TDA596 / DIT240 (2nd academic period 2017/2018)

Exam: Distributed Systems

22. Aug. 2018

Examiner: Olaf Landsiedel

Contact: olafl@chalmers.se, office phone: 772 10 96

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

Please answer 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. 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) Define the term “Distributed System”. Be brief and precise.

1 b) (3 points) Centralized algorithm, decentralized algorithm, and distributed algorithm.

- Please define the terms centralized algorithm, decentralized algorithm, and distributed algorithm.
- Briefly list the key differences between these three.
- Briefly, list two advantages of each approach.

1 c) (3 points) The Internet is one of the largest Distributed Systems today. Its architecture follows the so-called OSI Reference Model of 7 layers. Briefly name the seven layers and note the functions of each layer in keywords.

1 d) (2 points) Ethical challenges: Certain distributed systems such as BitTorrent and TOR trigger ethical challenges. List and briefly discuss two ethical challenges for each of these two systems.

2. Leader Election and Mutual Exclusion (10 points)

2 a) (4 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.
- When you compare both algorithms, what key advantages and disadvantages do your see for each?

2 b) (6 points) Mutual Exclusion in Distributed Systems

- What is mutual exclusion and its goal in Distributed Systems?
- What requirements should any algorithm that provides mutual exclusion in Distributed Systems provide? Please name and explain them.
- In the course, we discussed the Ricart & Agrawala algorithm for Mutual Exclusion. Please explain this algorithm.
- What is the message complexity of a single access operation of this algorithm (please explain)?
- How does the algorithm provide the requirements discussed above?
- Is Ricart & Agrawala a centralized, decentralized, or a distributed algorithm (please explain)?

3. Time and Synchronization (10 points)

3 a) (4 points) Assume we need to synchronize the physical clocks of "n" nodes but do **not** have access to a reference clock.

- Which algorithm (that we discussed in the course) would you choose and why?
- Briefly illustrate that algorithm. You can draw a figure to support your argumentation.
- Can this algorithm deal with a faulty clock? If so, how does it do this?
- For time synchronization between two nodes this algorithm relies on another algorithm. Please note which algorithms (that we discussed in the course) it might use and why.

3 b) (2 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?

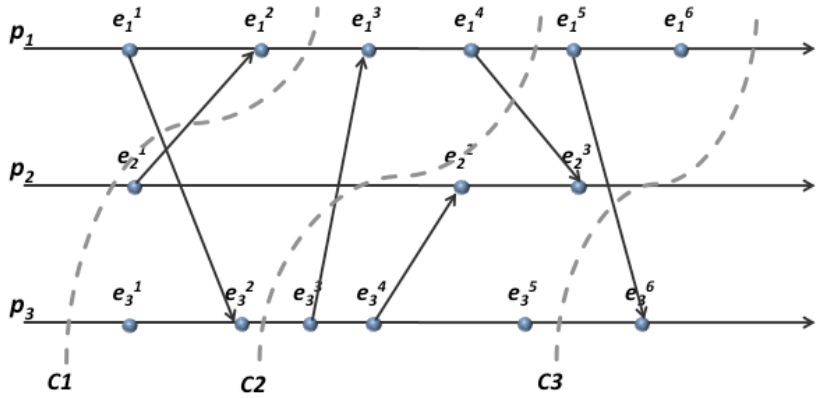
Event on Node 1	Event on Node 2
VC[1]= 4	VC[1]= 2
VC[2]= 2	VC[2]= 10

- Are these two events causally related or concurrent?

Event on Node 1	Event on Node 2
VC[1]= 4	VC[1]= 5
VC[2]= 2	VC[2]= 4

3 c) (4 points) We discussed the concept of consistent and inconsistent cuts.

- Please explain this concept briefly and formally define the term consistent cut.
- Please note for each cut (C1, C2, C3) depicted below whether it a consistent or inconsistent cut. Explain your reasoning briefly.



4. Consistency and Replication (10 points)

4 a) (3 points) Data-Centric vs. Client-Centric Consistency Models

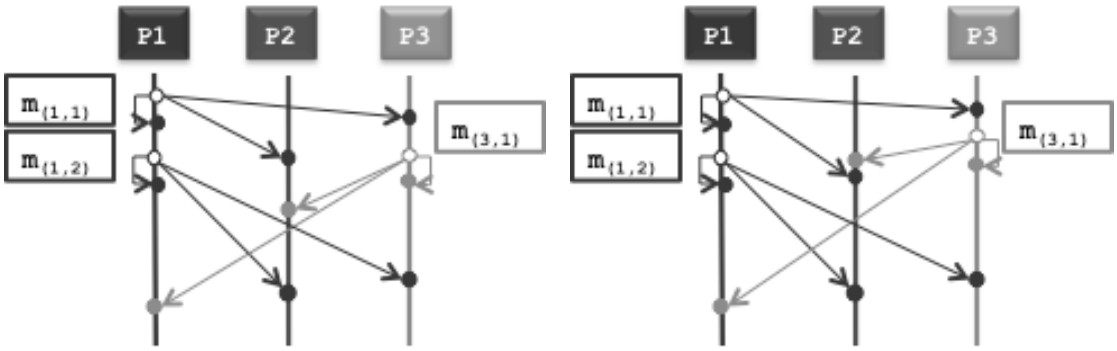
- Please explain the key difference between Data-Centric and Client-Centric Consistency Models
- Please define Eventual Consistency (within one short sentence).
- Is Eventual Consistency a Data-Centric or Client-Centric Consistency Model?

4 b) (3 points) We discussed the Bayou System, that provides a distributed calendar.

- What consistency model does Bayou use? And Why? How are activities ordered in Bayou?
- How are conflicts resolved in Bayou? Is this transparent, i.e., not visible, to the user?
- When are conflicts resolved? Does Bayou give a guaranteed upper bound for this?

4 c) (4 points) We discussed the concepts of Total Ordering, Sequential Ordering, and Causal Ordering

- Briefly explain and formally define each concept.
- Below you see two figures. For each figure, please note weather it describes Total Ordering, Sequential Ordering, and Causal Ordering. Briefly describe your decisions.



5. Fault Tolerance (10 points)

5 a) (3 points) Orphans: A client might crash while the server is performing a corresponding computation requested by the client. Such an unwanted computation is called an *orphan* (as there is no parent waiting for it after done).

- What problems do orphans cause?
- In the course, we discussed four strategies to deal with orphans. Please explain each of them.

5 b) (3 points) Failure Models: In the lecture we discussed different failure models. Please note four of them and describe each briefly.

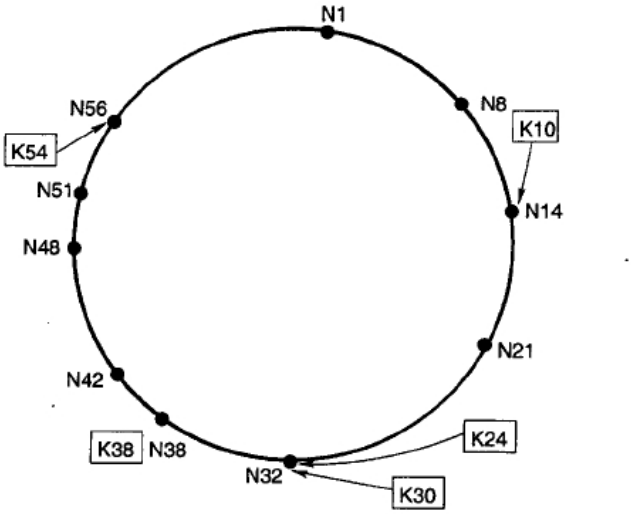
5 c) (4 points) We discussed the "Two Phase Commit" Protocol. As the name states, it consists of two phases.

- Please name and describe phase 1 briefly.
- Please name and describe phase 2 briefly.
- Please discuss what happens in case of a failure during *phase 1*, i.e., a node not replying because it crashed.
- Please discuss what happens in case of a failure during *phase 2*, i.e., a node not replying because it crashed.

6. Applications and Naming

6 a) (5 points) Distributed Hash Tables (DHTs): Below you find a picture of a Chord ring, with nodes N1, N8, N14 etc.

- Please list the finger-table of node N8, i.e., list to which nodes the finger-table points and explain your reasoning and calculations. Note: in this example, the finger table size is 6.
- Assume that node N8 aims to lookup the data item K54, stored on node N56. How is this lookup request routed from N8 to N56, the node storing data item K54? Please explain your reasoning.
- What are the key differences between the Chord and CAN DHTs?



6 b) (5 points) Amazon Dynamo, a (key, value) store for datacenters.

- Dynamo is a modified version of the Chord protocol. What are the key differences? And Why?
- Dynamo introduces the concept of virtual nodes, what are they used for?
- How are events in Dynamo timestamped?
- What consistency model does Dynamo use? And why?
- How are conflicts resolved in Dynamo? During which operation? And When?