Birkbeck

(University of London)

MSc EXAMINATION

Department of Computer Science and Information Systems

Cloud Computing (BUCI029H7)

CREDIT VALUE: 15 credits

Date of examination: Friday, 2nd June 2017
Duration of paper: 2:30pm – 4:30pm (2 hours)

RUBRIC

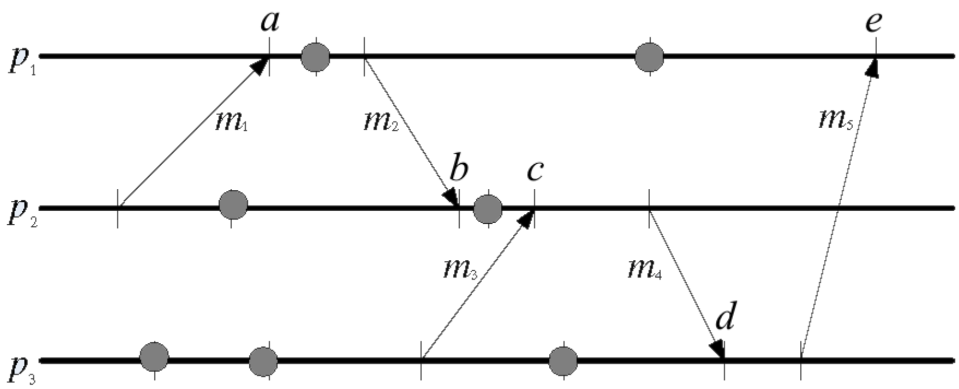
1. *This paper contains* five *questions for a total of* 100 *marks.*
2. *Students should attempt to answer* ***all*** *of them.*
3. *This paper is not prior-disclosed.*
4. *The use of non-programmable electronic calculators is permitted.*

©Birkbeck College 2017

1. **(20 marks)**

Give brief answers (in a few sentences) to the following questions.

(a) What are the Lamport timestamps of the events a, b, c, d and e respectively in the following space-time diagram? (5 marks)



(b) What is the difference between crash failure and Byzantine failure? Which one is more disruptive? (5 marks)

(c) What is eventual consistency? Why don't we insist on strong consistency in all distributed systems? (5 marks)

(d) In RESTful APIs, what do the constraints "stateless" and "cacheable" mean respectively? (5 marks)

2. **(20 marks)**

Give brief answers (in a few sentences) to the following questions.

(a) What are the pros and cons of the "stripes" design pattern compared with the "pairs" design pattern? (5 marks)

(b) What is the "order inversion" design pattern used for? What is the "value-to-key conversion" design pattern used for? (5 marks)

(c) What is the prerequisite for the usage of map-side join? What is the prerequisite for the usage of in-memory join? (5 marks)

(d) Why is MapReduce inefficient for complex iterative applications and interactive queries? What is the Discretized Stream (D-Stream) model of Spark? (5 marks)

3. **(20 marks)**

There is a large text file that contains all tweets about BBC programmes collected from Twitter in 2011–2015. It is stored in an HDFS over a number of machines.

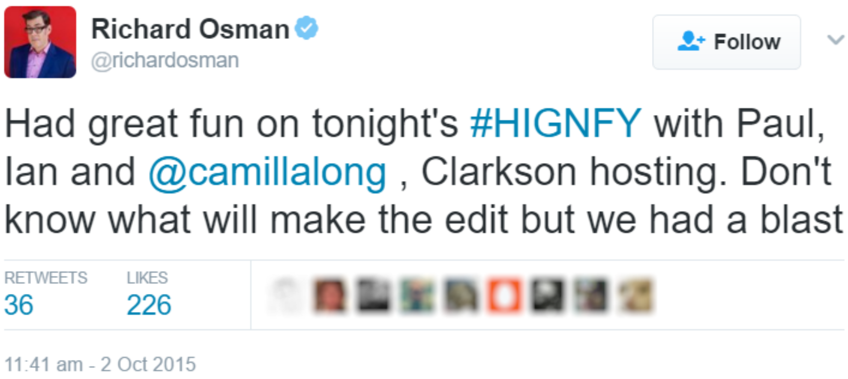
Each line of this file describes one tweet in the following format, where the different fields are separated by the | character (assuming that it does not occur in any tweet).

$user \mid text \mid retweets \mid likes \mid date$

For example, the line

> richardosman | *Had great fun on tonight's #HIGNFY with Paul, Ian and @camillalong , Clarkson hosting. Don't know what will make the edit but we had a blast* | 36 | 226 | 02/10/2015

represents the tweet as shown in the following figure.



Richard Osman
@richardosman

Had great fun on tonight's #HIGNFY with Paul, Ian and @camillalong , Clarkson hosting. Don't know what will make the edit but we had a blast

RETWEETS 36 LIKES 226

11:41 am - 2 Oct 2015

A tweet could contain hashtags (such as "*#HIGNFY*") and mention usernames (such as "*@camillalong*") in its text.

Write a MapReduce program (in pseudo-code) to calculate for each user the average number of likes of his/her tweets in 2015.
A combiner should be implemented to accelerate the computation.

4. **(20 marks)**

Consider the same large data file as described in the previous question.

Write a MapReduce program (in pseudo-code), using the "pairs" pattern, to calculate for each hashtag the number of co-occurrences with another hashtag if they have occurred in the same tweet before. For example, the output corresponding to the hashtag *#HIGNFY* could be as follows.

#HIGNFY, #brexit: 300
#HIGNFY, #trump: 200
#HIGNFY, #TransportforLondon: 100

The "in-mapper combining" pattern should be implemented to accelerate the computation.

5. **(20 marks)**

Suppose that a *directed* graph is stored as a file of adjacency lists (in the HDFS) as follows: each line of the file is in the format "$u : v_1, v_2, \ldots, v_k$" denoting that there is an outgoing link from node u to node v_i ($1 \leq i \leq k$), where u and v_i are integer node IDs.

$[NB]$ The graph has too many nodes to be loaded into the memory of any single machine.
$[NB]$ It is not required to use combiners or in-mapper combining.
$[Tip]$ More than one MapReduce job could be used to accomplish the task.

(a) Write a MapReduce program (in pseudo-code) to find the node with the maximum in-degree (i.e., the number of incoming links). (10 marks)

(b) Write a MapReduce program (in pseudo-code) to augment each node u's adjacency list with all the nodes reachable in two steps from u. In other words, if there is a link $u \to v$ and also a link $v \to w$, we shall add to the graph a link $u \to w$ unless it already exists. There should be no duplicate nodes in an adjacency list. (10 marks)