TUM I 13
Chair of Application and Middleware Systems
Prof. Dr. H.- A. Jacobsen

Tutorial
Distributed Systems (IN2259)

Sample Solution: Exercises on Consistency in Distributed Systems

Note: this sample solution is only a suggestion to solve the assignment; there might be various other possible solutions. Also note that this sample solution might still contain errors or other fallacies.

Exercise 1 Data-centric consistency

Assume a system with N processes P_1 - P_N and shared resources x, y, z. The processes only communicate through the resources.

Example: In Figure 1.1, you see four operations executed by the processes P_1, P_2 and P_3. $w(x)a$ in Line P_1 means that process P_1 writes the value a to resource x. $r(x)a$ in Line P_3 means the value a is read by P_3 from resource x.

The processes are running their operations concurrently. In most cases, there are multiple execution sequences that can be derived from the diagrams. An execution sequence represents one possible execution order of the processes. Depending on the consistency model, different execution orders are possible. An execution order is valid if the read and write semantics are modeled correctly. This means every read on a resource is preceded by the corresponding write. Two possible execution orders in Figure 1.1 are:

1. $w(x)a\ r(x)a\ w(x)b\ r(x)b$ is a valid execution sequence meaning that every read is preceded by the corresponding write.
2. $w(x)a\ w(x)b\ r(x)a\ r(x)b$ is not a valid execution sequence because of $w(x)b\ r(x)a$.

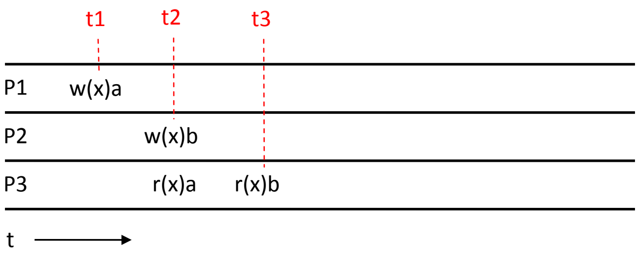


Figure 1.1: Example

(a) Strict Consistency means that any read on a resource x returns a value corresponding to the result of the most recent write on x. This definition implicitly assumes the existence of absolute global time. Is Figure 1.2 strictly consistent? If yes, find a valid execution sequence to prove it. If not, give an explanation.

Solution: $w(x)a\ r(x)a\ w(x)b\ r(x)b$

(b) Linearizability means that all operations are ordered according to their real-time timestamp in the execution sequence. We assume ordering according to a set of synchronized clocks. This implies the operation can be executed at an arbitrary point within a given interval. Is Figure 1.3 lineralizable? Is Figure 1.3 strictly consistent, assuming each operation is performed at the start of its time interval? If yes, find valid execution sequences to prove it. If not, give an explanation.

Solution: As the physical clocks of the processes in the system are synchronized up to a bounded error the execution of operation can happen at any point in the given interval.

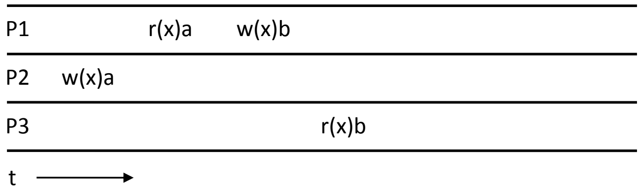


Figure 1.2: Strict Consistency

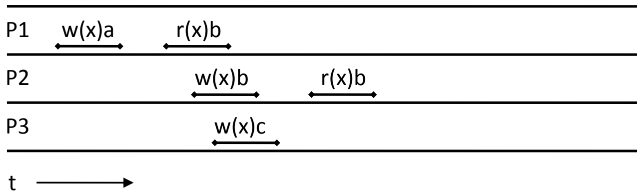


Figure 1.3: Linearizibility

- $w(x)a\ w(x)c\ w(x)b\ r(x)b\ r(x)b$ is a correct linearized sequence.
- Strict consistency is violated as sequence $w(x)a\ r(x)b$ in the beginning is not valid.

(c) Sequential Consistency means that any valid interleaving of read and write operation is acceptable behavior, but all processes see the same interleaving of operations. Nothing is said about physical time. Is Figure 1.4 sequentially consistent? Is Figure 1.4 linearizable? If yes, find valid execution sequences to prove it. If not, give an explanation. Consider the intervals only to check linearizibility.

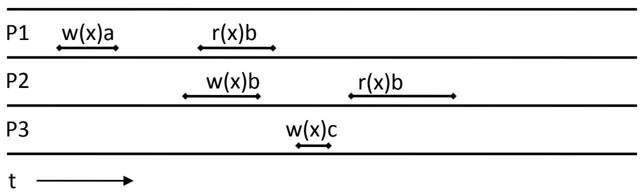


Figure 1.4: Sequential Consistency

Solution:

- $w(x)a\ w(x)c\ w(x)b\ r(x)b\ r(x)b$
- Assuming a global time, the ordering of events $w(x)b\ w(x)c\ r(x)b$ cannot be changed. Therefore every execution order is invalid and linearizability cannot be achieved.

(d) Causal consistency means that concurrent writes do not need to be seen in the same order by all readers. Causally related writes, however, must be seen in the same order by every process. This means processes can possibly have different execution sequences. Nothing is said about time. Is Figure 1.5 causally consistent? Is Figure 1.5 sequentially consistent? If yes, find valid execution sequences to prove it. If not, give an explanation.

Solution: Due to the following valid execution sequences for processes P1, P2, P3 the situation is causally consistent

- P1: $w(x)a\ w(x)c\ w(x)b\ r(x)b$
- P2: $w(x)a\ w(x)b\ w(x)c\ r(x)c$

P1	w(x)a	r(x)b
P2	w(x)b	r(x)c
P3	w(x)c	r(x)a

t ⟶

Figure 1.5: Causal Consistency

- P3: $w(x)b\ w(x)c\ w(x)a\ r(x)a$

For sequential consistency concurrent writes need to be seen in the same order by all readers. For example the schedule $w(x)a\ w(x)b\ r(x)b\ \underline{w(x)c}\ r(x)c\ \underline{r(x)a}$ is invalid. Similar, in every combination of write operations there will always be one read operation that cannot be satisfied, therefore all execution sequences are not valid and the situation is not sequentially consistent.

(e) Is Figure 1.6 causally or sequentially consistent? If yes, find valid execution sequences to prove it. If not, give an explanation.

P1	w(x)a				
P2		r(x)a	w(x)b		
P3		r(x)a	w(x)c		
P4				r(x)b	r(x)c
P5				r(x)c	r(x)b

t ⟶

Figure 1.6: Causal Consistency 2

Solution: The write operations $w(x)b$ by P2 and $w(x)c$ by P3 are concurrent although they are causally related to $w(x)a$. Therefore, the following valid execution sequences for P4 and P5 show that the situation is causally consistent

- P4: $w(x)a\ w(x)b\ r(x)b\ w(x)c\ r(x)c$
- P5: $w(x)a\ w(x)c\ r(x)c\ w(x)b\ r(x)b$

For sequential consistency concurrent writes need to be seen in the same order by all readers. As different write orders for $w(x)b$ and $w(x)c$ are necessary at P4 and P5 to provide the correct read order this situation is not sequential consistent.

(f) FIFO consistency means write operations by a single process are seen by all other processes in the order in which they were issued, but writes from different processes may be seen in different orders by different processes. Is Figure 1.7 FIFO? If yes, find valid execution sequences to prove it. If not, give an explanation.

Solution: The order of write operations $w(x)a$ and $w(x)c$ at P1 is also reflected in the read operations at P3, which makes the situation FIFO consistent.

$w_1(x)a\ r_2(x)a\ r_3(x)a\ w_2(x)b\ r_1(x)b\ r_2(x)b\ w_2(x)d\ r_3(x)d\ w_1(x)c\ r_3(x)c$

EXERCISE 2 Client-centric consistency

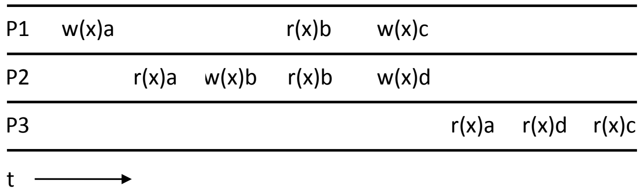


Figure 1.7: FIFO Consistency

Client-centric consistency considers consistency from the client's perspective. Client-centric consistency models apply in a mobile context. In Figure 2.8 you see six operations executed by different processes at the locations $L1$, $L2$ and $L3$. $w(x1)$ at Line $L1$ means that version $x1$ of resource x is written to the local replica at location $L1$. $r(x1)$ means that version $x1$ of resource x is read. This may for example represent a mobile device that contacts a local server to fetch emails of a user. After that the user changes the location (dotted line) and a different replica is contacted to fetch the emails again.

(a) Monotonic-read means that once read, subsequent reads on that data item return the same or a more recent value. Do you see any problems why monotonic-read in Figure 2.8 could not be guaranteed. If yes, how would you correct Figure 2.8?

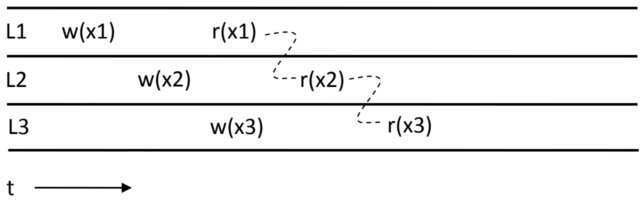
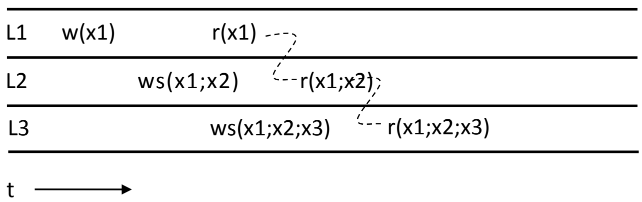


Figure 2.8: Figure 2.8: Monotonic-reads

Solution: The mobile client has to read a consistent version of data. Therefore the writes at location L1 have to be reflected at L2. Referring to the email example: The client wants to retrieve all emails when synchronizing with a local replica and not only parts of it.



(b) Monotonic-write means that a write operation by a process on a data item x is completed before any subsequent write operation can be issued on x by the same process. Do you see any problems why monotonic-write in Figure 2.9 could not be guaranteed? If yes, how would you correct Figure 2.9 ?

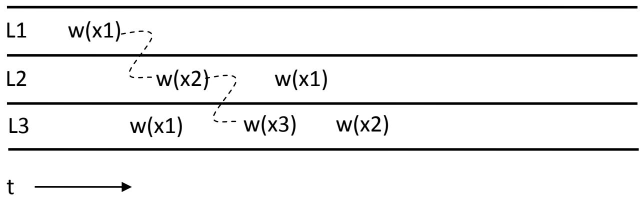
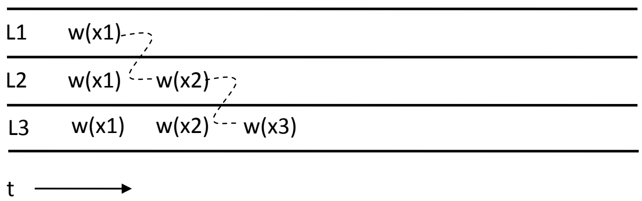


Figure 2.9: Figure 2.9: Monotonic-writes

Solution: In fact the write to the replica at location L1 is reflected in the update operations at location L2 and L3. The problem here is that at L2 the writes are executed in the wrong order. Since writes in different orders can lead to different results, monotonic writes cannot be guaranteed.



(c) Read-your-writes means that the effect of a write operation by a process on data item x will always be seen by a successive read operation on x by the same process. Do you see any problems why read-your-writes in Figure 2.10 could not be guaranteed? If yes, how would you correct Figure 2.10?

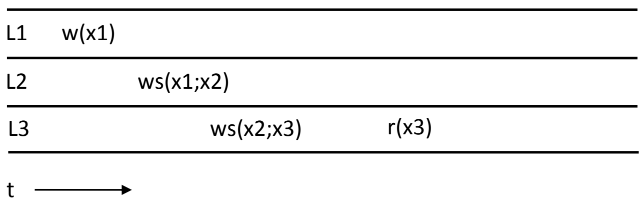
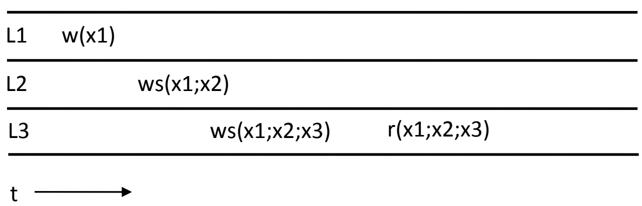


Figure 2.10: Read-your-writes

Solution: The write operation at L1, i.e., $w(x1)$ is not reflected at L3. Therefore, the situation is not read-your-writes consistent. In order to correct the situation, $w(x1)$ must be added to the write set at L3.



(d) Writes-follow-reads means that a write operation by a process on a data item x following a previous read operation on x by the same process, is guaranteed to take place on the same or a more recent value of x that was read. Do you see any problems why writes-follow-reads in Figure 2.11 could not be guaranteed? If yes, how would you correct Figure 2.11?

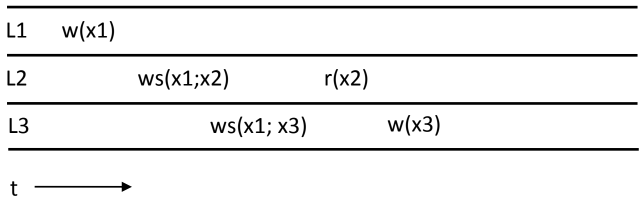


Figure 2.11: Writes-follow-reads

Solution: The write operation that led to the value read in L2 is not reflected at L3. Therefore, the situation is not writes-follow-read consistent. In order to correct the situation, $w(x2)$ must be added to the write set at L3.