IMPERIAL COLLEGE OF SCIENCE, TECHNOLOGY AND MEDICINE

EXAMINATIONS 2016

MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc in Advanced Computing
MSc in Computing Science (Specialist)
MRes in High Performance Embedded and Distributed Systems
MRes in Advanced Computing
for Internal Students of the Imperial College of Science, Technology and Medicine

This paper is also taken for the relevant examinations for the Associateship of the City and Guilds of London Institute

PAPER C410H

SCALABLE DISTRIBUTED SYSTEMS DESIGN

Monday 21 March 2016, 10:00
Duration: 70 minutes

Answer TWO questions

Paper contains 3 questions
Calculators not required

1 a Briefly explain each of the following concepts, and give an example where each could be applied in the context of scalable distributed systems.

i) Soft lease

ii) Coordination service

iii) Data locality

iv) Tail latency

b Bigtable is a distributed storage system for structured data.

i) Describe, with the help of one or more annotated diagrams, the architecture of Google BigTable. Your diagram should show clearly how (i) *read* queries and (ii) *write* queries are handled in BigTable.

ii) Explain why the designers of Bigtable decided not to support *multi-row transactions*.

iii) Describe a proposal for how the architecture of Bigtable could be extended to provide support for multi-row transactions with a *strong* consistency model. Your proposal should introduce as few changes to the BigTable design as possible.

iv) Describe (a) one possible query workload under which your design for supporting multi-row transactions would scale well, and (b) another workload under which it would not scale well.

The two parts carry, respectively, 40% and 60% of the marks.

2a Briefly explain each of the following concepts, and give an example where each could be applied in the context of scalable distributed systems.

i) Quorum

ii) Distributed hash table (DHT)

iii) Vector clock

iv) Distributed transaction

b MapReduce is a scalable data-parallel processing system for compute clusters.

i) Describe, with the help of a diagram, how MapReduce processes large datasets that are stored on a distributed file system.

ii) State the two API functions, including a description and the types of all input and output parameters, that MapReduce uses to specify the computation over the data.

iii) Describe how you would extend the MapReduce system to support the execution of more complex computation over data. Specifically, explain how you would support arbitrarily complex relational SQL queries, which may, for example, include several relational JOINs, over multiple datasets.

The two parts carry, respectively, 40% and 60% of the marks.

3 You work as a software engineer for SOCIALPET, a start-up company that wants to launch a new online social networking website targeted at pets. The relationships between pets are expressed as a large social graph, in which pets are nodes, and edges are friend relationships.

Anticipating future growth, your boss asks you to design a graph analytics system for processing the social graph. The system should be able to process the complete social graph in order to output global iterative statistics, such as identifying the top-k nodes that can act as *trend-setters*, i.e. they are connected to all other nodes with the shortest number of hops in the graph.

The social graph used as the input data should be stored in a distributed file systems, such as GFS or HDFS. You have complete freedom in deciding on the format and the partitioning of the data.

The design of the graph analytics system should satisfy the following requirements:

(R1) The system should be able to process graphs with a billion nodes and a trillion of edges efficiently.

(R2) The system should be incrementally scalable.

(R3) The system should support skewed social graphs, i.e. ones in which the difference between the minimum and maximum number of edges of nodes may be large.

(You should make justified decisions about any other requirements that are left unspecified.)

a Explain why using the existing MapReduce system would not be efficient in this scenario.

b Describe how you would represent the social graph data stored in the distributed file system, and justify your choice.

c Draw a diagram of your *system design*, clearly labelling all distributed components. For each component, explain its operation and justify its function.

d For each of the requirements, (R1)–(R3), explain how your design achieves it.

The four parts carry, respectively, 10% 25%, 35%, and 30% of the marks.