

MAKE SURE THAT YOUR HANDWRITING IS READABLE

- 1a If a client did not receive a response in an RPC, explain what may have gone wrong, and how the client can see what the failure was. 6pt
Either (1) the client's request got lost, (2) the server crashed after the request was received, or (3) the response was lost. There is essentially no way that the client can detect exactly what the failure was.
- 1b Consider an RPC client who wants a remote process to act on a database local to the RPC client. What would be a good way to make this happen? 6pt
The client machine should have its local database server accept RPC calls from the server. In that case, the client can simply call the server the ID of the operation, along with suitable parameters that the server should use when executing the RPC to the database.
- 1c Consider a series of independent RPC calls from a single client to different servers. What would be a convenient way to maximize the client's perceived performance? 6pt
We need to have every RPC called by a separate thread thus realizing parallelization of the actual RPCs.
- 2a Explain what DNS redirection entails. 4pt
Based on the IP address of the requesting client/resolver, a DNS server may decide to return a specific IP address of the name being looked up, e.g. redirecting the client to a nearby replica server in a Web hosting service.
- 2b Content Delivery Networks (CDNs) tend to redirect clients to a replica server that is best for the CDN. Considering that ISPs usually host the client's DNS resolvers, how can an ISP deviate from the CDN's redirection policy? 8pt
Considering that an ISP needs to handle many client requests, they can gradually discover various replica servers from the same CDN and decide to return a different answer than the one returned by the CDN's DNS server.
- 2c What would be needed to provide a client the illusion that it is always viewing the latest version of a static Web page, assuming it accesses pages through edge servers? 6pt
The minimal requirement is that (1) a client always accesses the same edge server and that (2) there is no communication between clients accessing different edge servers.
- 3a Under which conditions will replication lead to a scale-down in performance? 4pt
When there are relatively many writes in comparison to reads, assuming that writes need to be first propagated to all replicas before the next operation can be executed.
- 3b Explain the necessary and sufficient conditions for quorum-based replication. 4pt
Let there be N servers, and N_R the number of servers accessed for reads, and N_W the number for writes. Then, $N_W > N/2$ and $N_R + N_W > N$.
- 3c Assume that the cost of a read operation to a single server is 1 unit, and that of a write operation 2. If the read-to-write ratio is equal to 4, what would be the optimal read quorum in a quorum-based replication system with 10 servers? 8pt
 $N_W > 6, N_W + N_R = 11$. If nr are the number of reads and nw the number of writes, the total costs are $1 \cdot nr \cdot N_R + 2 \cdot nw \cdot N_W = nr(11 - N_W) + 2 \cdot \frac{nr}{4} \cdot N_W = nr(11 - \frac{1}{2}N_W)$, which is minimal when N_W is maximal, i.e., 10.
- 4a Explain the working of a two-phase distributed commit protocol. Provide the state-transition diagrams for coordinator and participant. 6pt
See book.

- 4b What happens if a participant crashes and recovers while it was in state READY? Assume the coordinator does not maintain a log. 6pt

The issue with this question is that the participant should find out if and which state transition it should make. To this end, it can check in which state the other participants are. If one is in the ABORT or COMMIT state, it should move to that state as well. If all are in the READY state, it will have to wait as the coordinator is the only process who can actually take a decision.

- 4c Explain why failing in the READY state will never lead to loss of data. 6pt

Before entering the READY state, the participant must have taken all measures to allow crashing and recovery in that state. This includes making sure that it can indeed reenter the state and thus that all data it was working on can also be recovered. This can be done by having written updates to a workspace on stable storage.

- 5a Explain what topic-based publish-subscribe entails. How would you realize it when having a message broker at your disposal? 6pt

Topic-based pub-sub involves two types of processes. Publishers submit messages to the middleware system where each message is tagged with a topic essentially describing what the message is about. Subscribers submit requests for receiving messages on a specific topic. Having a message broker makes an implementation easy: all messages and subscriptions are stored centrally and the broker matches messages to the topics specified by the subscribers, after which a matched message is forwarded to the subscriber.

- 5b What is the essential difference between topic-based and content-based publish-subscribe?. Give an example of a subscription for a content-based publish-subscribe system. 8pt

The essential difference is that content-based pub-sub operates on the message content instead of tags describing the topic of a message. Assuming the message content consists of two integer variables x and y , a typical subscription could be " $x > 2$ and $y < 10$ ".

- 5c Replication of subscriptions in a publish-subscribe system is fundamentally problematic. Why? 6pt

*The real problem is caused by searching for messages and subscriptions that match. In a topic-based system it may be possible to combine hashing and replication (i.e., you could decide to replicate all subscriptions and messages on topic X to machines matching some hash value of X), but this cannot be done for content-based systems. As a consequence, you would need to search all servers when trying to match a subscription against a message. This simply does not scale. **Note:** This turned out to be a poorly formulated question. Better would have been to replace "Replication" by "Distribution".xs*

Grading: The final grade is calculated by accumulating the scores per question (maximum: 90 points), and adding 10 bonus points. The maximum total is therefore 100 points.