Cluster and Cloud Computing Assignment 1 – HPC Twitter Processing

Problem Description

Your task in this programming assignment is to implement a simple, parallelized application leveraging the University of Melbourne HPC facility SPARTAN. Your application will use a large Twitter dataset for Sydney. Your objective is to identify the top 10 most frequently occurring hashtags (#) and the languages most commonly used for tweeting.

You should be able to log in to SPARTAN through running the following command:

ssh your-unimelb-username@spartan.hpc.unimelb.edu.au

with your University password. Thus I would log in as:

ssh rsinnott@spartan.hpc.unimelb.edu.au

If you are a Windows user then you may need to install an application like Putty.exe to run *ssh*. (If you are coming from elsewhere with different firewall rules, then you may need to use a VPN).

The files to be used in this assignment are accessible at:

- /data/projects/COMP90024/bigTwitter.json
  - this is the main 20.7Gb JSON file to use for your final analysis and report write up, i.e. **do not use the bigTwitter.json file for software development and testing.**
- /data/projects/COMP90024/smallTwitter.json
  - smallTwitter.json this a 23.9Mb JSON file should be used for testing;
- /data/projects/COMP90024/tinyTwitter.json
  - tinyTwitter.json this a 4.7Mb JSON file should be used for testing
  - You may also decide to use the smaller JSON files on your own PC/laptop to start with.

You should make a symbolic link to these files, i.e. you should run the following commands at the Unix prompt **from your own user directory on SPARTAN**:

```
ln –s /data/projects/COMP90024/bigTwitter.json
ln –s /data/projects/COMP90024/smallTwitter.json
ln –s /data/projects/COMP90024/tinyTwitter.json
```

Once done you should see something like the following **in your home directory**:

```
lrwxrwxrwx 1 rsinnott unimelb  40 Mar 22 15:06 bigTwitter.json -> /data/projects/COMP90024/bigTwitter.json
lrwxrwxrwx 1 rsinnott unimelb  39 Mar 22 15:06 smallTwitter.json -> /data/projects/COMP90024/smallTwitter.json
lrwxrwxrwx 1 rsinnott unimelb  38 Mar 22 15:06 tinyTwitter.json -> /data/projects/COMP90024/tinyTwitter.json
```

Your assignment is to (*eventually*!) search the large Twitter data set and identify the top 10 most commonly used hashtags and the number of times they appear. The results should look something like:

1. #covid19, 10,987
2. #qantas, 9,876
3. #maccas, 8,765

…

10. #trump, 1,234

Obviously, these numbers and hashtags are representative of the hashtags that occur in the file. A matching hashtag string can match if it has upper/lower case exact substrings, e.g. #covid19 and #COVID19 are a match. A hashtag should follow the Twitter rules, e.g. no spaces and no punctuation are allowed in a hashtag.

Your solution should also identify the languages used for tweeting and the number of times the language is used for the provided tweets.

1. English (en), 3,789,012
2. Urdu (ur), 234,567
3. German (de), 123,456
4. Chinese Simplified (zh-cn), 98,765

   …

10. Spanish (es), 789

Again, the numbers here are representative. See https://developer.twitter.com/en/docs/twitter-for-websites/twitter-for-websites-supported-languages/overview for details on language codes for twitter.

Your application should allow a given number of nodes and cores to be utilized. Specifically, **your application should be run once** to search the *bigTwitter.json* file on each of the following resources:

- 1 node and 1 core;
- 1 node and 8 cores;
- 2 nodes and 8 cores (with 4 cores per node).

The resources should be set when submitting the search application with the appropriate *SLURM* options. Note that you should run a single *SLURM* job three separate times on each of the resources given here, i.e. you should not need to run the same job 3 times on 1 node 1 core for example to benchmark the application. (This is a shared facility and this many COMP90024 students will consume a lot of resources!).

You can implement your search using any routines that you wish from existing libraries however it is strongly recommended that you follow the guidelines provided on access and use of the SPARTAN cluster. Do not for example think that the job scheduler/SPARTAN automatically parallelizes your code – it doesn't! You may wish to use the pre-existing MPI libraries that have been installed for C, C++ or Python. You should feel free to make use of the Internet to identify which JSON processing libraries you might use.

Your application should return the final results and the time to run the job itself, i.e. the time for the first job starting on a given SPARTAN node to the time the last job completes. You may ignore the queuing time. The focus of this assignment is not to optimize the application to run faster, but to learn about HPC and how basic benchmarking of applications on a HPC facility can be achieved and the lessons learned in doing this on a shared resource.

Final packaging and delivery

You should write a brief report on the application – **no more than 4 pages**!, outlining how it can be invoked, i.e. it should include the scripts used for submitting the job to SPARTAN, the approach you took to parallelize your code, and describe variations in its performance on different numbers of nodes and cores. Your report should also include a single graph (e.g. a bar chart) showing the time for execution of your solution on 1 node with 1 core, on 1 node with 8 cores and on 2 nodes with 8 cores.

Deadline

The assignment should be submitted to Canvas as a zip file. The zip file must be named with the first named student in each team (as listed in the sharepoint file sent around previously - https://unimelbcloud-my.sharepoint.com/:x:/g/personal/yao_pan_unimelb_edu_au/ESOBd3rWPfxAotVrTAnzcG0Bhm-fzirbjngopvqWhpfJsQ?e=ayOdah). This includes Forename-Surname-Student ID, e.g. *<Joe-Smith-0123456>.zip*. Only one report is required per student pair.

The deadline for submitting the assignment is: **Monday 8th April (by 12 noon!)**.

It is strongly recommended that you do not do this assignment at the last minute, as it may be the case that the Spartan HPC facility is under heavy load when you need it and hence it may not be available! You have been warned…!!!!

Marking

The marking process will be structured by evaluating whether the assignment (application + report) is compliant with the specification given. This implies the following:

- A working demonstration – **60% marks**
- Report and write up discussion – **40% marks**

Timeliness in submitting the assignment in the proper format is important. **A 10% deduction per day will be made for late submissions.**

You are free to develop your system where you are more comfortable with (at home, on your PC/laptop, in the labs, on SPARTAN itself - but not on the *bigTwitter.json* file until you are ready!). Your code should of course work on SPARTAN.