

CS582 Distributed Systems

Quiz 7 Solution (20 marks)

1. In Dynamo, N denotes the number of replicas, R denotes the read quorum, and W denotes the write quorum. For a fixed value N , R and W can be varied to achieve different levels of consistency, availability, performance, and durability suitable for a particular environment. For the following scenarios (assuming $N = 5$), suggest the most optimal value of R and W , while justifying your choice to receive full credit.

- a) For a GET request, read the value from the closest replica to minimize delay, making sure the value is not stale.

Solution:

$R = 1$

(0.5 marks)

$W = 5$

(0.5 marks)

Since we can only read from the closest replica, $R = 1$, and to ensure that this replica never returns a stale value, we must write to all replicas, so $W = 5$

(1 mark)

- b) Speed up PUT requests as much as possible while making sure GET requests never return stale values.

Solution:

$R = 5$

(0.5 marks)

$W = 1$

(0.5 marks)

To speed up PUT requests as much as possible, we can write to just one replica, so $W = 1$, but for non-stale values to be returned, we must read from all replicas, so $R = 5$

(1 mark)

- c) Minimize read and write latency while achieving high durability, ensuring data is not lost and stale values aren't read even if two replicas fail.

Solution:

$R = 3$

(0.5 marks)

$W = 3$

(0.5 marks)

To maintain the durability of writes, we must replicate writes to at least more than two out of five replicas, so $W = 3$. Moreover, we should read from at least three replicas to ensure both read and write quorums intersect in order to avoid stale reads, so $R = 3$

(1 mark)

2. In Memcache, the concept of leasing is used to tackle the thundering herds problem. Consider the following scenario involving Temucache, a Memcache clone with different but self-explanatory APIs, and three clients $C1$, $C2$, and $C3$:

At $T = 0$, the database has the key-value pair $[X, 30]$, while the Temcache server does not have the key X .

$T = 1$ $C1 : \text{temu_get}(X)$

$T = 2$ $C2 : \text{temu_get}(X)$

$T = 3$ $C1 : \text{db_get}(X)$

$T = 4$ $C3 : \text{db_put}(X, 60)$

$T = 5$ $C3 : \text{temu_delete}(X)$

$T = 6$ $C1 : \text{temu_put}(X, 30)$

$T = 7$ $C2 : \text{temu_get}(X)$

$T = 8$ $C2 : \text{db_get}(X)$

$T = 9$ $C3 : \text{temu_get}(X)$

$T = 10$ $C2 : \text{temu_put}(X, 60)$

$T = 11$ $C1 : \text{temu_get}(X)$

At the following points, is any value returned? If yes, state the value. Otherwise, explain why no value is returned.

- a) $T = 3$

Solution:

30

(2 marks)

- b) $T = 7$

Solution:

No value is returned as it is a cache miss.

(0.5 marks)

Because at $T = 4$, C3 put the key-value pair $[X, 60]$ into the database, thus deleting $[X, 30]$ from the Temucache server and invalidating C1's lease.

(1 mark)

Hence, no key-value pair is inserted into the Memcache server at $T = 6$, leading to a cache miss when C2 tries to read X at $T = 7$.

(1 mark)

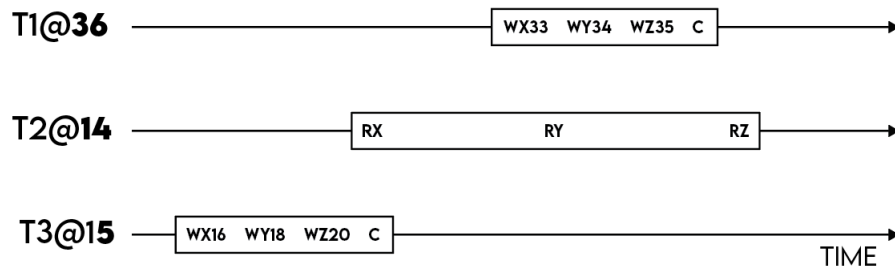
c) $T = 11$

Solution:

60

(2 marks)

3. Consider the following scenario involving Snapshot Isolation where the initial values of X, Y , and Z are 6, 7, and 8 respectively (written and committed by a much older transaction T_0).



- a) Assuming clocks are synchronized, and without accounting for Replica Safe Time, what value of X, Y , and Z are read by transaction T_2 ?

Solution:

$X = 6$

(0.5 marks)

$Y = 7$

(0.5 marks)

$Z = 8$

(0.5 marks)

- b) Maintaining the same assumption as in (a), if T_2 has the timestamp 44 instead of 14, what values of X, Y , and Z are read by T_2 ?

Solution:

$X = 33$

(0.5 marks)

$Y = 34$

(0.5 marks)

$Z = 35$

(0.5 marks)

- c) In Spanner, R/O transactions and R/W transactions are timestamped differently. If R/W transactions are timestamped like R/O transactions due to a glitch, what values of X, Y , and Z will T_2 read? Does this violate serializability and linearizability? Explain.

Solution:

$X = 6$

(0.5 marks)

$Y = 18$

(0.5 marks)

$Z = 20$

(0.5 marks)

Serializability will be violated

(0.5 marks)

Since T_2 is reading partially from T_0 and T_3 , we can not order them in any way

(1 mark)

Linearizability will be violated

(0.5 marks)

Since there is no serial order, let alone a real-time order

(1 mark)