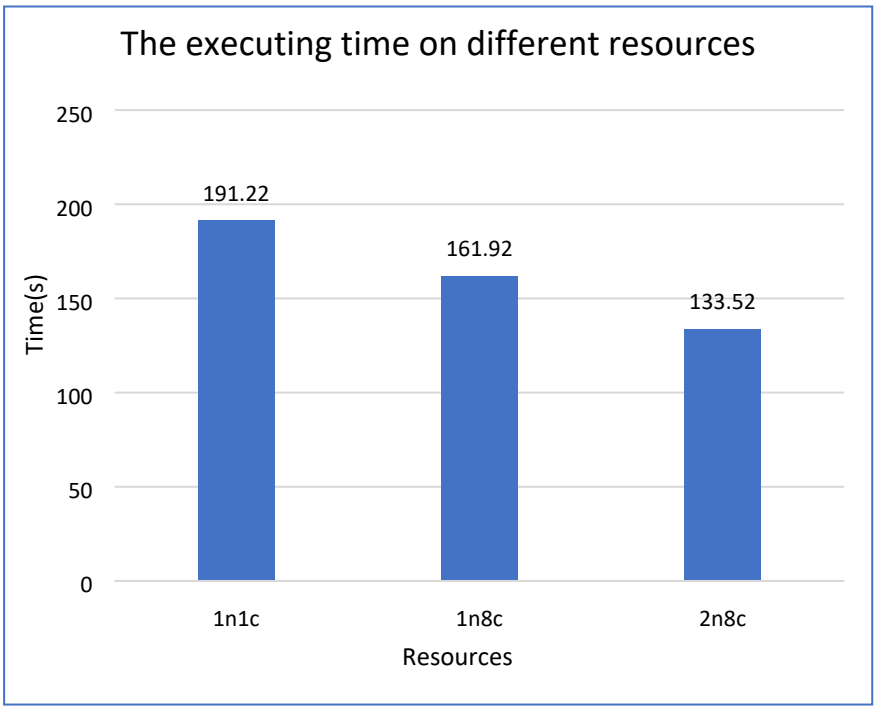
```
---------- 2-node-8-core ----------

1 #auspol 19878
2 #coronavirus 10110
3 #มาห่องเพิงอะไร 7531
4 #firefightaustralia 6812
5 #oldme 6418
6 #sydney 6196
7 #scottyfrommarketing 5185
8 #grammys 5085
9 #assange 4689
10 #sportsrorts 4516

1. English (en), 3107115
2. Undefined (und), 252117
3. Thai (th), 134571
4. Portuguese (pt), 125858
5. Spanish (es), 74028
6. Japanese (ja), 49929
7. Tagalog (tl), 44560
8. Undefined (in), 42296
9. French (fr), 38098
10. Arabic (ar), 24501

time used 133.52295207977295s
```

Resources	1n and 1c	1n and 8c	2n and 8c
Time(s)	191.22	161.92	133.52



Through the execution time results, it can be fount that using more nodes and cores to parallel can reduce the using time efficiently. However, this reduction is not a proportional reduction based on the number of nodes or cores.

From the perspective of program operation, the collaboration between threads during node (CPU) operation needs to be completed through the network or the motherboard bus, which has low efficiency and thus affects the overall performance, while multi-core nodes (CPUs) can be completed through shared cache and main memory, and the collaboration efficiency is higher.

In general, the partitioning of the parallelism which decomposes the computation activities and data into smaller tasks cause a positive influence on the execution time by using more nodes and cores. However, flow of information and coordination among tasks that are created in the partitioning stage and the merging of the tasks could be difficult to do. Therefore, the reducing of time due to the multi-node and multi-core is not linear.