

Application-Based Distributed Computing Questions

Exam Preparation for CS324

The following questions are designed to test the practical application of key Distributed Computing concepts.

Primary Backup Protocol

1. A global e-commerce platform uses a primary-backup protocol to ensure high availability. Describe how a migrating primary variant might help improve resilience for international transactions. What factors should be considered for selecting primary nodes?
2. Imagine a healthcare application that uses a local write primary-backup variant for real-time patient data updates. How would you handle synchronization issues between the primary and backup in a geographically distributed system?

Consistency Models

3. A social media app wants to ensure users see the latest posts when they access their feed but allows some delay for less important updates. Which consistency model would you recommend, and why?
4. A collaborative document editing tool needs to support users working from various locations. Discuss which client-centric consistency model is best suited for this scenario, and how it would function.

Device Naming and Tracking

5. Consider a logistics company that uses mobile tracking for its delivery vehicles. Which naming technique - forwarding pointers, broadcasting, or multicasting - would best balance efficiency and real-time tracking? Explain your choice.
6. In a smart city network, mobile devices need to be tracked as they move across zones. Explain how forwarding pointers could be utilized for seamless connectivity without excessive broadcasting.

Leader Election

7. In a distributed file-sharing system using peer-to-peer architecture, explain how the ring algorithm can be used for leader election among nodes to manage data indexing.
8. A network of sensors in a factory monitors different metrics. If one sensor node is selected as a leader to aggregate data, describe how the ring algorithm could support leader election.

Fault Tolerance

9. A distributed financial service must handle multiple types of failures (e.g., crash, Byzantine) for reliable transaction processing. Explain how k fault tolerance would be applied in this system to manage failure handling.
10. A university's distributed system is designed to manage examination data. Discuss how you would mitigate timing and omission failures to ensure data consistency across campuses.

Communication Techniques

11. A weather prediction system uses distributed nodes to collect and share data. Discuss how asynchronous communication would improve system reliability when nodes are spread across remote locations.
12. An IoT-based healthcare monitoring system requires persistent communication to store patient records in the cloud. Explain the choice of communication type and techniques that would support consistent data flow in case of network delays.

Security

13. A distributed e-voting system requires a high level of security for confidentiality, integrity, and authentication. Explain how symmetric and asymmetric encryption would help secure this system and ensure data authenticity.
14. In a distributed file storage service, digital signatures are used to verify data changes. Describe how this ensures non-repudiation and prevents unauthorized modifications.

Ordered Multicast

15. An online gaming platform needs to ensure all players see events (like scoring) in the same order. Would causally ordered or totally ordered multicast be more suitable? Justify your answer.
16. In a distributed auction system, bidders' actions need to be synchronized. Discuss how totally ordered multicast helps maintain fairness in bid visibility for all participants.

Clock Synchronization

17. A multinational financial company relies on accurate timestamps for transactions. Explain how the Network Time Protocol (NTP) and Cristian's algorithm could be used to maintain synchronized clocks across different regions.
18. An airline reservation system needs to log events consistently. Discuss how vector clocks might be used to avoid conflicts in concurrent updates to seat reservations.

Distributed Commit and 2-Phase Commit Protocol

19. A banking system handles inter-bank transfers using the 2-Phase Commit protocol. Explain how the protocol guarantees atomicity during transaction processing, even if some banks experience network delays.
20. In an online retail system, the 2-Phase Commit protocol is used to ensure stock availability across warehouses. Describe the process and challenges if a warehouse node fails during the commit phase.

Distributed Algorithms

21. In a distributed chat application, explain how the Ricart-Agrawala algorithm ensures that users' messages are displayed without conflict in the chat window.
22. For a global file storage system, discuss how the Bully algorithm can assist in electing a central node to manage file indexing tasks.

Chord DHT and Peer-to-Peer

23. A music streaming service uses a peer-to-peer network with Chord DHT for efficient content location. Explain how Chord facilitates fast lookups and reduces latency for users accessing files across regions.
24. In a decentralized e-library, Chord DHT is implemented to locate books across multiple nodes. Describe how the protocol supports efficient search and retrieval in a large-scale environment.