

Distributed Systems

Sample Questions

1. Describe precisely what is meant by a scalable system.
2. Why are transport-level communication services often inappropriate for building distributed applications?
3. Instead of letting a server register itself with a daemon as in DCE, we could also choose to always assign it the same end point. That end point can then be used in references to objects in the server's address space. What is the main drawback of this scheme?
4. Assume a client calls an asynchronous RPC to a server, and subsequently waits until the server returns a result using another asynchronous RPC. Is this approach the same as letting the client execute a normal RPC? What if we replace the asynchronous RPCs with asynchronous RPCs?
5. Using an auto-mounter that installs symbolic links as described in the text makes it harder to hide the fact that mounting is transparent. Why?
6. In NFS, attributes are cached using a write-through cache coherence policy. Is it necessary to forward all attribute changes immediately?
7. Explain how Coda solves read-write conflicts on a file that is shared between multiple readers and only a single writer.
8. Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?
9. To achieve totally-ordered multicasting with Lamport timestamps is it strictly necessary that each message is acknowledged?
10. Suppose that two processes detect the demise of the coordinator simultaneously and both decide to hold an election using the bully algorithm. What happens?