TDA596 / DIT240, 7.5 hp, (2nd academic period 2013)

Exam: Distributed Systems

25 April 2014

Examiner: Olaf Landsiedel

Contact: olafl@chalmers.se, cell phone: 070-9702443, 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. Write clearly and use the pages in a clever way so it is easy to read. Each question answer should be started on a new sheet of paper. 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

CHALMERS

1. Basics about Distributed Systems (10 points)

1 a) (1 points) Define the term "Distributed System" in your own words. Be brief and precise.

- Multiple devices
- Connected by a network
- Cooperating on some task

1 b) (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.

- The network is not reliable
- Latency is not zero
- Bandwidth is not infinite
- The network is not secure
- Topology does change
- There are multiple administrators
- Transport cost is not zero
- The network is not homogeneous
- Clocks are not always in sync
- Computers (hardware and software) are not reliable

1 c) (3 points) Select three challenges you listed in 1b) 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?

- Challenge A: (a) Impact, (b) mechanism(s) to deal with it
- Challenge B: (a) Impact, (b) mechanism(s) to deal with it
- Challenge C: (a) Impact, (b) mechanism(s) to deal with it

1 d) (2 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.

- DS would not be as we know them today
- There would be essentially no DS research
- There would be no distributed systems course
- Many others answers are valid
- Not correct: these computer would be highly loaded and we would be in huge trouble if one fails (because: in this case we would just buy another computer)

1 e) (2 points) Centralized Systems (e.g. Client Server Systems) vs. Decentralized Systems (e.g. Peer-To-Peer Systems): We discussed the differences between the centralized architectures and the decentralized architectures. Describe each system briefly and discuss two advantages of each:

- Centralized
  - Describe
  - Advantage
    - Easy to setup (+explain)
    - Easy manage and control: (+explain)
    - Reduced software complexity: little consistency problems etc. (+explain)
- Decentralized
  - Describe
  - Advantage
    - High scalability (+explain)
    - Higher Robustness (+explain)
    - No single point of failure (+explain)

2. Mutual Exclusion and Election (10 points)

2 a) (2 points) Define the terms "Mutual Exclusion" and "Election" in your own words. Be brief and precise.

i. Mutual Exclusion
How to coordinate between processes that access the same resource?
Manages access to (for example) resources
Goal: unique access

ii. Election Algorithms
Here, a group of entities elect one entity as the coordinator for solving a problem

2 b) (3 points) Token Ring for Mutual Exclusion.

i. Please describe how the Token Ring algorithm works.
In the Token Ring algorithm, each resource is associated with a token
The token is circulated among the processes
The process with the token can access the resource
Circulating the token among processes:
All processes form a logical ring where each process knows its next process
One process is given a token to access the resource
The process with the token has the right to access the resource
If the process has finished accessing the resource OR does not want to access the resource: it passes the token to the next process in the ring

ii. List at least two limitations of its design.
Token ring has a high-message overhead
When no processes need the resource, the token circulates at a high-speed
If the token is lost, it must be regenerated
Detecting the loss of token is difficult since the amount of time between successive appearances of the token is unbounded
Dead processes must be purged from the ring
(ACK based token delivery can assist in purging dead processes)

2 c) (2 points) Any algorithm for mutual exclusion must fulfill two goals: safety and liveness.

i. Please define these two goals, be brief and precise.
Safety: At most one process may execute in *critical section (CS)* at any time
Liveness: Every request for a CS is eventually granted

ii. Explain how the Token Ring algorithm achieves these.
Safety: There is only one token, and the resource cannot be accessed without a token
Liveness: Each process will receive the token

2 d) (3 points) We discussed the Ring Algorithm for Election.

i. Please describe how the Ring algorithm works. For simplicity, assume that there is only one initiator for the election. Please also note what assumption the algorithm makes on the topology.
This algorithm is generally used in a ring topology
When a process Pi detects that the coordinator has crashed, it initiates an election
$\mathbf{P_i}$ and sends an election msg. (E) to its next node.
It inserts its ID into the Election message
When process $\mathbf{P_j}$ receives the message, it appends its ID and forwards the message
If the next node has crashed, $\mathbf{P_j}$ finds the next alive node
When the message gets back to the process that started the election:
it elects process with highest ID as coordinator, and
changes the message type to "Coordination" message (**C**) and circulates it in the ring

ii. What is its message complexity? Assume that there are n nodes and one initiator.
O(2n) or 2n-1

3. Naming (10 points)

3 a) (2 points) Define the terms "naming" and "name resolution" in Distributed Systems in your own words. Be brief and precise.

- Names are used to uniquely identify entities in Distributed Systems
- Names are mapped to an entity's location using a *name resolution* [Entities may be processes, remote objects, newsgroups, ...]

3 b) (3 points) In the course we discussed two concepts for name resolution: Iterative and recursive name resolution. Briefly describe each concept and highlight their key differences.

Iterative: Naming lecture slide 36
Recursive Naming lecture slide 38
Key difference: Recursive: client not much involved, main load on name server. Iterative: stronger involvement of the client

3 c) (2 points) In the lecture we discussed the concept Chord. Chord is a Distributed Hash Table (DHT). Answer the following questions about Chord:

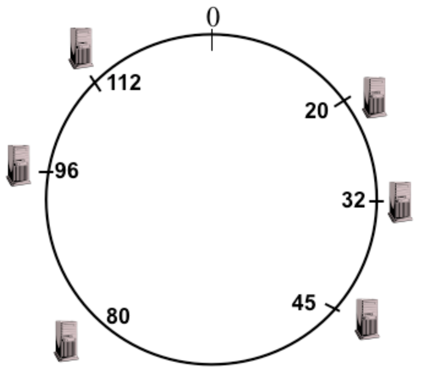
i. What topology do the nodes form?
Ring

ii. What operations does a DHT, e.g., Chord, provide?
Put and get

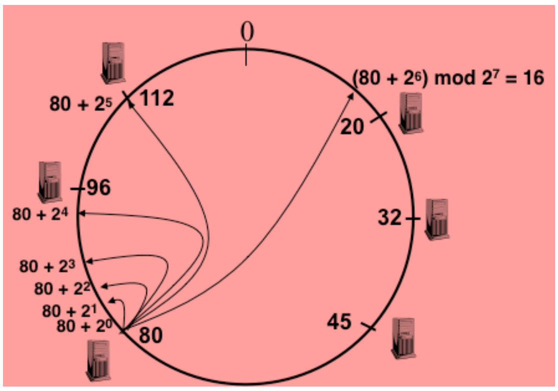
iii. How is redundancy in Chord achieved?
Data is stored on the neighboring nodes (the ones that follow in the ring)

iv. In Chord, how many hops does it take on average to lookup a data item? (Assume that the number of nodes in the Chord is "n").
Log (n)

3 d) (3 points) Finger tables in Chord: Below you find a picture of a Chord ring (on the left) and the finger table of node 80 (on the right). 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.



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



i	*ft[i]*
0	96
1	96
2	96
3	96
4	96
5	112
6	20

4. Time and Synchronization (10 points)

4 a) (1 points) In the course we discussed the concept of clock synchronization for physical

clocks. Why is clock synchronization important in Distributed Systems that rely on physical clocks?

- Timestamps would otherwise be invalid across different nodes

4 b) (2 points) Please answer the following questions about logical clocks.

i. How do logical clocks differ from physical clocks?

- No physical time.

ii. Note two types of logical clocks.

- Lamport and Vector clocks

iii. When do logical clocks "tick", i.e., when are they incremented?

- On an event

iv. When a node sends a message to another node, what does it do with its logical clock?

- Increment by one and piggyback with the message

4 c) (3 points) In the course, we discussed the concept of Vector Clocks. Please answer the following questions about Vector Clocks.

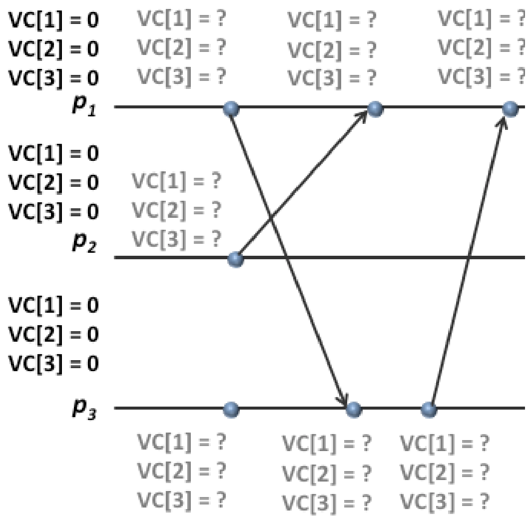
i. How many clocks does each node maintain?

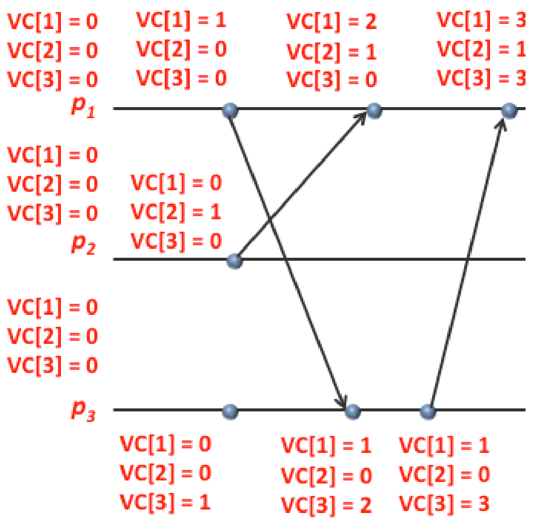
- One per node in the network

ii. Upon receiving a message, what does a node do with its vector clock(s)

- Local clock: increase by one
- Others: merge with piggybacked clock, take maximum

iii. 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 each event.





4 d) (4 points) In the course, we discussed how Vector Clocks help to distinguish causally related events and concurrent events.

i. Please explain how this can be achieved
Slides: 09_clocks02_OL, slide 45

ii. 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]= 1	VC[1]= 1
VC[2]= 0	VC[2]= 6

Causal relation, Node 2 has received vector clock of node 1
can only get this from communication -> causal relationship

- Are these two events causally related or concurrent?

Event on Node 1	Event on Node 2
VC[1]= 1	VC[1]= 0
VC[2]= 0	VC[2]= 6

No causal relation: Node do not each other vector clocks -> concurrent

5. Consistency and Replication (10 points)

5 a) (1 points) Define the term replication. Be precise and brief

- Replication is the process of maintaining data at multiple computers/nodes.

5 b) (1 points) List at least three benefits that replication provides. Explain these benefits briefly.

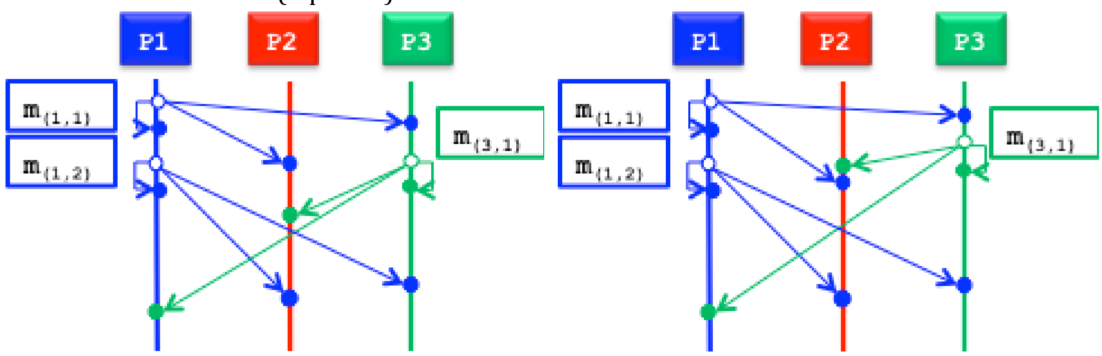
- Improving performance: A client can access the replicated copy of the data that is near to its location
- Increasing the availability of services: Replication can mask failures such as server crashes and network disconnection
- Enhancing the scalability of the system: Requests to the data can be distributed to many servers which contain replicated copies of the data
- Securing against malicious attacks: Even if some replicas are malicious, secure data can be guaranteed to the client by relying on the replicated copies at the non-compromised servers

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

i. Briefly explain each concept (1 point).

- Total Ordering: If process $\mathbf{P_i}$ sends a message $\mathbf{m_i}$ and $\mathbf{P_j}$ sends $\mathbf{m_j}$, and if one correct process delivers $\mathbf{m_i}$ before $\mathbf{m_j}$ then every correct process delivers $\mathbf{m_i}$ before $\mathbf{m_j}$
- Sequential Ordering: At any process, the set of messages received are in some sequential order. Messages from each individual process appear in this sequence in the order sent by the sender
- Causal Ordering: If process $\mathbf{P_i}$ sends a message $\mathbf{m_i}$ and $\mathbf{P_j}$ sends $\mathbf{m_j}$, and if $\mathbf{m_i \rightarrow m_j}$ (operator '$\rightarrow$' is Lamport's **happened-before** relation) then any correct process that delivers $\mathbf{m_j}$ will deliver $\mathbf{m_i}$ before $\mathbf{m_j}$

ii. Below you see two figures. For each figure, please note weather it describes Total Ordering, Sequential Ordering, and Causal Ordering. Briefly describe your decisions (3 points).



- Total Ordering: left no (?), right no (?)
- Sequential Ordering: left yes (?), right yes (?)
- Causal Ordering: left yes, right no

5 d) (4 points) We discussed the concept of monotonic writes.

i. Briefly explain the concept (1 point).

ii. Below you see three figures. For each figure, please note weather it describes monotonic writes. Briefly describe your decisions (3 points).

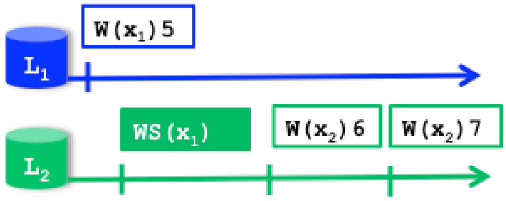


FIGURE 1

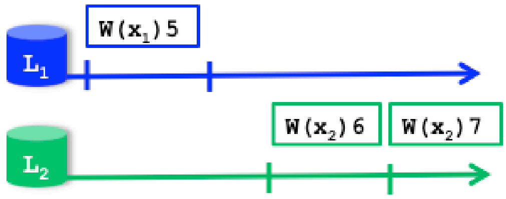


FIGURE 2

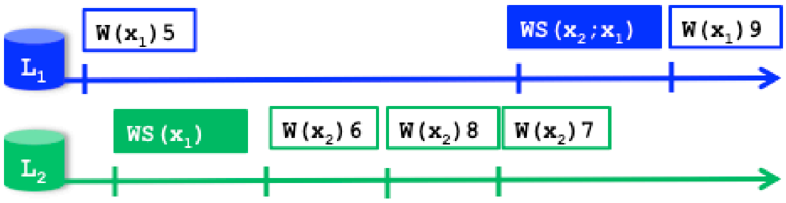


FIGURE 3

6. Fault Tolerance (10 points)

6 a) (2 points) Define the term "fault tolerance". Be brief and precise.

- Fault-tolerance is the property that enables a system to continue operating properly in the event of failures

6 b) (4 points) We discussed the "Byzantine Generals Problem".

i. 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 (1 point)?

- Honest generals: reach consensus even in the presence of traitors.
- Traitors: To break the consensus without the honest generals noting it: some part of the generals should think all agreed on X, another should think all agreed on Y.

ii. In the lecture we introduced an algorithm with multiple phases to enable consensus among the generals. Explain the algorithm and its different phases (2 points).

- Phase 1: vote on action / value
- Phase 2: exchange the votes
- Phase 3: conclude consensus from votes (need more detail

iii. Under what conditions can the generals achieve consensus. How many honest generals are required, assuming that there are *k* dishonest ones (1 point)?

- In their paper, *Lamport et al.* (1982) proved that in a system with ***k*** faulty processes, an agreement can be achieved only if ***2k+1*** correctly functioning processes are present, for a total of ***3k+1***.
- i.e., An agreement is possible only if more than two-thirds of the processes are working properly.

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

i. Please name and describe phase 1 briefly.

- Voting Phase
- Step 1: The coordinator sends a VOTE_REQUEST message to all participants
- Step 2: When a participant receives a VOTE_REQUEST message, it returns either a VOTE_COMMIT message to the coordinator indicating that it is prepared to locally commit its part of the transaction, or otherwise a VOTE_ABORT message.

ii. Please describe what happens during a failure in phase 1, e.g., a node not replying or a message getting lost.

- TODO

iii. Please name and describe phase 2 briefly.

- Decision Phase
- Step 1
  - The coordinator collects all votes from the participants.
  - If all participants have voted to commit the transaction, then so will the coordinator. In that case, it sends a GLOBAL_COMMIT message to all participants.
  - However, if one participant had voted to abort the transaction, the coordinator will also decide to abort the transaction and multicasts a GLOBAL_ABORT message.
- Step 2
  - Each participant that voted for a commit waits for the final reaction by the coordinator.
  - If a participant receives a GLOBAL_COMMIT message, it locally commits the transaction.
  - Otherwise, when receiving a GLOBAL_ABORT message, the transaction is locally aborted as well.

iv. Please describe what happens during a failure in phase 2, e.g., a not node replying or a message getting lost.

- TODO