CSI3344 Principles of Distributed Systems

Workshop Solution 04

Q1. 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?

A: No, this is not the same. An asynchronous RPC returns an acknowledgment to the caller, meaning that after the first call by the client, an additional message is sent across the network. Likewise, the server is acknowledged that its response has been delivered to the client. Two asynchronous RPCs may be the same, provided reliable communication is guaranteed. However the approach of executing two asynchronous RPCs is different from an execution of a normal RPC.

Q2. With persistent communication, a receiver generally has its own local buffer where messages can be stored when the receiver is not executing. To create such a buffer, we may need to specify its size. Give an argument why this is preferable, as well as one against specification of the size.

A: Having the user specify the size makes its implementation easier. The system creates a buffer of the specified size and is done. Buffer management becomes easy. However, if the buffer fills up, messages may be lost. The alternative is to have the communication system manage buffer size, starting with some default size, but then growing (or shrinking) buffers as need be. This method reduces the chance of having to discard messages for lack of room, but requires much more work of the system.

Q3. Explain why transient synchronous communication has inherent scalability problems, and how these could be solved.

A: The problem is the limited geographical scalability. Because synchronous communication requires that the caller is blocked until its message is received, it may take a long time before a caller can continue when the receiver is far away. The only way to solve this problem is to design the calling application so that it has other useful work to do while communication takes place, effectively establishing a form of asynchronous communication.

END OF THE WORKSHOP SOLUTION