

9 Concurrent and Distributed Systems (RNW)

Java Remote Method Invocation (RMI) is a Remote Procedure Call (RPC) system that allows invocation of methods on objects in remote Java Virtual Machines (JVMs).

- (a) (i) Describe what it means for a class to be labelled as *serializable*. [2 marks]
(ii) Describe what it means for a class to be a *remote class*. [2 marks]
(iii) Describe the function of RMI's *object registry*. [2 marks]
- (b) RPC implementations are unable to implement idealised *exactly-once* semantics, instead providing *all-or-nothing*, *at-most-once*, or *at-least-once* semantics.
- (i) Describe the server-side state obligation for each of *all-or-nothing*, *at-most-once*, and *at-least-once* semantics. [3 marks]
(ii) Describe the classes of distributed filesystem RPCs in which it is safe to use *at-least-once* instead of *all-or-nothing* semantics. [2 marks]
(iii) A distributed filesystem stores object replicas on multiple servers for fault tolerance. Successful reads must be made to at least Q_r servers, and writes must be made to at least Q_w servers. The client submits requests to all servers simultaneously using a "MultiRPC". Describe when MultiRPC must resend (a) reads and (b) writes due to packet loss. [2 marks]
- (c) Java is a garbage-collected language: when an object is unreachable, it will be freed. Cycles can exist in which otherwise disconnected objects reference one another, preventing reference counts from reaching zero. Cycle detection addresses this problem: execution is suspended and the object-reference graph is searched for memory that can be recovered. *Distributed garbage collection* faces many of the same challenges: cycles may exist between objects on different servers, which cannot be detected by local garbage collectors in individual JVMs.
- (i) How does RMI track distributed references to a local object that has been exported to a remote client? [2 marks]
(ii) A developer implements a distributed garbage collector (GC) for RMI. The GC queries all nodes for graphs of local and remote references, identifying disconnected cycles spanning multiple JVMs. JVMs are notified of remote references that can be safely released to break cycles. Unfortunately, this does not work: when the system is busy, live references get broken, causing user-visible failures. Describe a scenario in which this algorithm, without additional synchronisation, may lead to incorrectness. [5 marks]