

9 Concurrent and Distributed Systems (RNW)

Remote Procedure Call (RPC) allows procedures (functions, methods) to be forwarded over the network, and is a fundamental building block of distributed systems such as the Network File System (NFS).

(a) Explain, with respect to a client, what it means for an RPC call to be:

(i) *Synchronous* [1 mark]

(ii) *Asynchronous* [1 mark]

(iii) *Idempotent* [1 mark]

(b) Distributed-filesystem clients utilize different tradeoffs between performance and consistency for operations on the directory namespace (e.g., file or directory creation) versus those on file data itself. The following program creates and opens a file `foo`, writes some data to it, and closes it, via NFSv3:

```
fd = open("foo", O_CREAT | O_RDWR, 0755);
// POINT A
write(fd, data, sizeof(data));
// POINT B
close(fd);
// POINT C
```

For each of points *A*, *B*, and *C* in the program, discuss whether or not another NFSv3 client is guaranteed to be able to see the results of each of `open` and (if it has been called) `write`. [5 marks]

(c) The developers of a new distributed filesystem use a *persistent log* to ensure *all-or-nothing* retry semantics for filesystem operations. However, this comes at a substantial performance cost. Describe two circumstances under which filesystem RPCs can be excluded from the log while maintaining correctness, and give an example of each. [4 marks]

(d) The distributed filesystem's persistent RPC log is maintained in a *write-ahead log* stored in local disk blocks. Explain why large RPCs may require special care in the server-side log implementation. [2 marks]

(e) RPC protocols and write-ahead logging both rely on unique ID numbers. Explain why reusing client-generated RPC IDs for server log transaction IDs could harm each of *correctness*, *scalability*, and *security*. [6 marks]