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

Exam: Distributed Systems

5th of April 2018

Examiner: Olaf Landsiedel

Contact: olafl@chalmers.se, office phone: 772 10 96 or 072-974 48 54

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) (1 points) The following statement is attributed to Thomas J. Watson, Chairman and CEO of International Business Machines (IBM), in 1943: "I think there is a world market for maybe five computers". Please state the key consequences for distributed systems if this sentence had been the correct vision.

1 c) (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 d) (2 points) Peer-to-Peer Architectures vs. Client-Server.

- Please explain and summarize the key differences of the two approaches.
- Are Client-Server Architectures commonly designed for a centralized algorithm, decentralized algorithm, or distributed algorithm?
- Are Peer-to-Peer Architectures commonly designed for a centralized algorithm, decentralized algorithm, or distributed algorithm?

1 e) (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) (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 election 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) (5 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) (5 points) In the course, we discussed a Distributed Snapshot Protocol (Chandy and Lamport).

- Please explain this algorithm: How does it construct a snapshot?
- Please explain why this algorithm works correctly, i.e., it does not lead to inconsistent snapshots.

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 (with 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 concept of a Centralized Active Replication Protocol.

- Briefly explain this concept. You can draw a figure to support your argumentation.
- The protocol uses a central entity for parts of its operations. Briefly explain why this is a reasonable design.

5. Fault Tolerance (10 points)

5 a) (5 points) We discussed the "Byzantine Generals Problem".

- What general problem does its algorithm address?
- Why is this an important problem to address? You can illustrate your answer by briefly describing a situation/application that requires such an algorithm.
- In the "Byzantine Generals Problem" there are honest generals and dishonest generals (traitors). What is the goal of the honest generals? What is the goal of the traitors?
- In the lecture, we introduced an algorithm with multiple phases for the Byzantine Generals Problem. Explain the algorithm and its different phases.
- Under what conditions can the generals achieve consensus. How many honest generals are required, assuming that there are *k* dishonest ones?

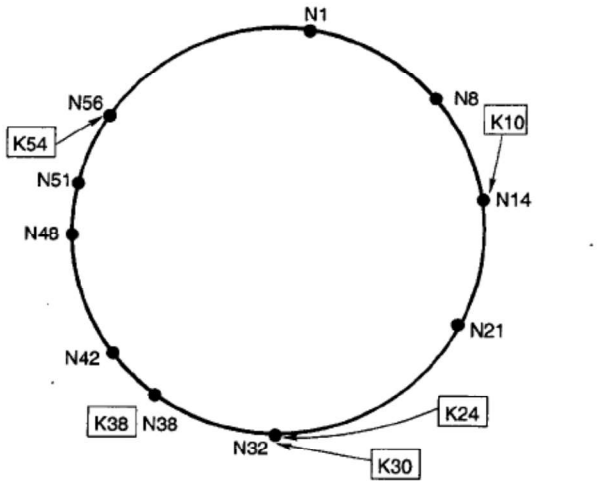
5 b) (5 points) We discussed the "Two Phase Commit" Protocol. As the name states, it consists of two phases.

- What general problem does the "Two Phase Commit" Protocol address?
- Why is this an important problem to address? You can illustrate your answer by briefly describing a situation/application that requires a "Two Phase Commit" Protocol.
- Please describe the two phases of the protocol 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 figures point 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 the 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?