For this assessment you should type your answers into AnswerBook.
Where indicated in the question, you can also upload handwritten work.

TIMED REMOTE ASSESSMENTS 2020-2021

MEng Honours Degree in Mathematics and Computer Science Part IV
MEng Honours Degrees in Computing Part IV
MSc Advanced Computing
MSc in Computing (Specialism)
for Internal Students of the Imperial College of Science, Technology and Medicine

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

PAPER COMP70022=COMP97016=COMP97017

SCALABLE SYSTEMS AND DATA

Wednesday 16 December 2020, 14:00
Duration: 140 minutes
Includes 20 minutes for access and submission

Answer ALL TWO questions
Open book assessment

Paper contains 2 questions

Section A (Use a separate answer book for this Section)

1 a   Pelican, or cold storage systems in general, are designed to be resource limited.

     i)   Explain what resource domains exist (using the example of Pelican) and why the resource domains are limited.

     ii)   Define a simple, yet meaningful strategy to group disks such that they are resource domain disjoint, i.e., two or more disks in the same group can be active at the same time (use two resource domains to simplify the problem and explanation).

   b   OLTP workloads are increasingly executed on novel hardware.

     i)   When executing an OLTP workload, why can the throughput drop as you increase the number of threads executing transactions?

     ii)   What general patterns in the workload can you exploit to increase the throughput? Define one specific strategy.

     iii)   How does a multicore/multisocket architecture affect the execution of an OLTP workload?

   c   VoltDB has been developed for use in main memory with many optimisations for the latter.

     i)   How would you redesign it for use on SSD?

     ii)   How does the choice of storage media affect the horizontal scalability?

     iii)   How are VoltDB's transactions affected if it is used on SSD?

   d   Multiple different approaches exist for storing semi-structured data in databases.

     i)   For what scenario is the model-mapping approach for XML shredding preferable? Conversely, when is structure-mapping to be preferred?

     ii)   How does this compare to document databases (with the example of MongoDB)?

     iii)   Which mapping approach does MongoDB follow? What is the challenge of this mapping approach when using MongoDB?

The four parts carry, respectively, 25%, 25%, 30%, and 20% of the marks.

Section B (Use a separate answer book for this section.)

2   The Dynamo distributed key/value store is deployed on six nodes A–F:
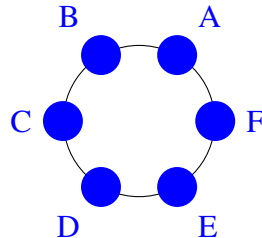
a   Explain why it is important for Dynamo to optimise for low **tail latencies** when handling requests.

b   Describe **three** features of the Dynamo design that help it achieve low tail latencies.

c   The Dynamo instance receives the following sequence of requests for key k from a client at the specified times. (The ‖ symbol indicates that two requests are handled concurrently.)

      $t = 2$: write k on node A
      $t = 4$: read k on node B
      $t = 6$: read+write k on node A ‖ read+write k on node B
      $t = 7$: read+write k on node A ‖ read+write k on node C
      $t = 8$: read+write k on node E

   i)   Write down how the vector timestamps associated with k evolve over time.

   ii)   Explain which request (or requests) above are likely to experience the highest latencies.

d   You want to insert a large number of keys that are part of the same limited key range to Dynamo. This *bulk loading* of keys should have a low impact on the tail latencies of concurrent read/write requests handled by the system.

     Describe an extension of the Dynamo design to enable it to efficiently support the bulk loading of keys. Illustrate your design with the help of a figure.

The four parts carry, respectively, 15%, 15%, 40%, and 30% of the marks.