Birkbeck

(University of London)

MSc EXAMINATION

Department of Computer Science and Information Systems

Cloud Computing (BUCI029H7)

CREDIT VALUE: 15 credits

Date of examination: Tuesday, 27th May 2014
Duration of paper: 2:30pm – 4:30pm (2 hours)

RUBRIC

1. *This paper contains* 5 *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.*

1. **(20 marks)**

Please give brief answers to the following questions.

(a) In Amazon Web Services (AWS), what do AMI, EC2, S3, EBS, EMR stand for respectively? (5 marks)

(b) In RESTful APIs, what HTTP methods should be nullipotent and what HTTP methods should be idempotent? (5 marks)

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

(d) What are the possible disadvantages of using a private cloud instead of a public cloud? (5 marks)

2. **(20 marks)**

Please give brief answers to the following questions.

(a) What are the 5Vs of Big Data? (5 marks)

(b) Which three properties cannot be provided by a distributed system simultaneously, according to the CAP theorem? Which one of them do NoSQL databases usually compromise in favour of the other two? (5 marks)

(c) What are the underlying assumptions of the Google File System (GFS)? (5 marks)

(d) Why is the standard hashing technique unsuitable for distributed indexing? How does consistent hashing solve this problem? (5 marks)

3. **(20 marks)**

Consider the following MapReduce program `word-count` which counts the total number of occurrences for each distinct word in a very large document collection.

```
class Mapper
    method Map(docid i, doc d)
        for all term t ∈ doc d do
            Emit(term t, count 1)

class Reducer
    method Reduce(term t, counts [c1, c2, ...])
        sum ← 0
        for all count c ∈ counts [c1, c2, ...] do
            sum ← sum + c
        Emit(term t, count sum)
```

(a) How can we modify the mapper function so that the program computes the document frequency for each distinct word (i.e., the number of documents containing that word)?
Please try to make minimal changes to the existing code. (5 marks)

(b) How can we modify the mapper function and also the reducer function so that the program computes the average document length in this collection?
Please try to make minimal changes to the existing code. (5 marks)

(c) How can we use combiners to accelerate the above program for average document length? (5 marks)

(d) How can we use the "in-mapper combining" design pattern instead of combiners to accelerate the above program for average document length? (5 marks)

4. **(20 marks)**

Consider the following MapReduce program that counts the number of co-occurrences for each word-pair (i.e., the number of times those two words both occur within a certain neighbourhood such as a sentence) in a very large document collection. The "pairs" design pattern has been used.

```
class Mapper
    method Map(docid i, doc d)
        for all term w ∈ doc d do
            for all term u ∈ Neighbours(w, d) do
                Emit(pair (w, u), count 1)          ▷ Emit count for each co-occurrence

class Reducer
    method Reduce(pair p, counts [c1, c2, ...])
        s ← 0
        for all count c ∈ counts [c1, c2, ...] do
            s ← s + c                               ▷ Sum co-occurrence counts
        Emit(pair p, count s)
```

(a) Are we able to use the reducer function directly as the combiner function? Why? Is it usually better to set the number of map tasks larger than the number of computer nodes in the cluster? Why? (5 marks)

(b) How can we use the "stripes" design pattern to re-write the above program? Please try to make minimal changes to the existing code. (5 marks)

(c) What are the pros and cons of the "stripes" design pattern in comparison with the "pairs" design pattern? (5 marks)

(d) Explain in words how the given program with the "pairs" design pattern can be extended to compute the relative frequencies, i.e., conditional probabilities

$$P(b|a) = \frac{\text{count}(a, b)}{\text{count}(a)} = \frac{\text{count}(a, b)}{\sum_{b'} \text{count}(a, b')}.$$

Hint: Make use of the "order inversion" design pattern. (5 marks)

5. **(20 marks)**

Suppose that an undirected graph is described in a file by its edge set as follows: each line of the file is in the format "n_1, n_2" representing an undirected edge between two nodes where n_1 and n_2 are integer node IDs.

(a) Please write a MapReduce program (in pseudo-code) to normalise the given data so that each undirected edge between two nodes n_1 and n_2 appears exactly twice in the file: one as "n_1, n_2" and the other as "n_2, n_1". (5 marks)

(b) Please write a MapReduce program (in pseudo-code) to find all the paths up to length L (say 5) in the graph. The final paths should not include any loop, though it is OK to have duplicate paths. (5 marks)

(c) What is the number of MapReduce job iterations required to compute all the paths up to length L, using your above program? (5 marks)

(d) The input file of undirected edges can be regarded as a relational table with two columns n_1 and n_2.
What are the three possible ways to perform a natural join of the table with itself using MapReduce? Which one of them has the fastest speed? Which one of them has the least restriction? (5 marks)

