

Cluster and Cloud Computing Assignment 1 – Multicultural City

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 and a grid/mesh for Sydney to identify the languages used in making Tweets. Your objective is to count the number of different languages used for tweets in the given cells and the number of tweets in those languages and hence to calculate the multicultural nature of Sydney!

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

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

with the password you set for yourself on *karaage* (<https://dashboard.hpc.unimelb.edu.au/karaage>). Thus, I would log in as:

```
ssh rsinnott@spartan.hpc.unimelb.edu.au  
password = my karaage password (not my UniMelb password)
```

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 20Gb+ 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 small JSON file that should be used for testing with 5000 tweets;
- [/data/projects/COMP90024/tinyTwitter.json](#)
 - tinyTwitter.json this a very small JSON file that should be used for initial testing with 1000 tweets;
 - You may also decide to use the tiny/small JSON files on your own PC/laptop to start with.

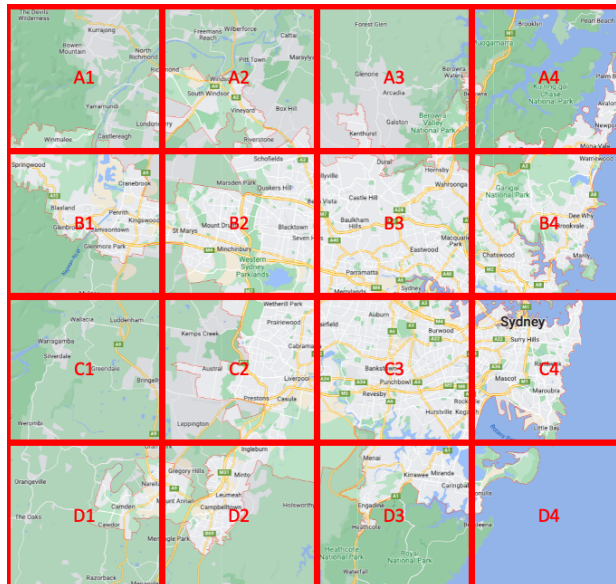
You should make a symbolic link to these files, i.e. you should run the following commands at the Unix prompt **from your own user directory on SPARTAN**:

```
ln -s /data/projects/COMP90024/bigTwitter.json  
ln -s /data/projects/COMP90024/smallTwitter.json  
ln -s /data/projects/COMP90024/tinyTwitter.json  
ln -s /data/projects/COMP90024/sydGrid.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  
lrwxrwxrwx 1 rsinnott unimelb 41 Mar 22 15:06 sydGrid.json -> /data/projects/COMP90024/sydGrid.json
```

The *sydGrid.json* file includes the latitudes and longitudes of a range of gridded boxes as illustrated in the figure below, i.e., the latitude and longitude of each of the corners of the boxes is given in the file.



Your assignment is to (eventually!) search the large Twitter data set (*bigTwitter.json*) and **using the language used when tweeting, the number of tweets** in those languages and the **tweet location** (lat/long) count the total number of tweets in a given cell that are made in different languages. The final result will be a score for each cell with the following format, where the numbers are obviously representative.

Cell	#Total Tweets	#Number of Languages Used	#Top 10 Languages & #Tweets)
A1	11,111	11	(English-9,000, Chinese-555, French-444, ...Greek-66)
A2	22,222	22	(English-21,000), Turkish-77, Swedish-66, ...French-2)
A3	33,333	33	etc etc
A4	44,444	44	
		...	
D3	55,555	55	
D4	66,666	66	

Here cell A1 has 11,111 tweets in total with 11 different languages used for tweets with the most popular being English (9,000 tweets), Chinese (555 tweets), French (444 tweets) with 10th most popular being Greek (66 tweets). Cell A2 has 22 languages used for tweeting with the most popular being English (21,000), Turkish (77 tweets), Swedish (66 tweets) and French being the 10th most popular language (2 tweets). Information on the classification of languages used for tweeting is given in <https://developer.twitter.com/en/docs/twitter-for-websites/supported-languages>. You may treat Simplified Chinese (zh-cn) and Traditional Chinese (zh-tw) as both being Chinese. Tweets with *null* or *undefined* (*und*) for the language attribute can be ignored. Further information on languages that might be used for tweeting is given in https://en.wikipedia.org/wiki/IETF_language_tag. Tweets with no location information can be ignored. Tweets made outside of the Grid can also be ignored.

If a tweet occurs right on the border of two cells, e.g., exactly between the B1/B2 cell border then assume the tweet occurs in B1 (i.e., to the cell on the left). If a tweet occurs exactly on the border between B2/C2 then assume the tweet occurs in C2 (i.e., to the cell below). If a tweet occurs anywhere else on the boundary of a cell, e.g. the upper or leftmost border of A1 then it can be regarded as being in cell A1.

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 solution 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 students named in each team and their student Ids. That is, *ForenameSurname-StudentId:ForenameSurname-StudentId* might be *<SteveJobs-12345:BillGates-23456>.zip*. Only one report is required per student pair.

The deadline for submitting the assignment is: **Wednesday 6th 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.