TDA596 / DIT240 (2nd academic period 2018/2019)

Exam: Distributed Systems

21. Aug. 2019

Examiner: Philippas Tsigas, philippas.tsigas@chalmers.se, phone: 772 54 09

Contact: Beshr Al Nahas, beshr@chalmers.se, office phone: 772 61 46

Means allowed: Nothing except paper, pencil, pen and English - xx dictionary.

Please answer all questions (1 to 6).

General information:

- All questions should be answered in English.
- Each question answer should be started on a new sheet of paper.
- You can write some of the answers on the printed exam questions papers. If you do so, make sure to separate these pages from the questions bundle, include these pages in your handed booklet, give them page number, and write your exam code on them (as you do with a normal answer sheet).
- Write clearly and use the pages in a structured way so your answers are easy to read. All answers should be motivated, explained, elaborated, detailed, precise and accurate.

Important suggestion: Read all questions before answering. Plan your time so that you can (at least) write a brief answer to all questions (and sub-questions). Please notice the weight that is given to each question (and sub-question).

Grading: GU: G 24p, VG 48p; CTH: 3:a 24p, 4:a 36p, 5:a 48p of maximum 60 points.

Review: Please keep your exam code. Information about individual exam review will be published on the course website.

Department of Computer Science and Engineering
Chalmers University of Technology

1. Basics about Distributed Systems (10 points)

1 a) (2 points) When designing or developing a distributed system, its distributed nature presents a number of challenges, which we discussed in the course. List at least six such challenges.

1 b) (3 points) Select three challenges you listed in the previous answer and detail on them. For each, focus on the following two questions:

- Why do these make developing Distributed Systems challenging?
- What mechanisms did we study in the course to deal with them?

1 c) (3 points) One way to classify distributed systems is based on the **communication paradigms**.

- We discussed the two related problems of space coupling and time coupling. Describe each of them briefly.
- Message queues is one indirect communication mechanism. Briefly describe it and briefly explain how it enables space and time decoupling.

1 d) (2 points) **Ethical challenges:** Certain distributed systems such as BitTorrent, Bitcoin and TOR trigger ethical challenges. Please select two of these systems and for each of them please discuss two ethical challenges.

2. Leader Election and Mutual Exclusion (10 points)

2 a) (5 points) **Electing a leader** among multiple nodes in the network

- What properties should a leader election algorithm in distributed systems have?
- What challenges does leader election in distributed systems face?
- Please name two algorithms we described in the course that elect a leader and briefly explain only one of them.
- Compare both algorithms, what key advantages and disadvantages do your see for each?

2 b) (5 points) **Mutual Exclusion** in Distributed Systems

- What is mutual exclusion and its goal in Distributed Systems?
- In the course, we discussed the **Token Ring algorithm for Mutual Exclusion**. Please explain this algorithm.
- What is the message complexity of a single access operation of this algorithm (please explain)? What is the message complexity when no process needs to access the resource?
- What do you do when the token is lost? Why is it challenging to detect a lost token?
- Any algorithm for mutual exclusion must fulfill two goals: safety and liveness. Explain how the algorithm achieves these.

3. Time and Synchronization (10 points)

3 a) (2 points) In the course we discussed the concept of **clock synchronization for physical clocks**. The manufacturer of the timer specifies the upper and the lower bound that the clock skew may fluctuate. This value is known as maximum drift rate (ρ).

- How far can two clocks drift apart if they were last synchronized since Δt seconds to UTC?
- If the maximum drift permissible in a distributed system is X seconds, how often shall the clocks resynchronize?

3 b) (3 points) Assume we need to synchronize the physical clocks of "n" nodes to **an accurate reference clock**.

- Name two algorithms we discussed in the course that could you choose. Explain why you chose them.
- Briefly illustrate one of the two algorithms. You can draw a figure to support your argumentation.

3 c) (5 points) In the course, we discussed how **Vector Clocks** help to distinguish causally related events and concurrent events.

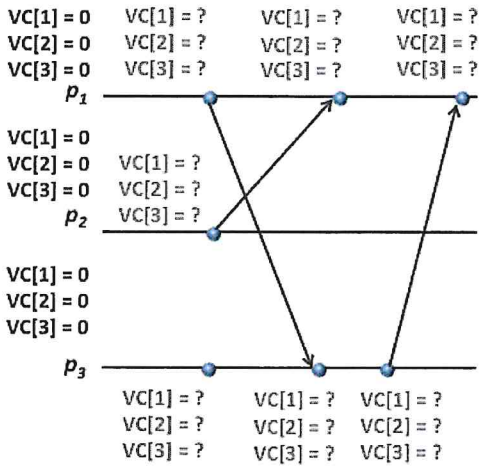
- Below you see pairs of Vector clocks. Note for each pair whether they denote concurrent or causally related events. Briefly explain your reasoning.
  - Are these two events causally related or concurrent?

Event on Node 1	Event on Node 2
VC[1]= 8	VC[1]= 9
VC[2]= 6	VC[2]= 8

  - Are these two events causally related or concurrent?

Event on Node 1	Event on Node 2
VC[1]= 8	VC[1]= 6
VC[2]= 6	VC[2]= 10

- Below you find a figure of a vector clock for three nodes. All vector clocks are initialized to zero. Please list the vector clocks for all the shown events (marked with dots).



4. Consistency and Replication (10 points)

4 a) (5 points) We discussed the concepts of **primary-based consistency protocols** and shown the two variants: Remote-Write Protocol and Local-Write Protocol. Answer the following questions and briefly reason about your answers.

- Are these protocols centralized, decentralized or distributed?
- What consistency model do they implement?
- Which one is good for a single writing client?
- Which one is good for multiple writing clients?
- Which one to use for a flight booking service?
- Which one to use for a Dropbox-like service?

4 b) (5 points) We discussed the concepts of **monotonic reads and writes.**

- Briefly explain the two concepts.
- Below you see three figures (1-3). For each figure, please note weather it describes monotonic writes. Briefly explain your decisions.

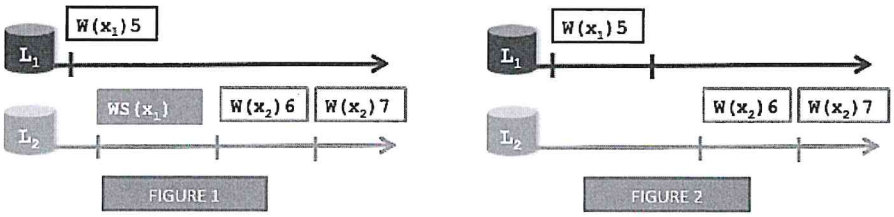


FIGURE 1

FIGURE 2

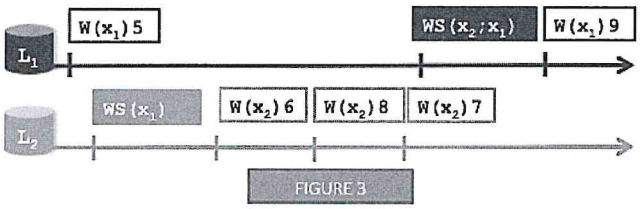


FIGURE 3

- Below you see another three figures (4-6). For each figure, please note weather it describes monotonic reads. Briefly explain your decisions.

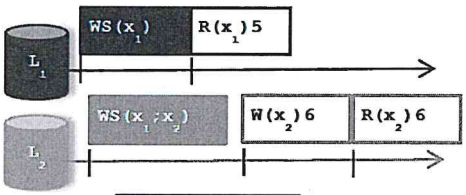


FIGURE 4

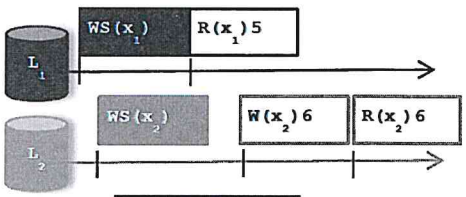


FIGURE 5

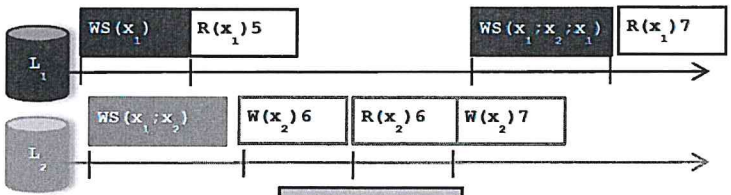


FIGURE 6

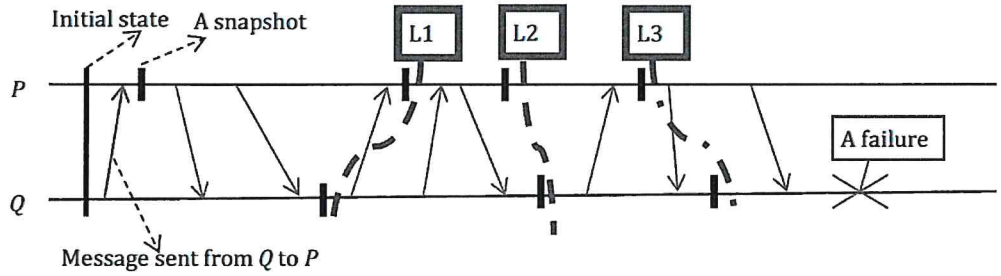
5. Fault Tolerance (10 points)

5 a) (5 points) We discussed **Process Failure Detection** in a distributed system.

- Briefly describe it.
- There are two policies for failure detection: active and passive. Briefly explain each of them.
- Explain the **timeout mechanism** for failure detection. What challenge could it face in realistic unreliable networks?
- **FUSE** is an example system that implements failure detection. Briefly explain it. You can draw a figure to support your explanation.

5 b) (5 points) In a fault-tolerant distributed system, backward recovery requires **checkpointing**.

- Briefly define checkpointing and the use of it.
- What is a recovery line?
- Briefly explain the **Domino Effect** when trying to find a recovery line. When could it happen?
- The figure below shows two processes communication and snapshots. Decide which of the lines (L1, L2, L3) make a valid recovery line and which do not. Explain your decision.



6. Naming and Applications (10 points)

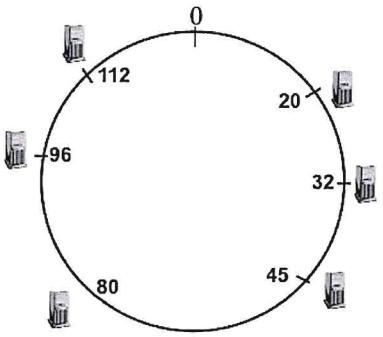
6 a) (3 points) We discussed **Bitcoin and blockchain consensus** and described two types of blockchains: Public and Private.

- Briefly compare Public and Private blockchains: Who can participate and how is consensus guaranteed? Briefly discuss the cost of each system and whether this cost is centralized or distributed.
- Does Bitcoin use a public or private blockchain? Why did the designers choose this paradigm?

6 b) (4 points) We discussed the **BitTorrent** protocol and the use of trackers: centralized and distributed (also known as trackerless).

- What are the pros and cons of dependence on a tracker?
- How does a "trackerless" BitTorrent work? Describe briefly. What problems does it solve compared to a centralized tracker?

6 c) (3 points) **Finger tables in Chord**: Below you find a picture of a Chord ring and the finger table of **node 80**. Please complete its finger table, i.e., list to which nodes the figures point and explain your calculations.
Note: in this example the namespace size is 0 to 127, i.e., finger table size is 7.



Chord Ring

i	*ft[i]*
0	?
1	?
2	?
3	?
4	?
5	?
6	?

Finger Table of node 80