

FIT3143 Tutorial Week 4

Lecturers: ABM Russel (MU Australia) and Vishnu Monn (MU Malaysia)

SAMPLE SOLUTION

1. Discuss the three types of failure in IPC.
 - Lost request
 - Lost response
 - Crashed server
2. Discuss the pros and cons of one of the three different techniques for implementing concurrent access to multiple servers (threads, early reply, call buffering)
 - a) Threads - Very flexible and allows programmers to obtain optimal performance on the client side. Much more difficult to program and requires significant effort by the developer on the client side.
 - b) Early reply - Allows for overlap of computation on the client and server, however doubles the number of messages required to be sent. Also, potentially under-utilises the server machine.
 - c) Call buffering - Robust and fault tolerant technique that enables the client to assume no responsibility. Again, as above it allows concurrent computation on client and server and enables the buffer service to better utilise/manage the server's time. More messages required for a single operation.
3. Discuss how RPC performance could be improved.
 - Concurrent access to multiple servers
 - Serving multiple requests simultaneously
 - Reducing per-call workload of servers
 - Reply caching of idempotent remote procedures
 - Proper selection of timeout values
 - Proper design of RPC protocol specification
4. You have been asked to architect the software stack for a mobile application that requires running some compute intensive image processing algorithms. Discuss the appropriateness, advantages and disadvantages of either an RPC or IPC based solution.
 - a) IPC - Given the unreliable network connection an IPC approach over mobile network is probably not a good choice. It would require substantive effort on the part of the developer to ensure reliable computation. Potentially a local IPC approach could be beneficial, leveraging the multiple compute resources on the device. This

would also overcome any reliance on networks. Ultimately it would depend on the magnitude of compute required and the available power to support it.

b) RPC - An RPC approach would be far more tolerant of the unreliable network common to mobile platforms. Depending on the size of images being transferred over the network it could provide an ideal blend of local and remote compute (particularly if one of the concurrent access approaches discussed above are implemented). The main downside here is the sole reliance on a network connection. If the mobile device loses service the application will fail to operate, which is probably not a desired feature.