UNIVERSITY OF OXFORD
SOFTWARE ENGINEERING PROGRAMME

Wolfson Building, Parks Road, Oxford OX1 3QD, UK
Tel +44(0)1865 283525 Fax +44(0)1865 283531
info@softeng.ox.ac.uk www.softeng.ox.ac.uk

Part-time postgraduate study in software engineering

Cloud Computing and Big Data, CLO

8th – 12th July 2019

ASSIGNMENT

The purpose of this assignment is to test the extent to which you have achieved the learning objectives of the course. As such, your answer must be substantially your own original work. Where material has been quoted, reproduced, or co-authored, you should take care to identify the extent of that material, and the source or co-author.

Your answers to the questions on this assignment should be submitted using the Software Engineering Programme website — `www.softeng.ox.ac.uk` — following the submission guidelines. When submitting the assignment online, it is important that you formally complete all three assignment submission steps: step 1, upload your files; step 2, check your files; step 3, read through the declaration, and click the **I Agree** button followed by the **Submit now** button. The deadline for submission is 12 noon on Tuesday, 27th August 2019.

We hope to have preliminary results and comments available during the week commencing Monday, 7th October 2019. The final results and comments will be available after the subsequent examiners' meeting.

ANY QUERIES OR REQUESTS FOR CLARIFICATION REGARDING THIS ASSIGNMENT OR PROBLEMS INSTALLING SOFTWARE SHOULD, IN THE FIRST INSTANCE, BE DIRECTED TO THE PROGRAMME OFFICE WITHIN THE NEXT TWO WEEKS.

CLO Module Assignment July 2019

Introduction
The assignment is designed to allow you to demonstrate knowledge of systems, processes and approaches for dealing with big data in the cloud.

You must show a good understanding of big data methodologies, including the ability to design big data systems in the cloud. You must also show the ability to create applications and systems that can process big data.

Assessment objectives
This assignment is being assessed. Like other modules, you will be assessed dependent on demonstrating certain things.

In this case the main criteria for a good grade is that you can create a system that processes a large dataset in parallel, and that you can understand and concepts, principles and approaches for designing reasonably complex systems.

In particular:

- Have you understood the principles and design characteristics of a big-data architecture?
- Can you implement and design simple analytics to run big-data jobs? Can you design a system that scales in the cloud? Are you able to define and design big-data analytics systems that support real-time and batch data?
- Have you addressed high-availability, reliability and failover in your design?
- Do you understand the challenges, emerging work and tradeoffs between different approaches? In particular, can you articulate clearly why different big data and cloud technologies are better or worse for certain tasks?

Domain – what is the problem?

The Chicago city authorities publish a CSV file containing a record of every journey undertaken in a taxi dating back to 2013. The assignment will be about handling this data as well as designing a system to handle batch and real-time taxi data.

There are three parts to this challenge:

1. Analysing a large data set to produce specific answers about that data.

2. Running that analysis in a cloud environment to understand scaling characteristics of the data processing system chosen.

3. Designing (but not implementing) a production system to handle ongoing batch and real-time processing of similar data.

In parts 1 and 2, weight will be given to seeing the outputs of your data processing system. You must show that you have created a system that runs in parallel on more than one cloud node. However the examiners will NOT deploy, install or test any code you write or system you build. The only submission should be a single document with appendices.

Please do not submit any code other than included in direct answer to a question or in an appendix.

Part 1. Data analytics *(overall weight - 35%)*

The data that is provided for the assignment is taxi data from Chicago from 2013 onwards. The data includes the fares, distance, time, locations and other interesting data.

The data format is a CSV file, and there are two different files.
The first is a full dataset, while the second is a partial subset (sampled).

It is recommended that you use the partial data set to test your system. Once the system is working you should then run the analysis on the full dataset.

The dataset is approximately 69Gb (unzipped). The file is encoded with GZip (gz) to save on storage space (17.3Gb) and is available at:

https://chictaxi.s3-eu-west-1.amazonaws.com/chictaxi.csv.gz

The S3 URL is:

s3a://chictaxi/chictaxi.csv.gz

There is a sampled subset of the data available, which is approximately 1/2000th of the size (35Mb unzipped, 11.7Mb gzipped).

This dataset is available at:

https://chictaxi.s3-eu-west-1.amazonaws.com/small.csv.gz

or

s3a://chictaxi/small.csv.gz

Both files have a header line at the top with the field names.

The data has the following format:

Column Name	Description	Type
Trip ID	A unique identifier for the trip.	Plain Text
Taxi ID	A unique identifier for the taxi.	Plain Text
Trip Start Timestamp	When the trip started, rounded to the nearest 15 minutes.	Date & Time
Trip End Timestamp	When the trip ended, rounded to the nearest 15 minutes.	Date & Time
Trip Seconds	Time of the trip in seconds.	Number
Trip Miles	Distance of the trip in miles.	Number
Pickup Census Tract	The Census Tract where the trip began. For privacy, this Census Tract is not shown for some trips. This column often will be blank for locations outside Chicago.	Plain Text
Dropoff Census Tract	The Census Tract where the trip ended. For privacy, this Census Tract is not shown for some trips. This column often will be blank for locations outside Chicago.	Plain Text
Pickup Community Area	The Community Area where the trip began. This column will be blank for locations outside Chicago.	Number
Dropoff Community Area	The Community Area where the trip ended. This column will be blank for locations outside Chicago.	Number
Fare	The fare for the trip.	Number
Tips	The tip for the trip. Cash tips generally will not be recorded.	Number
Tolls	The tolls for the trip.	Number
Extras	Extra charges for the trip.	Number
Trip Total	Total cost of the trip, the total of the previous columns.	Number
Payment Type	Type of payment for the trip.	Plain Text
Company	The taxi company.	Plain Text
Pickup Centroid Latitude	The latitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Number
Pickup Centroid Longitude	The longitude of the center of the pickup census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Number
Pickup Centroid Location	The location of the center of the pickup census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Point
Dropoff Centroid Latitude	The latitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Number
Dropoff Centroid Longitude	The longitude of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Number
Dropoff Centroid Location	The location of the center of the dropoff census tract or the community area if the census tract has been hidden for privacy. This column often will be blank for locations outside Chicago.	Point

You need to create parallelizable big data analysis programs to answer the following questions.

You must demonstrate that you are doing these calculations in a way that utilizes either multiple computers or multiple threads.

Please show the results for the full dataset.

1) How many records would you classify as bad? We could consider a bad record to be one where the trip is not voided, but also if there is an anomalous speed, distance or cost. Justify your approach. Once you have defined this, ensure that all further answers are based only on good data.

2) How many records are there? How many records for each year of the dataset?

3) For each taxi, calculate the average revenue per day excluding tolls (i.e. Fare + Tips). Identify the most successful taxi in 2017 in terms of total revenue (Fare + Tips).

4) Taking 1 hour periods throughout the day (from midnight to midnight) across the complete dataset:
    a. What is the average speed of taxis during each period?
    b. Which is the period where drivers in total earn the most money in terms of fares?
    c. Which is the period of the day where drivers in total earn the most in tips?

    Where a trip crosses a boundary (where the drop off is in a different period to the pickup), assign that trip to the period which it is in more. If the trip exactly straddles two periods then assign it to the earlier period. If a trip crosses more than two boundaries, assign it to the period where the midpoint of the journey happened.

5) What is the overall percentage of tips that drivers get? Find the top ten trips with the best tip per distance travelled. Create a graph of average tip percentage by month for the whole period.

6) Using 2017 data only, and using a clustering algorithm, identify the best 5 locations for a driver to start a trip on a weekday evening between 5-7pm. Plot these on a map.

    Identify the best 5 locations for Saturday nights at 10-11pm. Plot these.

7) Using commonly available libraries for location to zipcode mapping, identify the most common pair of zipcodes to start/stop a journey in.

You must show the code that you wrote to calculate each of these measures and some of the log output from each interaction, either in the body of the document or in appendices.

Part 2 - Scaling *(overall weight - 15%)*
Create a single script that runs the analysis of questions 1-5 (excluding graphs) and time how long it takes to run on 1, 2, 4, and 8 processors. Please include the script in an appendix or main document. You must show scaling across different connected machines (e.g. 4 servers each with 2 cores to demonstrate 8 processors). You may use a subset of the data (at least 1Gb) or the whole dataset.

Show enough log to demonstrate that this has successfully run across multiple machines and processors.

Graph the time taken against the number of processors and identify the Karp-Flatt metric for this workload on the system you have chosen to implement on, using the 1 and 8 processor numbers.

For parts 1 and 2, there are no word limits, but please use one or more appendices wisely so that the main document is not just a long listing of code and logs!

Part 3. Designing a big data analysis system to be deployed in the cloud
(overall weight - 50%)

Please design a big data processing system to process this taxi data in the cloud on an ongoing basis. The system should support analysis including:

- batch (regular) queries,
- ad-hoc querying by data scientists, and
- real-time analysis with alerting.

In addition to this publicly available data, assume that the authorities collect the specific driver and car of each trip as well.

Assume that there are a number of batch reports to be created that allow the authorities to analyse each driver and car as well as providing overall information on each area, tolls, routes, etc.

In addition, the system needs to be able to calculate live data, such as who has drive the most distance in the last 1 hour, 4 hours, 24 hours (all rolling). The authorities would like to be able to identify busy periods, when drivers are under-occupied vs overly busy, as well as tracking if drivers or cars are getting sufficient downtime.

You should provide a clear outline of your preferred design for solving this, and make sure that you cover issues including:

- Please draw an architecture diagram showing the resulting system.
- How the data is ingested into the system and stored?
- Which big data framework or algorithms are you going to choose? Why?
- How does your chosen approach efficiently process the data?
- Which cloud infrastructure and why?
- How are you going to scale this?
- Which language(s) are you going to use to process the data and why?
- How can you monitor and maintain this system?
- How can the system handle real-time analysis and alerting?
- How you would add new datasets and queries to the system?

Do not implement this system! This is a design exercise only. Please do not include code for any part of this system.

Please ensure that for each design decision that you make, outline the reasons for this decision and why you took that approach.

Word limit 1500 - 2500 words

Final thoughts

- You are not expected to completely implement a production system! A real-life solution is out-of-scope.
- The examiners will not install any code you write. Only the document is being assessed.
- Clearly document any assumptions you make.
- You (and the examiner) must be confident that there are no major flaws in the design and that it is implementable.
- Properly formatted references/a bibliography will be appreciated.

Derivative works

You may be able to find partial solutions to the problems on the web. It is strongly recommended that you *do not* use them: they are likely to implement slightly different specifications, they typically won't help your critical review, and they won't help you learn about cloud computing and Big Data. More importantly, the assessors will not be able to judge your understanding of this domain.

You must make clear the source and the extent of any derivative material, and reference any works that you used to help you clearly.