Cluster and Cloud Computing Assignment 2
Global Twittering

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.

Assignment Description

The software engineering activity builds on the lecture materials describing Cloud systems and especially the NeCTAR Research Cloud and its use of OpenStack; on the Twitter APIs, and CouchDB and the kinds of data analytics (e.g. MapReduce) that CouchDB supports. 28 teams have been allocated a given English-speaking, global city (in alphabetical order *Adelaide, Atlanta, Birmingham, Boston, Brisbane, Chicago, Dallas, Detroit, Dublin, Edinburgh, Glasgow, Houston, London, Los Angeles, Melbourne, Miami, Montreal, New York, Perth, Philadelphia, Phoenix, Singapore, San Antonio, San Diego, San Francisco, Sydney, Toronto, Washington*). Assignment of teams to cities has already been made in the lecture through a random and transparent process. The task is to harvest as many tweets as possible from "your" cities[1] on the NeCTAR Research Cloud and undertake a variety of social media data analytic scenarios that tell interesting stories of life in your cities.

The teams should develop a Cloud-based solution that exploits a multitude of virtual machines (VMs) across the NeCTAR Research Cloud for harvesting tweets through the Twitter APIs (both Streaming and the Search API interfaces). The teams should produce a solution that can be run (in principle) across any node of the NeCTAR Research Cloud to harvest and store tweets. Teams have been allocated four medium sized VMs with 8 cores (32Gb memory total) and up to 250Gb of volume storage and 100Gb of object storage. All students have access to the NeCTAR Research Cloud as individual users and can test/develop their applications using their own (small) VM instances. (Remembering that there is no persistence in these small, free and dynamically allocated VMs).

The solution should include a Twitter harvesting application for their allocated city. The teams are expected to have multiple instances of this application running on the NeCTAR Research Cloud together with an associated CouchDB database containing the amalgamated collection of Tweets from the harvester applications. The CouchDB setup may be a single node or a replicated setup. A key aspect of this is in removing duplicate tweets, i.e. the same tweet captured more than once by a Cloud image instance, or designing the system in such a way that duplicate tweets will not arise. Teams may use GeoCouch (the spatial index for CouchDB) for displaying the tweets and results of the analysis or GoogleMap APIs.

Teams are also expected to develop a range of analytic scenarios, e.g. using the MapReduce capabilities offered by CouchDB for their allocated city. All teams **must** support sentiment analysis of their city, e.g. searching for tweets containing positive sentiments (happy, ecstatic, …), negative sentiments (unhappy, terrible, …) or emoticons like ;o), :o), :), or :o(, >:o(etc and establishing whether people are happier in the morning or in the night time or if there are parts of their cities that are happier than others. Teams **should** actively explore more advanced solutions for sentiment

[1] Noting that Twitter can/will block users from downloading too much data at any given point – hence your application/team must be mindful of these limitations and ensure that you don't exceed your quota. It is suggested that the application focuses on periodically harvesting a limited number of tweets from a given area. Every city will provide enough data for your analysis, so don't think by having New York, you will have more data or any advantage than say Dublin.

analysis rather than simple term searching, e.g. *not happy* is a negative sentiment. In addition to this sentiment analysis scenario, teams should explore other scenarios based on their cities. 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:

- Who is the most prolific tweeter? Who has the most followers? Who has been re-tweeted the most? Which person/organization is liked/disliked the most? Which tweeter has travelled the most according to the locations (latitude/longitude) of the tweets they have made? How tweets can spread in space and time, e.g. like a rumour?
- The different languages used when tweeting in the team's city and whether certain cultures (as given by the language their device has been set up for, e.g. en = English; it = Italian etc) are generally more positive or negative?
- Comparing public interest in sports for their city and how sentiment may change with wins/losses and/or position in the league table for any given sports team?
- Analysis of tweets related to topical themes, e.g. what people think of politicians such as Barack Obama or Tony Abbott or David Cameron,
- or combinations of these and other scenarios.

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 a city. Teams are encouraged to combine twitter data with other data of relevance to the city, e.g. information on weather, sport events, TV shows, visiting celebrities, stock market rise/falls etc. For Australian city teams, the AURIN system (https://portal.aurin.org.au) offers a rich source of data about the Australian cities and can be used.

The result of the assignment will be a fully populated *instance* (singular) of a CouchDB for their international city together with a range of data analytics stories associated with the selected cities.

For the implementation teams are recommended to use a commonly understood language across team members – most likely Java or Python. Information on building and using Twitter harvesters can be found on the web, e.g. see https://dev.twitter.com/ and related links to resources such as Tweepy and Twitter4J.

Error Handling

Issues and challenges in using the NeCTAR Research Cloud for this assignment should be documented. You should describe in detail the limitations of mining twitter content and language processing (e.g. sarcasm). You should outline any solutions developed to tackle such scenarios. Removing duplicates of tweets should be handled. The database may however contain re-tweets. You should demonstrate how you tackled working within the quota imposed by the Twitter APIs through the use of the Cloud. You should describe how your system was designed to be robust and provide any degrees of fault tolerance.

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, e.g. based on the tweets particular to that city. Teams are encouraged to write this report in the style of a paper than can ultimately be submitted to a conference / journal. An example of such a paper is on the LMS.

Each team member is also 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 15-20 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 of Tweet analysis and snapshots of the web apps/maps displaying certain Tweet 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 NeCTAR Research Cloud and tools and processes for image creation and deployment.

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 handed in separately.

Implementation Requirements

Teams are expected to use:

- a version-control system such as BitBucket or GitHub for sharing source code.
- MapReduce based implementations for analytics where appropriate, using CouchDB's built in MapReduce capabilities. You may also use Hadoop for this task if desired.
- The entire system should have scripted deployment capabilities. This means that your team will provide a script, which, when executed, will create and deploy the virtual machines and orchestrate the set up of all necessary software on said machines (e.g. CouchDB, the twitter harvesters, web servers etc.) to create a ready-to-run system. Note that this setup need not populate the database, but demonstrate your ability to orchestrate the necessary software environment on the NeCTAR Research Cloud. Teams should use Ansible (http://www.ansible.com/home) for this task.
- The server side of your analytics web application must expose its data to the client through a ReSTful design. Teams must demonstrate that the ReSTful API is functional via a web browser or through the command line (Authentication or authorization is NOT required).

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

The team assignment submitted to the lecturer through the LMS. The zip file must be named with your City, i.e. *<London>.zip*.

Individual reports describing your role and your teams contributions should be submitted separately. Your individual report should be named <*City-Student Id*>.zip, e.g. *<London-123456>.zip* indicating which Team/city you were involved in. These will be submitted by the PRAZE system on the LMS.

The deadline for submitting the team assignment is: **Monday 13th May (by 1pm!)**.

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 deployment – **25% marks**
- A working demonstration of tweet harvesting and CouchDB utilization for specific city analytics scenarios – **25% marks**
- Detailed documentation on the system architecture and design **– 20%**
- Report and write up discussion including pros and cons of the NeCTAR Research Cloud and supporting twitter data analytics – **20% marks**
- Proper handling of the errors and removal of duplicate tweets – **10% marks**

The (confidential) assessment by your peers in your team 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. This should include the key Twitter analytics scenarios supported as well the design and implementation choices made. Each team has **up to 15 minutes** to present their work. **This will take place on Wednesday 14th May (14 teams present) and 21st May (14 teams present).** A schedule for demonstrations for each of the teams will be drawn up in due course.

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 NeCTAR Research Cloud.

Appendix – Randomly Selected Teams

The LMS includes a Spreadsheet (under the Lectures tab) of the team membership and importantly the randomly assigned team – city pairings. For final confirmation and to remove any further ambiguity, the team-city pairings are:

Team1 = Edinburgh
Team2 = Detroit
Team3 = London
Team4 = Melbourne
Team5 = Miami
Team6 = Dallas
Team7 = Los Angeles
Team8 = Chicago
Team9 = Adelaide
Team10 = New York
Team11 = Glasgow
Team12 = Houston
Team13 = Phoenix
Team14 = Dublin
Team15 = Montreal
Team16 = Atlanta
Team17 = Birmingham
Team18 = Brisbane
Team19 = Boston
Team20 = Philadelphia
Team21 = Singapore
Team22 = Perth
Team23 = San Francisco
Team24 = San Diego
Team25 = Toronto
Team26= Sydney
Team27 = Washington
Team28 = San Antonio

#Good luck! LoL! ;o)