Cluster and Cloud Computing Assignment 2 - Australia Social Media Analytics on the Cloud

Background

In development and delivery of non-trivial software systems, working as part of a team is generally (typically!) the norm. This assignment is very much a group project. Students will be put into software teams to work on the implementation of the system described below. These will be teams of up to 5 students. In this assignment, students need to organize their team and their collective involvement throughout. There is no team leader as such, but teams may decide to set up processes for agreeing on the work and who does what. Understanding the dependencies between individual efforts and their successful integration is key to the success of the work and for software engineering projects more generally. If teams have "*issues*", then please let me know asap and I will help resolve them.

Assignment Description

The software engineering activity builds on the lecture materials describing Cloud systems and especially the UniMelb Research Cloud and its use of OpenStack; on CouchDB and the kinds of data analytics (e.g., MapReduce) that CouchDB supports. The project focuses on an export of Twitter data from the Australian Data Observatory (ADO - www.ado.eresearch.unimelb.edu.au)[1], data to be harvested by students from the Mastodon APIs and data from the Spatial Urban Data Observatory (SUDO - https://sudo.eresearch.unimelb.edu.au). The focus of this assignment is to use a large Twitter corpus (that will be provided) to tell interesting stories of life in Australian **and** importantly how social media data can be used alongside/compared with/augment the official data available within the SUDO platform to improve our knowledge of life in Australia. Teams can download data from the SUDO platform, e.g., as JSON, CSV or Shapefiles (see workshop week 2). This data can/should be included into the team's CouchDB database for analysis and comparison with the social media data. Furthermore, posts (*toots*) from the Mastodon APIs should be harvested (streamed) to further explore the social media stories. Note that Mastodon has much less data and the vast majority cannot be geocoded accurately to Australia, hence there is no need for extensive Mastodon analysis and integration/detailed comparison with the Twitter/SUDO data.

The teams should develop a Cloud-based solution that exploits virtual machines (VMs) on the UniMelb Research Cloud for harvesting and processing Mastodon *toots*. The teams should produce a solution that can be run (in principle) across any node of the UniMelb Research Cloud to harvest and store social media data and scale up/down as required with the data being harvested, processed and incorporated into the *existing* (running) CouchDB database. Multiple Mastodon servers can/should be used to support the different scenarios (and to lessen the load on any one Mastodon server). Teams are expected to have multiple instances of this application running on the UniMelb Research Cloud together with an associated CouchDB database containing the amalgamated collection of *toots*, tweets and SUDO data. The CouchDB setup may be a single node or based on a clustered setup.

There are multiple Mastodon servers that can be used for data collection (see https://mastodonservers.net/) for example. Teams are required to obtain an API Key to use these servers – noting that the servers are often not as mature/well-resourced as Twitter. Students should explore Mastodon servers that have a connection with Australia or a connection with the specific scenarios being explored, e.g., crypto currencies, LGBTQI, … noting that Mastodon users may come from anywhere, i.e., Mastodon.au, Aus Social, Mastodon Melbourne include users from around the world. Students may want to explore other sources of data they find on the Internet, e.g., information on weather, sport events, TV shows, stock market rise/falls, official statistics on Covid-19 however these are not compulsory to complete the work.

[1] Note that it is unlikely that Twitter data will be accessible through the APIs, hence a large volume of data will be provided directly. (Blame Mr Musk!) ☹

Teams have been allocated 8 virtual CPUs and 500Gb of volume storage. Students may also have access to the NeCTAR Research Cloud as individual users and can test/develop their applications using their own (small) VM instances, e.g., using personal instances such as pt-12345, noting that there is no persistence in these small, free and dynamically allocated VMs.

Teams are expected to develop a range of analytic scenarios, e.g., using the MapReduce capabilities offered by CouchDB for social media analytics and comparing the data with official data from SUDO. Teams are free to explore any scenarios that connect "in some way" to the SUDO data. Teams are encouraged to be creative here. *A prize will be awarded for the most interesting scenarios identified!* For example, teams may look at scenarios such as:

- How many tweets mention Covid-19 or coronavirus and are these clustered in certain areas, e.g., rich vs poor suburbs or in statistical areas where there are more/less hospitals etc? How does this compare with Mastodon data, e.g., if 1% of the tweets mention Covid-19, do we see the same percentages in the Mastodon toots?
- How many tweets mention the war in Ukraine, and can we determine public opinion in areas where there are different European languages spoken, e.g., Ukrainian, Russia, German etc? How does this compare with Mastodon data, e.g., for posts that mention the Ukraine war, how does the Australian/global sentiment compare?
- Do the different languages used when tweeting correlate with the cultures we would expect to find in those areas, e.g., more Chinese live in Box Hill in Melbourne hence we would expect to see for tweets tagged as Chinese from those suburbs, or Italians in Carlton etc? How does the language tweeted compare to the amount of non-English Mastodon posts from a multi-language Mastodon server such as https://mastodonservers.net/server/928-mastodon-ai?
- Does language use, e.g., vulgar words used in Twitter happen more or less in wealthy or poor areas? Do we seem the same amount of vulgarity in Mastodon data in Australia/globally?
- Do we see the same topics being discussed on Mastodon as we do on Twitter?

The above are examples – students may decide to create their own analytics based on the data they obtain. Students are not expected to build advanced "general purpose" data analytic services that can support any scenario but show how tools like CouchDB with targeted data analysis capabilities like MapReduce when provided with suitable inputs can be used to capture the essence of life in Australia.

A front-end web application is required for visualising these data sets/scenarios.

For the implementation, teams are recommended to use a commonly understood language across team members – most likely Java or Python. Information on processing tweets can be found on the web, e.g. see https://dev.twitter.com/ and in the lecture (week 6) on Mastodon client data harvesting. Teams are free to use any pre-existing software systems that they deem appropriate for the analysis and visualisation capabilities, e.g. Javascript libraries, Googlemaps etc. Existing sentiment analysis tools may be used, e.g., NLTK, Vader, TextBlob. Existing topic models can be used, e.g. LDA, BERT.

Error Handling

Issues and challenges in using the UniMelb Research Cloud for this assignment should be documented. You should describe the limitations of mining twitter content and language processing (e.g., sarcasm). You should outline any solutions developed to tackle such scenarios.

Final packaging and delivery

You should collectively write a team report on the application developed and include the architecture, the system design and the discussions that lead into the design. You should describe the role of the team members in the delivery of the system and where the team worked well and where issues arose and how they were addressed. The team should illustrate the functionality of the system through a range of scenarios and explain why you chose the specific examples. Teams are encouraged to write this report in the style of a paper than can ultimately be submitted to a conference/journal.

Each team member is expected to complete a confidential report on their role in the project and the experiences in working with their individual team members. This will be handed in separately to the final team report. (This is not to be used to blame people, but to ensure that all team members are able to provide feedback and to ensure that no team has any member that does nothing!!!).

The length of the team report is not fixed. Given the level of complexity of the assignment and total value of the assignment a suitable estimate is a report in the range of 20-25 pages. A typical report will comprise:

- A description of the system functionalities, the scenarios supported and why, together with graphical results, e.g., pie-charts/graphs/maps of tweet/toot analysis for the scenarios;
- A simple user guide for testing including system deployment and end user invocation/usage of the systems;
- System design and architecture and how/why this was chosen;
- A discussion on the pros and cons of the UniMelb Research Cloud and tools and processes for image creation and deployment;
- Teams should also produce a video of their system that is uploaded to YouTube (these videos can last longer than the UniMelb deployments unfortunately!) and the link for this included in the report;
- Reports should also include a link to the source code (github or bitbucket). It is recommended that all students commit their code to the code repository rather than delegate this to a single team member. This can provide an evidence base if teams have "issues".

It is important to put your collective team details (team, city, names, surnames, student ids) in:

- the head page of the report;
- as a header in each of the files of the software project.

Individual reports describing your role and your teams' contributions should be submitted through a Qualtrics link that will be sent through in due course.

Implementation Requirements

Teams are expected to use:

- a version-control system such as GitHub or Bitbucket for sharing source code.
- MapReduce based implementations for analytics using CouchDB's built in MapReduce capabilities.
- the system should support scripted deployment capabilities. This means that your team will provide a script, which, when executed, will create one or more Mastodon client harvesters and any components used to clean/process the data before connecting to and incorporating the cleaned/processed data into the *existing* CouchDB database. You may decide to create a small VM and/or Docker instance to support this process as part of the script. Note that this setup is not intended to create and dynamically deploy the entire application. It is just to demonstrate scaling of the system, e.g., if you wanted to dynamically create and integrate the data from many harvesters. Teams should use Ansible (http://www.ansible.com/home) for this task.
- Teams may wish to utilise container technologies such as Docker, but this is not mandatory.
- The server side of your analytics web application may expose its data to the client through a ReSTful design. Authentication or authorization is NOT required for the web front end.

Teams are also encouraged to describe:

- How fault-tolerant is your software setup? Is there a single point-of-failure?
- Can your application and infrastructure dynamically scale out to meet demand?

Deadline

One copy of the team assignment is to be submitted through Canvas. The zip file must be named with your team, i.e. *<CCC2021-TeamN>.zip*.

Individual reports describing your role and individual team member contributions should be submitted a Qualtrics link that will be distributed in due course. These individual reports will be completion of web-based forms, i.e., they do not require Word/PDF documents etc.

The deadline for submitting the team assignment is **Monday 22nd May (by 12 noon!)**. Note that this is a hard deadline as we are almost at the end of the course!

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 of the Cloud-based solution with dynamic scaling – **25% marks**
- A working demonstration of toot harvesting, use of the Twitter corpus and CouchDB utilization for specific analytics scenarios – **25% marks**
- Detailed documentation on the system architecture and design **– 20%**
- Report and write up discussion including pros and cons of the UniMelb Research Cloud and supporting twitter data analytics – **20% marks**
- Proper handling of errors – **10% marks**

The (confidential) assessment by your peers in your team on the Qualtrics system will be used to weight your individual scores accordingly. Timeliness in submitting the assignment in the proper format is important. **A 10% deduction per day will be made for late submissions.**

Demonstration Schedule and Venue

The student teams are required to give a presentation (with a few slides) and a demonstration of the working application. The presentation should include the key data analytics scenarios supported as well as the design and implementation choices made. Each team has **up to 15 minutes** to present their work. **This will take place on:**

- **9am-10am 23rd May (4 teams present)**
- **4.15pm-5.15pm 23rd May (4 teams present)**
- **5.15pm-6.15pm 23rd May (4 teams present)**
- **3.15-5.15pm 24th May (8 teams present)**

Note that given the numbers of teams this year, not all teams will be able to present – however all teams should be prepared to present. I will advise on Canvas how the randomised selection process will be arranged.

As a team, you are free to develop your system(s) where you are more comfortable with (at home, on your PC/laptop, in the labs...) but obviously the demonstration should work on the UniMelb Research Cloud.