Two hours

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Distributed Systems

Date: Monday 21st May 2018

Time: 14:00 - 16:00

Please answer all Questions.

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) Assuming a scenario in which two people in the same building are playing a game with someone in another continent, give *two* of the three reasons that were studied in this course unit as to why coordination in distributed systems is hard, drawing your evidence from the assumed scenario. (2 marks)

   b) Briefly explain in what way a cloud-based web-based office application suite (e.g., Google Docs) exemplifies both functional and non-functional reasons for distributing systems. (2 marks)

   c) Some of the seven kinds of transparency discussed in this course unit can be addressed using protocols in the Internet protocol stack. State the name and briefly define *three* transparencies that *cannot* be addressed by such protocols. You do not need to elaborate or provide examples of scenarios where such transparencies are specifically desirable. (3 marks)

   d) Assume you work for a company one of whose income streams comes from selling snapshots (to, say, media outlets) on behalf of members of the public who upload them to the company's website.

   There are occasions in which the company is flooded with a sudden surge of photos (e.g., in big events, such as football matches or music festivals, etc.) and in a growing number of these the infrastructure has been unresponsive. This is partly because of the time taken to write the photos to a persistent back-end store, and partly because front-end-serving such large photos to be rendered in the browser of a potential buyer is expensive in terms of network bandwidth.

   When this happens, latency worsens and media outlets may not get the very short response times they expect. (They normally go through lots of pictures, which are potentially streaming in in real time, so as to select one or more for publication in their own websites). This could ultimately lead to a fall in sales due to customer dissatisfaction.

      i) Assume now that your technical manager asks you to explain the notion of *elasticity in response to demand* in the specific context of the bottleneck in the company's back-end store. Briefly write up the explanation you would give her. (2 marks)

      ii) Further assume that, in her conversation with you, your technical manager notes that a consultant had suggested that using a data parallel computation could play a fundamental role in addressing the front-end problem. She then asks you to explain what is meant by *data parallel* computation in general. Briefly state the answer you would give her. (2 marks)

      iii) Finally, assume that your technical manager decides to exploit this data parallel approach in order to only serve small thumbnails to the media outlets for browsing, rather than the much larger original photos. She then asks you

to briefly describe the high-level functional components of a data parallel solution for converting large photos into thumbnails as they arrive. Briefly describe the solution you would submit to her. (3 marks)

e) In this course unit, you have learnt about some architectural paradigms for distributed systems. Consider Figure 1. It depicts fourteen distributed components `C1` to `C14`. The following notation is used in the diagram:

- dashed shafts with a triangular arrowhead (such as the ones from `C1` to `C5`, or from `C14` to `C12`) denote requests, i.e., message sends to which a response is expected;
- solid shafts with triangular arrowheads (such as the ones from `C8` to `C5`, or from `C12` to `C14`) denote responses, i.e., message receives that correspond to a request; and
- solid shafts with circular arrowheads (such as the ones from `C4` to `C2`, or from `C13` to `C4`) denote data pushes, i.e., message sends to which no response is expected).

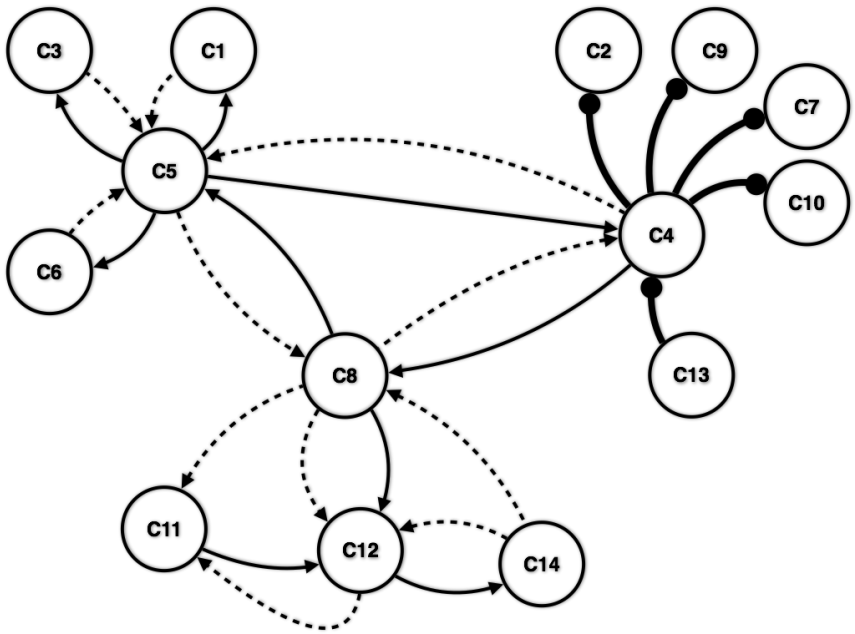

Figure 1:

Now, for each of the clusters defined in parts 1(e)i to 1(e)iii (below), state the architectural paradigm that seems to apply to that cluster, briefly justifying your answer in terms of the pattern and type of messages that are exchanged among its components:

i) {`C1`, `C3`, `C5`, `C6`} (2 marks)
ii) {`C2`, `C4`, `C7`, `C9`, `C10`, `C13`} (2 marks)
iii) {`C8`, `C11`, `C12`, `C14`} (2 marks)

2. a) State which kind of motivation for distributing systems lies behind the modern approach of distributing the workload between CPU and GPU, which is particularly important in gaming. Briefly explain your analysis. (2 marks)

   b) Briefly explain in what way the Bully Algorithm can play a role in the 2-Phase Commit (2PC) protocol. (2 marks)

   c) Briefly explain in what way the use of the Berkeley Algorithm instead of Cristian's Algorithm improves expectations regarding the performance of 2PC. (2 marks)

   d) Using one or more of the axioms of distributed computing discussed in this course, briefly explain why, if those axioms hold and no countermeasures (such as the Internet protocols) are deployed, it is impossible to design an algorithm that allows the two generals in the Two Generals' Problem to safely agree on a time to attack. In your answer, you are free to rely on the *reductio ad absurdum* proof that was sketched in the course notes. (3 marks)

   e) Consider the sequence diagram in Figure 2. It depicts processes `P1` and `P2` and their interactions over time with resources `R1` and `R2`. The following notation is used:

      - The vertical arrows are process and resource timelines, therefore the interactions denoted by the horizontal arrows take place in time sequence starting from the top.
      - The four interaction types, or events, that are represented in the diagram are `request x-lock`, `grant`, `release`, and `wait`. They are denoted by the labels right above the horizontal arrows.
      - The interpretation of the events is assumed to be obvious (where `x-lock` is shorthand for *exclusive lock*, i.e., a mechanism for mutual exclusion over access to the resource that grants the lock).
      - Both network time and processing time are ignored, hence the horizontal arrows and the absence of vertical bars.

      i) State whether the system in Figure 2 will reach the end of the sequence shown or will deadlock before it gets there. (1 mark)

      ii) Again with reference to Figure 2, for each of the four Coffman conditions that must hold for deadlock to occur, explain whether it holds or not in this diagram at the point in time marked by the circled letter `T` to the left. Note that you do not need to write down a definition for each Coffman condition, you only need to refer to each one by the name it was given in this course unit. Make explicit any assumptions you need in your argument. (2 marks)

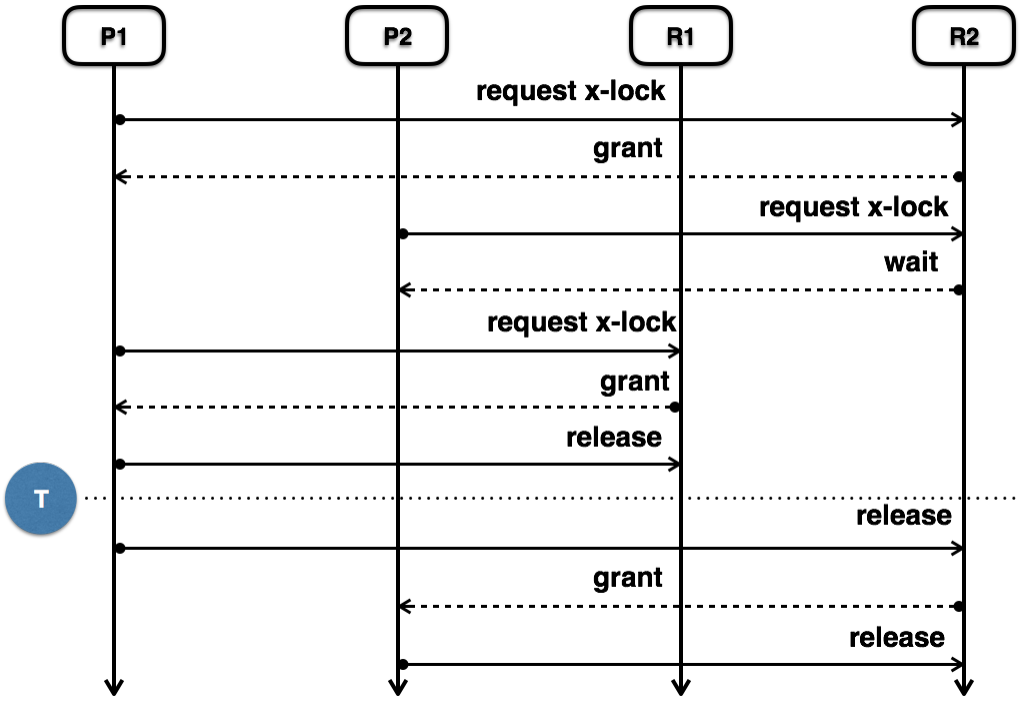

Figure 2:

f) Using the same notation as in part 2e above, the system in Figure 3 is in deadlock at the end of the sequence shown. Draw a new diagram that is a re-sequencing of Figure 3 (i.e., contains exactly the same events but in different order) that does not deadlock, and explain why the change you have made avoids that. (4 marks)

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

$$\begin{array}{ccc} A & \rightarrow & B \\ C & \rightarrow & E \\ C & \rightarrow & B \\ A & \rightarrow & C \\ E & \rightarrow & A \end{array}$$

Argue that this is (or is not) a consistent set of assertions. If it is consistent, add one single assertion that makes it inconsistent. If it is not consistent, remove one single assertion that makes it consistent. (4 marks)

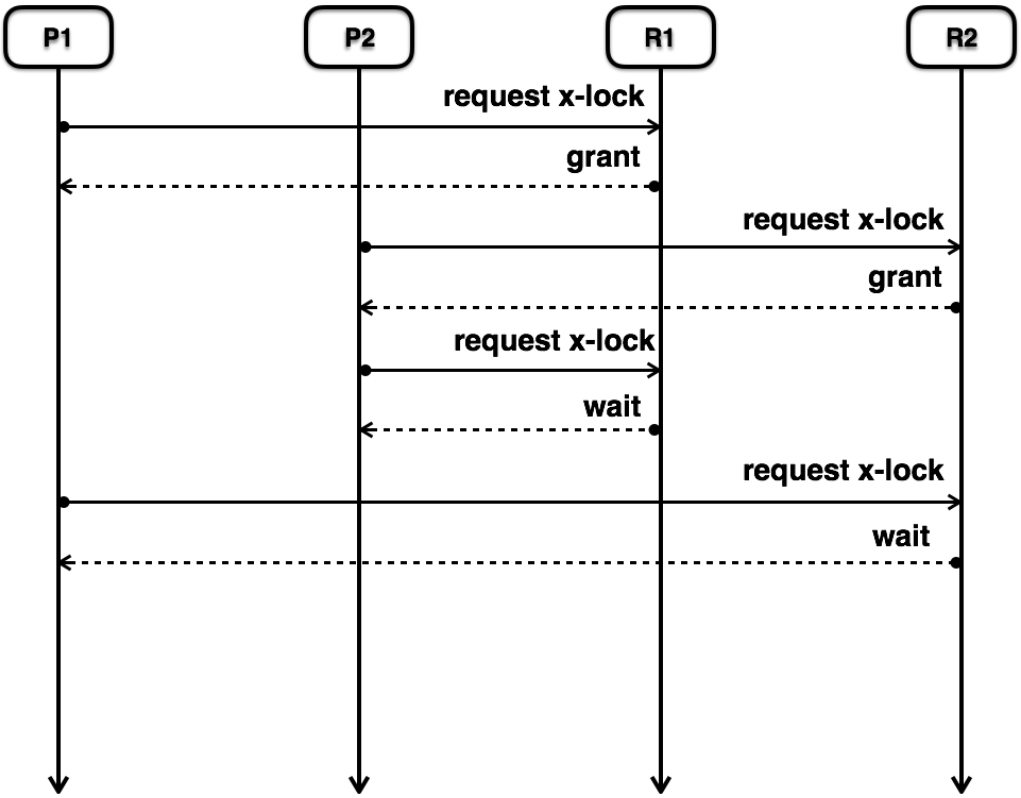

Figure 3:

END OF EXAMINATION