COMP 18112

Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Distributed Systems

Date: Thursday 18th May 2017

Time: 09:45 - 11:45

Please answer any TWO Questions from the THREE Questions provided.

Use a SEPARATE answerbook for each QUESTION.

This is a CLOSED book examination

The use of electronic calculators is permitted provided they are not programmable and do not store text

[PTO]

1. a) The following two items are about the motivations for distributing systems that were discussed in this course unit.

   i) Recall that one motivation for distributing systems that was discussed was to get customers involved in a company's business processes. Give an example of a class of websites (or web businesses) you are familiar with that achieves this purpose. Your answer must briefly explain the overall idea behind the chosen class of websites but it need not mention any concrete, specific, existing website, though your answer may be clearer if you do. (4 marks)

   ii) The motivation referred to in item (1(a)i) above can be said to be of a functional nature. Briefly define what, in contrast, is meant by a *non-functional* motivation and briefly comment on how this kind of motivation might apply to the class of websites you have chosen in your answer to item (1(a)i), above. (2 marks)

   b) One form of cloud computing is exemplified by servers (for example, Google Docs) that deliver functionality (for example, word processing, spreadsheet, etc.) similarly available in standalone desktop clients (for example, LibreOffice). It could be argued that one motivation for using cloud applications is the fact that most people now use many different devices. In the *specific* context of the examples just given, briefly explain what problems arise from using many different devices in terms of the transparencies discussed in the course, then briefly suggest how a cloud application can better provide the desired transparencies than multiple clients could. (6 marks)

   c) The following four items are about the *architectural paradigms for distributed systems* discussed in this course unit. Assume that a component C and a component C' communicate through message exchange.

   i) Briefly contrast the paradigms known as *direct message-exchange* and *remote procedure call* in terms of message semantics (i.e., the kind of payload, or message content, they carry). (2 marks)

   ii) Briefly explain in what sense the use of *direct message-exchange* requires the timelines of C and C' to synchronize for interprocess communication to take place. (2 marks)

   iii) Briefly contrast the early-binding and the late-binding versions of the remote method invocation (RMI) architectural paradigm. (1 mark)

   iv) Consider the W3C web services architecture and its reliance on service directories. Which of the two forms of the RMI architectural paradigm does that reliance on service directories make possible? (1 mark)

d) Assume that you are a software engineer in a company that is planning a move towards using web-based distributed computing. The technical manager has called for a discussion of the challenges involved. She is worried by the fact that the existing systems were originally designed with a view to being particularly efficient when transferring data from primary to secondary memory. She asks you whether you agree with her that these performance concerns will no longer be as significant in the new systems. State whether you agree with your technical manager and briefly explain why. (4 marks)

e) Assume that a notice board (i.e., a place where people leave messages for other people to see and perhaps respond to). Assume that we equate *sending* to *writing/leaving a message in the board* and *receiving* to *reading a message in the board*. Now, consider the blocking primitives studied in the course unit. State which forms of send and receive used in this notice board scenario are blocking (if any) and which are non-blocking (if any), making sure that you explain the main points of the analogy. (3 marks)

2. a) Assume that, in a setting where the cost of using the interconnect is not negligible, someone in your development team has stated that *"It is always better (in terms of reducing the total elapsed time) to have as many parallel computations as possible."*. Your technical manager turns to you and asks whether you agree with your colleague. State whether you do and briefly give your reasons for holding this view.
(4 marks)

b) A general principle discussed in this course unit states that we can hope to reduce the total elapsed time taken to perform a computation on a collection of data items if we can apply the computation to *partitions* of the collection (i.e., smaller chunks of it) in parallel. Now, assume that there are three machines of equal processing power, and that the input collection was partitioned into three chunks C_1, C_2 and C_3 (where the size of the C_1 chunk is double that of the equal-sized C_2 and C_3 chunks) with a view to scheduling a different chunk (one-to-one) to each machine. Now assume that you are a software engineer and someone in your development team has stated that *"This set-up may not allow us, as seems to be the intention, to do the work in a third of the elapsed time that would be needed in a sequential set-up using only one of the machines."*. Your technical manager turns to you and asks whether you agree with your colleague. State whether you do and briefly give your reasons for holding this view. (4 marks)

c) The following three items are about *massively-distributed processing* in the specific context of MapReduce engines.

i) Briefly define the notion of a *barrier* in concurrent systems, then briefly describe how this notion is used in MapReduce engines. (4 marks)

ii) Briefly describe the negative consequence(s) of using a barrier when there is a high likelihood of imbalanced workloads across mapper and reducer instances.
(4 marks)

iii) In a master-workers approach to parallelization/distribution of load, the system must (somewhere) implement the *split*, *spawn* and *merge* primitive operations that characterize the approach. Now, in the specific context of MapReduce engines, briefly comment on where one can see splitting, spawning and merging taking place. (3 marks)

d) For the next two items, consider the fact that the `traceroute` command (for example, as issued from a Linux shell) sends a packet to a given destination machine and displays the route taken (i.e., the individual hops that make up the path) from the issuing machine to the destination machine along with, for each hop in the route, measurements of the round-trip time (RTT) from the issuing machine to the corresponding node. For example, if you issue the command `traceroute www.facebook.com` from a machine in Manchester, the command will show the route taken for a packet to reach the Facebook web server(s) and, for each hop in the route, three separate readings of the RTT taken by the packet.

i) Assume that you issue the command `traceroute www.facebook.com` from a shell in Manchester *twice*. Briefly explain why you must not expect the route taken to be the same in both cases. You must cast your explanation in terms of the Internet protocol stack as discussed in this course unit. (1 mark)

ii) Again, as in item 2(d)i above, assume you issue (just once this time) the command `traceroute www.facebook.com` from a machine in Manchester. Having done so, you note that, in this case, the RTT for each hop has grown larger the further (in the path) that hop is from the origin. Use one of the axioms of distributed computing studied in this course unit to explain this observation. (1 mark)

e) Assume that you are a software engineer in a company that relies on a bespoke distributed system over the Internet. This system is currently experiencing low latency. A colleague of yours has suggested that this problem can be solved by fixing into the code the optimal paths that message-exchange events should follow. (Assume here that what constitutes an optimal path has been determined from past experience.) The technical manager turns to you and asks you to tell her whether you agree or disagree with your colleague and, in either case, to briefly justify your view in terms of the Internet protocol stack as discussed in this course unit. (2 marks)

f) Consider again the scenario in item 2e above. Would your answer be different if, rather than the Internet, the interconnect were also bespoke (such as in a data centre, of the kind engineered by, for example, Google)? State whether or not it would be, and briefly justify your answer. (2 marks)

3. a) Briefly explain why a *cache* is often used in a distributed system, and give an example of a potential drawback in the use of caches. (2 marks)

   b) Briefly describe how a cache is used and what *specific* positive effect it has in the case of the domain name system (DNS) and in the case of the IMAP protocol. (4 marks)

   c) Briefly define the general notion of an atomic transaction, then, from one or more application areas of your own choosing, give *one example* where the atomicity of a transaction is crucial for the correct outcome of a distributed computation and *another example* where a lack of atomicity does not impact on the correctness of the outcome. (4 marks)

   d) The next two items are about the contrast between stateful and stateless protocols. Assume the following sequence of events:

   - at time t_1, a client C sends an M_1 message to a server S;
   - at time $t_2 > t_1$, S responds to the M_1 message from C with an M_2 message;
   - at time $t_3 > t_2$, C sends an M_3 message to S;
   - at time $t_4 > t_3$, S responds to the M_3 message from C with an M_4 message, then increments its own local counter of M_4 messages sent.

   i) Explain whether, on the evidence given (i.e., the above sequence of events), the message exchange protocol that is being run is a stateless or a stateful one. (3 marks)

   ii) Under the assumption that there may be hundreds of thousands of clients (such as C) concurrently sending messages to an endpoint (such as S), explain why *statefulness* is undesirable, all else being equal. (3 marks)

   e) By appeal to one or more of the Coffman conditions for deadlock, argue that if the only resource (for example, a file) shared by two processes P and P' is updated by both of them, then, in principle, deadlock between P and P' is possible. (3 marks)

   f) Assume that, when using a logical clock system, we have the following relationships between events A, B, C and D in a distributed system:

$$\{A \to C, A \to B, D \to A, C \to D\}$$

   Provide a brief argument that this is (or is not) a consistent set of assertions. (4 marks)

g) Define what is meant by a *critical section* of code and use it to argue whether the statement *"A critical section of code is one that never reads from and writes to the same resource."* is true or false. (2 marks)