

CSCI6421: Distributed Systems Practice Problems

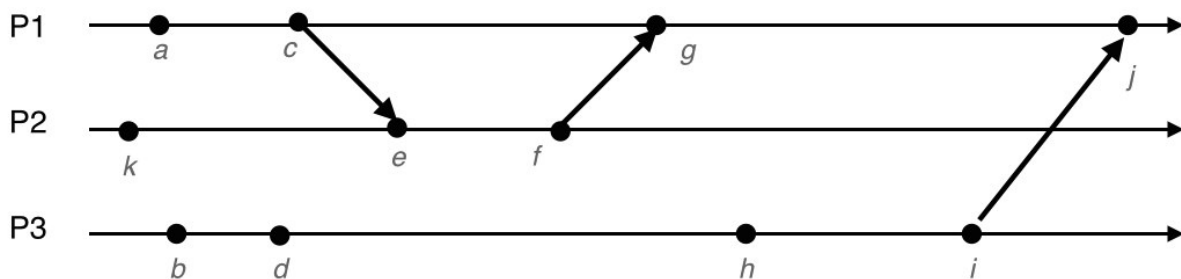
These practice problems are designed to give you a sense of the type of questions that might be asked in your exam. However, it should not be considered a complete coverage of topics you might be asked about. Also, these questions were designed for a written test; for an online exam the format of the exam may need to be adapted (particularly Q10 would need to be modified).

1. True or False

- _____ Two programs communicating over sockets must be written in the same programming language.
- _____ Peer to Peer services are the only type of distributed system where a node is both a client and a server.
- _____ Lamport clocks cannot always put all events into a total order, but vector clocks always can.
- _____ Virtualization provides strong isolation, but it is generally slower than running code natively.
- _____ Containers are a good choice for operating system kernel developers because they can be started very quickly, allowing you to debug as you modify the container's OS kernel.
- _____ Most cloud applications use UDP to communicate with clients over the Internet.
- _____ A Decentralized Architecture will always provide better reliability compared to a centralized one.

2. Clocks and Ordering

Annotate the process timeline with Vector Clocks for each event.



Using your vector clocks, indicate whether an event happens before (\rightarrow), after (\leftarrow) or is concurrent (\parallel).

- b _____ g g _____ i
- j _____ h f _____ j

3. Distributed Systems Challenges

Pick 3 of the distributed systems challenges below and write 3-4 sentences for each one explaining a system or technique we learned about that helps resolve them.

Challenges: heterogeneity, openness, security, fault handling, concurrency, quality of service, scalability, transparency

4. Threads and Processes

What is a case where it is preferable to use multiple threads instead of multiple processes?
What is a case where the reverse is true?

5. Virtualization

a) Why do virtual machines use technologies like memory virtualization? Why can't VMs be given real physical memory addresses to work with directly?

b) Why are containers "lighter weight" than VMs?

6. Architecture

What is an example of a system that follows a layered architecture? Give an example of one Distributed Systems Challenge that is made easier by a layered architecture and one that becomes harder.

7. Fault Tolerance

You are designing a distributed system that uses replication to mask faults for a stateless application. If we assume at most $f=4$ faults can ever happen at one time, then how many replicas are needed in each of these cases:

a) Replicas either respond correctly, do not respond at all, or can return any incorrect value.

b) Replicas either respond correctly or do not respond at all.

d) Replicas either respond correctly or return an error code: NULL.

8. Scheduling

Suppose you are designing a control system for an autonomous car. The car has a variety of sensors (cameras, range finders, etc), and each new reading generates an analysis task that processes the data and potentially adjusts the car's behavior. What kind of scheduling algorithm would you use for these analysis tasks? Why?

9. Scaling

Suppose you have a stateful service. What techniques could you use to scale it vertically or horizontally? (describe each individually)

10. Case Study

For this question you will be judged on the completeness, creativity, and clarity of your answer. Note that the case study description is intended to be somewhat vague. You should specify what assumptions you are making and how that affects your solution.

You are building a sensor network to study a mountain chain with several active volcanos. The network will be composed of two types of nodes: 1) Small nodes that have a radio with a radius of about 100 meters and are low power with limited processing and memory capabilities; 2) Large nodes that have a radio range radius of 1000 meters and have greater storage and processing capabilities than small nodes. The nodes are designed to be dropped out of an airplane over a wide area.

Each of the nodes will gather data that will be analyzed by scientists to detect potential eruptions. Both types are capable of gathering the same data from their sensors. A data point takes 1KB of memory. The small nodes have 10KB of total memory available and the large nodes have 1,000KB.

The scientists want to be able to answer questions like: what is the average data reading in a geographic region covered by several sensors? However, in the future they may devise more complicated programs that will run on each sensor, so they want to have a flexible system.

Draw diagrams and write an explanation to justify your design for this system. Which of the distributed systems challenges below does your system help with? How?

Challenges: heterogeneity, openness, security, fault handling, concurrency, quality of service, scalability, transparency