UNIVERSITÄT SIEGEN

Fakultät IV
Betriebssysteme und verteilte Systeme
Prof. Dr. rer. nat. Roland Wismüller

Excercise Sheet 7
Solution

Lecture Distributed Systems
Winter Term 2023/24

Exercise 1: Clock Drift

a) A simple example is the distributed compilation of a program using $make$. Make is controlled by a so-called makefile, in which dependencies e.g. between source and object files are specified, as well as rules specifying how target files can be created from source files. For optimization reasons, such a rule is only applied if the source file has changed since the last time it was used, so that only really necessary compilation processes take place. Make is based on the time stamps of the files stored in the file system that indicate the last change. If the clocks on the individual computers involved deviate from each other, inconsistent states may occur, e.g. a source file may have an older date than the corresponding object file, even though the source code was changed after the last translation process. Erroneously, the recompilation of this file would not take place.

b) With the additional leap seconds inserted, some days take exactly 86401 seconds instead of 86400. This is necessary because an earth rotation does not take exactly $24 * 60 * 60$ seconds, but a small fraction of a second longer. So leap seconds have nothing to do with a possible inaccuracy of the atomic clock, but only serve to adapt to the calendar.

Exercise 2: Clock Synchronisation

There are several ways to synchronize distributed clocks. One of these is Cristian's protocol. A client process P requests the time in a message m_r from a time server S, and receives the time t in a message m_t from S. The problem is that because of the signal propagation time m_t, the time t is out of date again when arriving at P. Therefore, P measures the round trip time T_{round} from sending m_r to receiving m_t and sets its clock to $t + T_{round}/2$ ($T_{round}/2$ is the estimated propagation time of the message m_t).

The accuracy of the procedure is limited by:

- the assumption that the runtimes of m_r and m_t are the same (systematic error).
- the variations in the message durations (statistical error).

In a LAN, the variations (jitter) occurs due to competition for the medium (collisions, buffer times in switches) and due to the processing of messages in the operating systems of P and S. For a LAN, the accuracy is usually within $1ms$. In the Internet, additional inaccuracies occur because the messages are delayed in the routers. For WANs, the accuracy is probably within $5 - 10ms$.

Also with the help of GPS an exact synchronization of clocks is not possible. Due to the known position of satellites and the GPS receiver, the message transfer time can be calculated very accurately, but still varies, e.g., due to weather conditions. In addition, the positions are also afflicted with errors.

Exercise 3: Programming: Clock Synchronization

The programming solution for this task can be found in the archive `l07eFiles.zip`[1] on the lecture's web page.

Due to the different speed of the local clocks (drift), the synchronization must be repeated regularly.

[1] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/l07eFiles.zip`

Exercise 4: Clock Synchronization

a) The client should select the request with the minimum round trip time of $20ms$ = $0.02s$. If it can set its clock immediately after the response arrives, it would set its clock to $t_{Server} + T_{round}/2$, i.e. 11:51:25.232 + 0.02/2 = 11:51:25.242. Since the client can usually only determine the answer with the minimum round trip time later (after some requests), it must proceed as follows: Suppose the local time of the client is t. It then sets the clock to $t+(t_{server}+T_{round}/2-t_{arrival})$ = $t+(t_{server}+T_{round}/2-(t_{send}+T_{round}))$ = $t+25.242-24.734 = t+0.508$, i.e. the clock is advanced by $508ms$.

The accuracy is $\pm 10ms$.

If the minimum transfer time is $8ms$, then the setting remains the same, but the accuracy improves to $\pm 2ms$.

b) To synchronize the clock with an accuracy of $1ms$, if a transmission time of at least $8ms$ is given, the maximum round trip time is $T_{round} = 2 \cdot 1ms + 2 \cdot 8ms = 18ms$.

Exercise 5: Lamport and Vector Time (Mandatory exercise for 6 CP)

Exercise 6: *Happened-Before* Relation, Lamport and Vector Time (Mandatory exercise for 6 CP)

Exercise 7: Programming: Vector Clock

You will find the solution to this problem in in the archive `l07eFiles.zip`[2] on the lecture's web page.

[2] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/l07eFiles.zip`