

7 Concurrent and Distributed Systems (RNW)

Correct handling of time is critical to the correctness (and performance) of distributed systems such as network file systems and databases. Consider the following two approaches.

(a) Physical clock synchronisation

(i) Define *clock skew* and *clock drift*. [2 marks]

(ii) A client running Cristian's Algorithm observes a local clock time of 1399157100.00s at the start of its RPC, and 1399157100.10s at the end of its RPC. The RPC returns a server timestamp of 1399157100.05s. What client-server clock skew will the algorithm calculate? Justify your answer. [2 marks]

(iii) Using Cristian's Algorithm as the underlying primitive, propose a time synchronisation algorithm that measures and compensates for clock drift. [2 marks]

(b) Distributed logical clocks

The *make* build tool relies on file-system timestamps to determine whether an object file is older than a source file: if so, the object file is rebuilt; if not, then a rebuild is avoided. However, it is common practice to store source code in a distributed file system and object files in a local temporary directory. The Network File System (NFS) stores file creation and modification times based on the server clock, whereas the local file system uses a local clock.

(i) Describe the two failure modes *make* may experience if client and server clocks are out of sync. [2 marks]

(ii) One solution is to use NTP to synchronise client and server clocks. Describe two reasons why this might work poorly in practice. [2 marks]

(iii) The NFS developers decide that *Lamport Clocks* may be able to solve the problem, as they track the *happens-before* relationship. During a particular run, *make* finds a source logical timestamp s is less than the object logical timestamp o . Explain why it is problematic to use Lamport Clocks to conclude that the object file should not be recompiled. [4 marks]

(iv) Explain why *Vector Clocks* might be more suitable than Lamport Clocks for this problem. [2 marks]

(v) Explain what a Vector-Clock-based *make* should do if there is no defined *happens-before* relationship between source vector s and object vector o . [4 marks]