UNIVERSITÄT SIEGEN

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

Excercise Sheet 11
Solution

Lecture Distributed Systems
Summer Term 2020

Exercise 1: Replication

a) The three most important points that speak in favor of generating replicas are reliability, data safety, and access speed.

b) Types of replicas:

- Permanent: permanently available replicas - initial set of replicas that make up a distributed data store (e.g. mirroring of web pages, replicated DNS servers).
- Server-initiated: initiated on demand by the server - copies of a data store that exist to enhance performance and which are generated by the initiative of the owner of the file storage (e.g. web hosting services, content delivery networks).
- Client-initiated or client caches: created by the client - local storage facility used by the client to temporarily store a copy of the previously requested data.

Exercise 2: Active Replication

No. For example, take read operations that take place on unmodified data, or commutative write operations. In principle, these situations allow different sequences to exist on different replicas. However, it can be difficult, if not impossible, to determine whether, for example, two write operations are commutative.

Exercise 3: Quorum Protocols

The read and write quorum must meet the following two conditions (where v is the total number of votes available):

(1) $r + w > v$

(2) $w > v/2$

Condition (1) ensures that the intersection between each read and write quorum is not empty, i.e. in a partitioned system an object cannot be read in one partition and written in another.

Condition (2) prevents write-write conflicts.

Since each replica has exactly one vote, $N = v$ applies.

a) $N = 10$, $r = 6$, $w = 5$

Here condition (2) is violated; write-write conflicts are not detected.

b) $N = 10$, $r = 3$, $w = 7$

Here condition (1) is violated; read-write conflicts are not recognized.

c) $N = 10, r = 1, w = N$

This is a correct quorum protocol. By $r = 1$ and $w = N$ one replica is sufficient for reading, whereas all replicas are needed for writing. Under the assumption that read accesses occur (much) more frequently than write accesses, this protocol is runtime-optimal. In the case of network partitioning, however, write accesses are no longer possible.

(This protocol is also known as '*write locks all, read locks one*'. It corresponds to the procedure shown for primary-based protocols for sequential consistency.)

d) $N = 10, r = 5, w = 6$

This is also a correct quorum protocol. By $r = 5$ and $w = 6$ read accesses become more complex (in comparison to protocol c), however write accesses in network partitions are possible starting from six nodes.

Exercise 4: Quorum-based Replication

The following choices (write quorum N_W, read quorum N_R) are allowed: (10,1), (9,2), (8,3), (7,4) and (6,5).

$N_W \leq 5$ is not allowed, otherwise write conflicts may occur. Of course, higher values for N_R would also be possible for each N_W, but this is not really useful.

Exercise 5: Programming: Consistency Protocols

a) You will find the solution to this problem in the archive `l11eFiles.zip`[1] on the lecture's web page.

b) The difficulty in implementing a *local-write* protocol lies in finding out which server is the current primary server, i.e. where the primary copy of x is currently located. For this further mechanisms like location services have to be implemented.

Exercise 6: Any Questions?

[1] `http://www.bs.informatik.uni-siegen.de/web/wismueller/vl/vs/l11eFiles.zip`