Queen Mary
University of London

**BSc Examination by course unit**

***xx$^{th}$ May 2017***

***ECS640 Big Data Processing Duration: 2 hours 30 minutes***

**YOU ARE NOT PERMITTED TO READ THE CONTENTS OF THIS QUESTION PAPER UNTIL INSTRUCTED TO DO SO BY AN INVIGILATOR**

## Answer ALL FOUR questions

Calculators are not permitted in this examination. Please state on your answer book the name and type of machine used.

Complete all rough workings in the answer book and cross through any work that is not to be assessed.

Possession of unauthorised material at any time when under examination conditions is an assessment offence and can lead to expulsion from QMUL. Check now to ensure you do not have any notes, mobile phones or unauthorised electronic devices on your person. If you do, raise your hand and give them to an invigilator immediately. It is also an offence to have any writing of any kind on your person, including on your body. If you are found to have hidden unauthorised material elsewhere, including toilets and cloakrooms it will be treated as being found in your possession. Unauthorised material found on your mobile phone or other electronic device will be considered the same as being in possession of paper notes. A mobile phone that causes a disruption in the exam is also an assessment offence.

**EXAM PAPERS MUST NOT BE REMOVED FROM THE EXAM ROOM**

**Examiners: Félix Cuadrado, Graham White**

**Question 1 (25 marks]**

a) You have a dataset of Call Detail Records (CDRs) from a mobile operator that registers whenever a mobile phone user has engaged into any activity within the mobile network. Each CDR in the dataset is contained in one line with the following fields:

**`[PhoneNumber, BaseStationID, Timestamp, CDRType<Connect,Call,SMS>]`**

CDRType is a String that can take three different values : **`"Connect"`**, when the mobile phone gets reception for the first time from a specific Base Station; **`"Call"`**, when a phone call is sent or received; and **`"SMS"`**, when a Short Message is sent or received in the phone.

Design a MapReduce program that obtains the total number of users that connected to each base station on two given days (in order to aid with comparisons of large demonstrations of people across the same areas).

Use pseudocode for the program specification. You must clearly define the input and output of each one of your functions. State in your solutions any assumptions that are made as part of the program, as well as the behaviour of any custom function you deem necessary.
The code flow must be **explained**, discussing the input and ouput of each function that has been defined. You may use a diagram to illustrate the overall data flow.

**[16 marks]**

b) Suppose the program created in 1.a) will be executed on a dataset of 1 billion CDRs. 25% of the CDRs register Connect actions, 25% Call interactions, and 50% SMS interactions. The CDR file has a size of 128 GB. Which element of Hadoop is in charge of deciding how many resources will be used in order to execute this job? Explain how many worker nodes will be involved in the job, assuming the cluster has a very large number of free nodes, and HDFS is configured with a block size of 128MB. State any assumptions you feel necessary when presenting your answer.

**[9 marks]**

**Question 2 (25 marks]**

a) You have a dataset from geopolitical events collected by the GDELT open Data Service. The data reports analysis of global events as collected by news and press outlets. The format of the dataset is one line for each event, following the format

**[EventId,timestamp,country,CAMEOCode,peopleCount(optional)].**
An example reading would be*:*
***[Washington,20170121,USA,1413,500000].***

**CAMEOCode** categorises the type of event (e.g. ***1413*** is the category for "Demonstrate for rights" events). **peopleCount** provides an estimate of the number of people involved in the event.

You have the following Map/Reduce program in pseudocode, which takes as input the just described GDELT dataset.

```
Map (String eventId, Data event)
If(data.getPeopleCount()>1000)
     emit(event.getCountry(), new Pair(eventId, data.getCount())
Reduce (String country, List<Pair[String,int]> events)
List sortedEvents = sorteventsByCountAndGetTop5(events);
For(event: sortedEvents){
     emit (country, event)
}
```

Note that in the pseudocode a Pair object represents a tuple of two elements. The Boolean field isCheckIn has a true value when the user enters the station, and false when the user leaves the station.

What will be the outcome of the Map/Reduce job? Explain the outcome as well as the nature of information exchanged in the job execution.

**[8 marks]**

b) Define a Combiner in the context of MapReduce jobs. Explain what would be the performance impact of adding a Combiner to the previous Map/Reduce job.

**[8 marks]**

c) HDFS is the component of Hadoop in charge of data storage.

i. Describe two characteristics of HDFS that provide an advantage when running Hadoop jobs.

**[5 marks]**

ii. When an external client wants to read a file stored in HDFS, describe who receives the request, and who attends the read request.

**[4 marks]**

**Question 3 (25 marks]**

a) The majority of processing in Spark is performed with transformations such as map(), filter, or reduceByKey().
Answer the following questions regarding the behaviour of such transformations when running Spark jobs.

i. Explain in which nodes are Spark transformations executed.

**[3 marks]**

ii. Define the concept of deferred execution of Spark transformations. Explain when are Spark transformations executed, providing the rationale for that timing.

**[3 marks]**

iii. Explain what is the outcome of a Spark transformation.

**[2 marks]**

iv. Discuss whether a single Spark transformation such as the ones mentioned before is more or less expressive (i.e. they can express more types of computation) than one pair of Map and Reduce functions.

**[3 marks]**

b) The following code snippet shows a Spark job written in Scala.

```
val lines = sc.textFile("hdfs://inputPath")

val words = lines.flatMap(lines => lines.split("\\s") )
val hashtags = words.filter(word => word.startsWith("#"))
val results = hashtags.map(word => (word, 1)).reduceByKey((a,b)=>a+b)

counts.saveAsTextFile("hdfs://outputPath ")
```

We provide for reference the following information about the Spark-specific functions appearing in the program:
**`filter`** is a transformation that for each input element will either generate the same element as output if the condition expressed in the function is true, or no element at all if it is false.
**`map`** is a transformation that for each input element generates one output element, whose value is dictated by the function.
**`flatMap`** is a transformation that for each input elements generates a variable number of output elements, extracted from an output collection generated by the provided function.
**`reduceByKey`** is a transformation that takes as input an RDD of pairs of elements, first groups all the pairs, putting together the ones with the same key, and generates one pair as a result of the group. The key of the result is the common key, and the value is the result of reducing the group of values into a single one by applying the function.
**`saveAsTextFile`** is an action that saves the refered object to the provided HDFS path.

i. Explain the result obtained by the program when executed over an input dataset formed by a collection of Tweets. The format of each line is identical to the one from the Rio2016 dataset from coursework 1, namely:

```
epoch_time;tweetId;tweet_text;device
```

**[2 marks]**

ii. Compare the Spark program with an equivalent set of Hadoop jobs. In order to do so, first discuss the number of Hadoop jobs necessary to perform the equivalent computation. Compare the expected performance in both platforms, in case there would be any significant difference. Assume both jobs take the data from the same HDFS folder.

**[5 marks]**

c) Explain the relationship of parallel computation speedup with the number of processors utilised. Use the concepts of Amdahl's Law to illustrate your explanation.

**[7 marks]**

**Question 4 (25 marks]**

a) Many graph algorithms such as Single Source Shortest Path, or Pagerank are executed in parallel for processing very large graphs.
Explain why the implementations of these algorithms in parallel require iterations. Your explanation should also detail what is the unit of data being parallelised in these cases. Discuss what qualities are required in a batch processing platform in order to support the efficient execution of such graph algorithms.

**[8 marks]**

b) Data partitioning is a technique utilised in many data-intensive parallel computing systems.
Compare the concept of data partitioning for general MapReduce jobs with data partitioning for large-scale graph processing jobs. For each type of job, you should define what elements are being partitioned, the reason behind the partitioning, as well as the implications for job execution.

**[7 marks]**

c) This question is about Stream processing systems.

i. Explain whether large-scale stream processing systems achieve low or high throughput, as well as low or high latency in comparison with batch processing systems.

**[4 marks]**

ii. Stream processing computing components (e.g. Storm bolts) process one piece of data at a time. Explain how can a stream component keep track of past elements (e.g. in order to perform a count).

**[3 marks]**

iii. Define the sliding window technique of stream processing components. Illustrate its application with an example.

**[3 marks]**

**End of Paper**