CLUSTER AND CLOUD COMPUTING ASSIGNMENT-1

HPC TWITTER PROCESSING

Harshal Shah - 1020849
Kamakshi Banavalikar - 1039197

Problem Description:

The main task of the assignment is to implement a simple, parallelized application leveraging the University of Melbourne HPC facility SPARTAN. The main objective of the assignment is to identify the top 10 most frequently occurring hashtags (#) and the most commonly used language for tweeting.

Specifications:

The dataset consists of three json files, the tinyTwitter, the smallTwitter, and the bigTwitter. The tinyTwitter and the smallTwitter files are used for testing the program whereas the bigTwitter is used for final implementation. In addition, the python programming language is used to write the code and Message Passing Interface (MPI) library in python is used for parallel processing. Slurm is a job scheduling system that has also been used to send jobs to the Spartan HPC.

Methodology:

The program begins by importing the significant modules namely Counter, time, JSON, mpi4py, re, operator, etc. This is followed by the declaration of variable and the necessary data structures. Furthermore, the ISO 639-1 codes have also been initialized for future references. Additionally, these are a few methods defined:

1. Convert: To convert the tuple into the dictionary.
2. Rename_keys: To rename the keys of the dictionary to make it more logical.
3. checkEnglish: To check if there are any non-English characters in the hashtag.

The crucial part of the assignment was to successfully load and parse the bigTwitter file without storing the entire file into the memory. This has been taken care of by reading the file line by line. Consequently, the processors process each line in parallel, based on their rank. The rank and the size of the processors are fetched using the get_Rank and the get_Size functions available from the MPI library. After computing the results from all the different processes, MPI's gather function plays an important role in merging all the results which can help the master process to generate the final outcome towards the end of the task performed by all the processes.

The final result includes the frequently occurring hashtags and the languages used for tweeting, the method calculate_tweets help in performing the task. This function takes a single line of the tweet in the json format as an input and extracts all the possible string that starts with "#" and the ISO language code used for tweeting. However, there are certain rules to be considered for a string to be a hashtag. Every string that starts with a "#" is not necessarily a significant hashtag. A valid hashtag is the one that does not contain any punctuations or special characters, except hash (#) or underscores(_) which can be used between the letters or the numbers. Also, a string that contains only numbers after the # symbol is not a valid hashtag. Additionally, the string hashtag which starts with numbers is not a valid hashtag too. In this implementation, the calculate_tweets method handles the preprocessing part of the strings and helps finding the valid hashtags. It ignores the strings that have punctuations in between the strings as per the hashtag conditions. Correspondingly, it does not include the strings that consist only

of digits. It also converts all the hashtags to lower cases before computing the final count. The function allows the strings that end with a full stop(.) and remove that extra character as the string can still be considered a valid hashtag. This method also ignores the hashtags which have non-English characters used. Another criterion for a valid hashtag is to check for the length of the string to be greater than 1. Once all the above conditions are satisfied, the function appends the hashtags into a list and the languages used for tweeting into another list.

Furthermore, after the MPI's gather function does its work, the master process (with rank 0) does the final work of separating the two lists. One list contains the hashtags and the second list contains the languages used for tweeting. The Counter library helps in the calculation of the most frequently occurring hashtags and the languages which are mostly used for tweeting. Finally, the results are stored in the dictionary and are printed with the help of the operator module in python in the reverse/ descending order of the counts, thereby giving the desired results. The results obtained for the report are the most frequently used hashtags and the languages used for tweeting from the bigTwitter file.

Comparing the nodes and the cores:

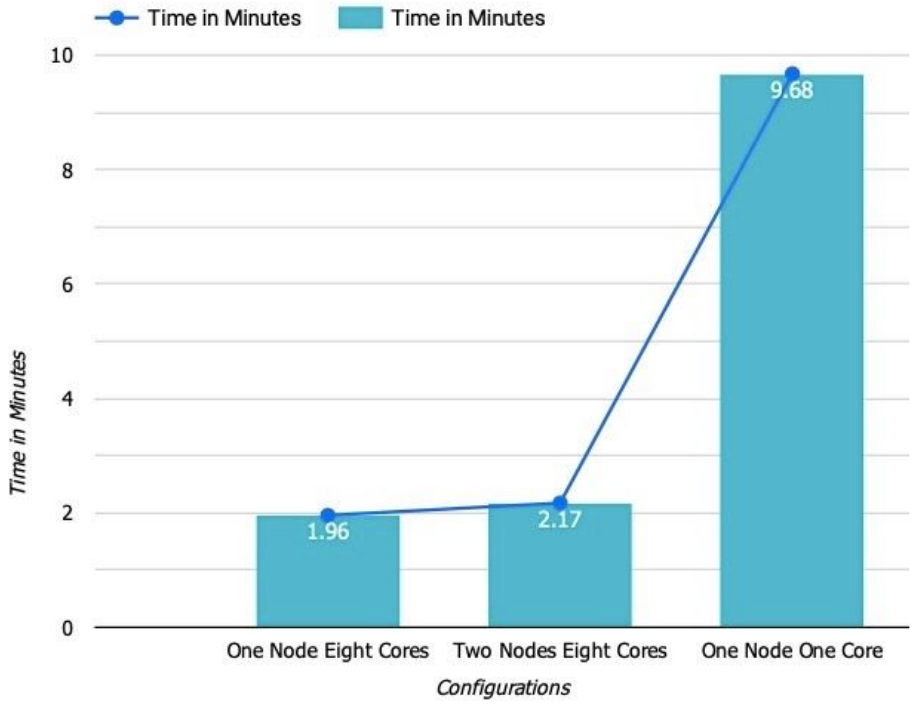

Figure 1: Time taken for Execution

The above graph illustrates the time in minutes vs configuration graph. As the configuration changes the time variable also changes. It has been observed that the one node one core is the slowest as it consumes more time, this is because the processing load is completely on a single core. On the other hand, the two nodes' eight cores and the one node eight cores have similar time consumption but are faster than the one core's performance. This indicates that the eight cores distribute the tasks and there is no burden on a single core which leads to efficient run time. Moreover, the table below signifies that one node eight core's real-time is the most but its system time is the least.

Hence, this comparison concludes that parallel processing is more efficient and powerful.

Time / Configuration	One node Eight cores	Two nodes Eight cores	One node One core
Real-Time	1m 57.979s	2m 11.189s	9m 41.174s
User Time	13m 54.891s	7m 22.657s	9m 30.884s
System Time	1m 42.705s	1m 18.361s	0m 9.880s

Table 1: Real, User and System time for all three slurm configurations

Slurm Scripts:

***One Node One Core*:**

```
#!/bin/bash
#SBATCH --ntasks=1
#SBATCH --time=0-00:15:00
#SBATCH --partition=physical

# Load required modules
module load Python/3.5.2-goolf-2015a

# Launch multiple process python code
echo "Searching for hashtags"
time mpiexec -n 1 python3 Assignment_1_Code.py -i
```

***Two Node Eight Cores*:**

```
#!/bin/bash
# 2 nodes, 4 tasks per node = 8 cores
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=4
#SBATCH --time=0-00:20:00
#SBATCH --partition=physical

# Load required modules
module load Python/3.5.2-goolf-2015a

# Launch multiple process python code
echo "Searching for hashtags"
time mpiexec -n 4 python3 Assignment_1_Code.py -i
```

***One Node Eight Cores*:**

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks=8
#SBATCH --time=0-00:15:00
#SBATCH --partition=physical

# Load required modules
module load Python/3.5.2-goolf-2015a

# Launch multiple process python code
echo "Searching for hashtags"
time mpiexec -n 8 python3 Assignment_1_Code.py -i
```

<u>Conclusion</u>:

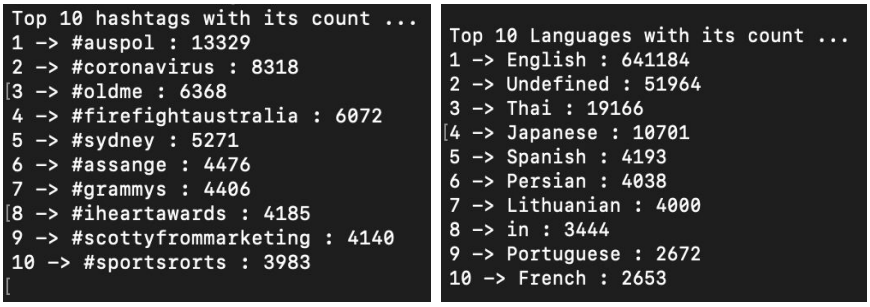

Figure 2: Final Counts of hashtags and languages

In the top 10 hashtags, #auspol is the most commonly used hashtag, this indicates that the users are more opinionated about Australian Politics. The second most common hashtag is #coronavirus, this shows that the users are concerned and are spreading awareness about the deadly coronavirus. Nevertheless in the top 10 languages used to tweet, English has the highest count, which is quite reasonable since Sydney has the majority of English speaking crowd. On the contrary, there are a lot of users who use language which is undefined by the ISO 639-1 codes.