TECHNISCHE UNIVERSITÄT DARMSTADT

TK1: Distributed Systems - Programming & Algorithms

Exam Preparation Questions
Submission Date: -

Prof. Dr. Max Mühlhäuser
Dr. Immanuel Schweizer

Jens Heuschkel, MSc.
Michael Stein, MSc.

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

The following list of questions does NOT represent typical exam questions in three respects:
- exam questions will be, for the most part, less "open form"
- we will make an effort to formulate them such that ambiguities are avoided
- there might be a larger portion of questions related to a solution to be sketched, written in pseudo-code, "computed", etc.

Nevertheless, the questions below give you a good indication of the "severity level" and the "level of detail" you should prepare for.

DISCLAIMERS:
- the questions are provided as a courtesy; they do not warrant any implications about what we may or may not ask in the exam
- in particular, we do not claim that we have covered all relevant (sub)issues relevant for the exam.

Chapter 1:

- Of which components does a computer network consist?
- What distinguishes a distributed system from a computer network?
- Name and explain the three basic problems of distributed systems.
- Name and explain three requirements of distributed systems.
- How does the P2P paradigm differ from the client/server model?
- Describe the structure of a 3-Tier Architecture.

Chapter 2.1:

IPC

- Define Interprocess Communication in one sentence.
- What elements does a socket address have?
- Describe an approach to identify a connection if several connections may ‘end’ at a server’s socket.
- In the context of interprocess communication, what is meant by the term integrity?
- What are the important characteristics of UDP (reliability, omission, ordering)?
- Sketch a scenario where UDP should be used rather than TCP. What characteristics make UDP superior in this scenario?
- How does Nagle’s algorithm work?
- What’s the basic idea of the Advanced Message Queuing Protocol?

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

RPC

- State three basic properties of RPC.
- Explain the three possibilities for binding introduced in the lecture.
- Precisely describe the characteristics of at-least-once semantics.
- Sketch a sequence diagram that demonstrates the idea of parallel handlers on the server side.
- Explain three general issues of RPC. For each issue, describe a relevant real-world scenario where the issue is relevant

Web Services

- What's the purpose of a WSDL port type?
- Give the WSDL definition (types, messages, portType and operations) of the following web service, where MyClass has the attributes "double q" and "int i".
  public MyClass getValue(double x)
- Explain three key design principles of REST.
- Explain the difference between a choreography and an orchestration.

Chapter 2.2:

Publish/Subscribe

- Explain downstream duplication and draw an example graph demonstrating the idea.
- For JMS, explain the difference of synchronous and asynchronous message consumptions.
- Explain "time decoupling" and give a real-world example.
- Specify formally: A notification n matches a filter F.

Tuple Spaces:

- What basic operations does Linda provide?
- Sketch in a diagram how the Master-Worker pattern works.

Distributed Shared Memory:

- Explain the basic idea of shared memory.
- Discuss the differences of DSM and Message Passing in terms of efficiency.

Chapter 2.3:

- What are the key characteristics of Cloud Computing?
- Explain the service model "Platform as a Service".
- What's the key motivation behind grid computing?
- Discuss the meaning of Amdahl's law in the context of grid and cloud computing
- What is an Amazon Elastic Block Store (EC2)?
- What is a chunk in the context of GFS?
- Sketch a scenario where a key/value store is used rather than a relational database.
- What functions have to be specified if MapReduce is used?

Chapter 2.4:

- Describe the three steps of an agent migration.

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

- Sketch the difference of Client-Server with RPC and Mobile Agents in a diagram.
- Describe a possible application of agents and discuss the advantages of agents in this scenario.
- How does agent cloning differ from agent roaming?

Chapter 2.5 / Chapter 3.5:

Formal Approaches:
- What's the motivation of using formal approaches in the context of distributed systems?
- When are two services said to be behavioral equivalent?

Local Algorithms:
- Describe three possible goals of topology control.
- What knowledge about the surrounding topology does a node in the LOCAL model have initially?

Chapter 3.1:

- What is meant by the term linear waiting?
- What observation does Gustafson's Law rely on?
- Explain the safety property "deadlock freedom"

Chapter 2.2:

- Define precisely the difference between a synchronous distributed system and an asynchronous distributed system.
- State the two observations that Christian's Algorithm builds on.
- Define the happened before relation.
- When is a state S' said to be reachable from state S?
- Sketch an example scenario where it is desirable to make a snapshot of a distributed system.
- What guarantees does the Chandy-Lamport algorithm provide?

Chapter 3.3
Reliable/unreliable failure detectors (FD) in synchronous/asynchronous systems.
- Why can a FD distinguish unsuspected/suspected or unsuspected/failed, but not running/failed?
- Why is time distance between two "alive" messages so important for unreliable FD?

Mutual Exclusion Algorithms
- A process can participate in a Mutex based on three (kinds of) function calls, "enter()", "exit()" and "access()". How do the first two differ from the latter?
- What is the bandwidth, client delay and synchronization delay of the "central server" algorithm?
- Which are the pros and cons of the ring-based algorithm?
- Which of the desired properties does the logical clock based algorithm fulfill?

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

Voting Algorithm
- What is the difference between a voting algorithm and an election algorithm?
- What does "equal effort" and "equal opportunity" stand for? Under which condition(s) is it impossible to fulfill both fairness conditions?
- Describe the easy-to-program approximation to the ideal/fair solution.
  Draw the illustration of a sample solution for 13 nodes and mark the voting set for node 7

Election Algorithms
- Try to describe *precisely* (as formal as possible)
  - which problem an election algorithm is supposed to solve
  - which assumptions are made (mostly: about the setup)
  - which property/ies an election algorithm should fulfill
- assume a list of known process numbers "2,4,6,8,10";
  after crash of the elected leader, process 6 and then 4
  start an election; provide a log file of all messages exchanged
  until the algorithm(s) terminate(s) in the form
  N->M: <type> (where N and M are process numbers, <type> is
    a message type of the algorithm
  In cases where the sequence of messages is not fully deterministic, choose one possible sequence. Assume that before a process can send a second message and before the receiver can react, at least one other message must have been send *IF* such a message is possible according to the algorithm (otherwise, the algorithm will just be thought to pause for one "message-transmission-period")
- solve the problem above for the case of elections on rings

Chapter 3.4

Multicast
- How does a multicast have to go beyond "syntactic sugar" for N times sending-a-message (for N recipients) in order to deserve the name multicast (several options possible)?
- Why does Basic-Multicast fulfill (one of) these properties?
- Basic Multicast has a problem that some other multicast algorithms also have; what is the name of this problem and what exactly IS the problem?
- For R-IP-Multicast:
  - o which datastructures does a sending process piggyback on onto the message to be multicast? (what is their meaning?)
  - o for a receiving process (that gets an IP-Deliver), what are the three possibilities concerning the sequence number S of the message received, and how does the receiving process react?
  - o as to the other datastructures that are checked during IP-Deliver, what is checked and how does the process react?
- For FIFO-, Causal- and Total-Multicast:
  - o For which of the three may it be the case that some messages are delivered in different order at different (receiving) processes?
  - o Is Causal-Multicast also a FIFO-Multicast? Why (not)?
  - o Imagine a Forum (electronic bulletin board, Newsfeed, ...) to display three meta data for any message: (1) (local, receiving) <sequence number>,
    (2) <Sender>, and (3) <Subject> or <RE: Subject>, the latter for replies
    Show an example of an inconsistency experienced by a user IF
    + Total-Ordering is not given
    + FIFO-Ordering is not given

TELEKOOPERATION
Fachbereich Informatik
Hochschulstr. 10
64289 Darmstadt

+ Causal-Ordering is not given

- For realizations of Total-Ordering ...
  - o via "Sequencer": how does the sequencer react when it receives, via B-Deliver, a message m plus unique identifier i
  - o via "Sequencer": what is the advantage of the algorithm over an algorithm in which the sender FIRST requests a global sequence number from the sequencer and THEN broadcasts its message along with this sequence number instead of the unique identifier, and under which assumption(s) is this really an advantage?
  - o for "Collective Agreement": when exactly does a process TO-Deliver a message to the user?
  - o for "Collective Agreement": Why can a message with an Agreed Sequence Number that arrives at the front of the holding queue not be re-ordered any more to a later position in the queue?