UNIVERSITÄT
SIEGEN

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

Excercise Sheet 10
Solution

Lecture Distributed Systems
Winter Term 2024/25

Exercise 1: Totally Ordered Multicast / Lamport time stamps

No, it is sufficient to multicast any other message type as long as this message has a timestamp greater than that of the received message.

Exercise 2: Transactions

Transactions are a concept closely related to mutual exclusion algorithms because, like these algorithms, they protect a shared resource against simultaneous access by multiple processes. Furthermore, transactions can allow a process to access and modify multiple data elements within a single atomic operation. If the process is interrupted during the transaction, everything is restored as it looked before the transaction started.

a) The most important characteristics of transactions are:

- Atomicity - For the outside world, the transaction will be carried out indivisibly.
- Consistency - The transaction does not violate any system invariants.
- Isolation - Concurrent transactions do not affect each other.
- Durability - After a transaction is committed, the changes remain permanently.

b) A deadlock can occur in the context of concurrency when two transactions wait for each other to release a resource. Since the goal of concurrency is to execute multiple transactions at the same time while maintaining a consistent status of all data elements, a concurrency control is necessary here.

Exercise 3: Two-Phase Commit

A participant could wait in its INIT status for a VOTE-REQUEST message from the coordinator. If this message is not received after a certain time, the participant simply decides to cancel the transaction locally and sends a VOTE-ABORT message to the coordinator.

Similarly, the coordinator can block in the WAIT state, where it waits for the other participants to vote. If not all votes can be determined within a certain time, the coordinator should also decide to cancel the transaction and send a GLOBAL-ABORT to all participants.

The third status in which it is possible to block is the READY status of the participants. Here you could block because a participant is waiting for the global voting result sent by the coordinator. If this message is not received within a certain time, the participant cannot simply decide to cancel the transaction. Instead, it must determine which message the coordinator has sent. The easiest way to solve this problem is to block each participant until the coordinator is up and running again.

Exercise 4: Consistency Models (Mandatory exercise for 6 CP, submit via moodle!)

Exercise 5: Sequential Consistency (Mandatory exercise for 6 CP, submit via moodle!)

Exercise 6: Causal Consistency

a) P_2: Reading x and writing y creates a causal dependency. I.e., $W_1(x)1$ is causally before $W_2(y)2$. But to explain the behavior at P_3, the global order would have to be $W_2(y)2$, $R_3(y)2$, $R_3(x)0$, $W_1(x)1$, which contradicts this ($W_1(x)1$ is after $W_2(y)2$).

b) The causal consistency may be sufficient. The problem is that reactions to changes in stock values should be consistent. Changes to independent stock values can be displayed in different order.