Concurrent and Distributed Systems - 2017–2018

Supervision 3: RPCs and time

Q1 Definitions

Briefly define each of these terms, and give an example of its use.

(a) idempotent

(b) location transparency

(c) marshalling

(d) pure names

(e) impure names

(f) batching

(g) middleware

(h) clock skew

(i) clock drift

Q2 NFS

Write a 300-word description of what NFS is, how it works, when you might use it and why. Compare and contrast NFS versions 2 and 3.

Q3 Time

For each of the following potential uses of time, for which is a local oscillator sufficient, for which is synchronised real time across a distributed system sufficient (e.g., UTC as distributed using NTP), and for which is a logical or vector clock mechanism most appropriate? Explain your answers.

(a) local process scheduling

(b) local I/O

(c) distributed filesystem consistency consistency

(d) cryptographic certificate/ticket validity checking

(e) tracing causal links in a distributed application over many nodes

Q4 NTP

NTP is a simple protocol that is foundational to other distributed systems protocols (e.g. the Kerberos authentication protocol will not work if two machines' clocks are too far apart, so they must be synchronised first).

Write a Java program which acts as an NTP client and prints the current time, as estimated from the NTP server, to the console whenever it is run. For example:

```
bash$ java -jar current-time.jar
Fri Feb 21 11:01:23 GMT 2014
bash$
```

You will need to send UDP packets in an appropriate format to an NTP server. The Computer Lab has a set of NTP servers which you may wish to use:

```
ntp0.cl.cam.ac.uk
ntp1[bc].cl.cam.ac.uk
```

Please make sure your packets conform to NTP version 3 or later (see RFC 1305 or later) and that you don't send more than two packets to the server each time you run your program.

Q5 Vector clocks

Vector clocks are a technique for determining a causal ordering of events in a distributed system. They can also help to prevent certain (undesirable) orderings from occurring by holding back events. This question asks you to apply the general algorithm and also compare vector clocks with Lamport clocks. Note that this question refers to the form of vector clocks presdented in lecture, in which timestamps are updated on both message send and message receive.

(a) Given an initial state and the sequence of messages below, show the value of the Lamport and vector clocks at each node (A, B, C, D) at each send or receive event it participates in.

Event	Lamport				Vector			
	A	B	C	D	A	B	C	D
initial								
A → B			-	-			-	-
A → C		-		-		-		-
A → D		-	-			-	-	
B → D	-		-		-		-	
B → C	-			-	-			-
C → D	-	-			-	-		
C → A		-		-		-		-
A → B			-	-			-	-
C → B	-			-	-			-
final								

(b) Using the Lamport and vector clocks calculated above, state whether or not the following

events can be determined to have a *happens-before* relationship.

Events		Lamport	Vector
A $\rightarrow$ C sent	A $\rightarrow$ D sent		
A $\rightarrow$ D sent	B $\rightarrow$ C sent		
B $\rightarrow$ C sent	C $\rightarrow$ B sent		