CS 425: Distributed Systems
Homework 2

Net ID: ps71

5/6/2024

Question 1: Walt Disney and Pokemon wants to build a virtual theme park where young customers will all wear VR goggles and be connected via a Gnutella P2P system. At one point of time, the Gnutella topology looks like a virtual ring with 15 processes, with each process connected to its immediate three clockwise neighbors and immediate three anticlockwise neighbors (thus each process has six neighbors). All links are bidirectional. Note that any node can be the original query-sending node (for each of the three parts below). Answer three parts:

Part (a): A process sends a Query message with TTL=4. How many of the processes, apart from the sender, receive this Query message?

We assumed that the message was transmitted with a TTL, and then it the TTL decreases by 1.
Original sender node received the message with TTL:4
New nodes that received message with TTL:3 = 6
New nodes that received message with TTL:2 = 6
New nodes that received message with TTL:1 = 2
Total nodes to receive the message = 14

Part (b): What is the minimum TTL in a Query message to ensure all nodes in the system receive the Query (no matter the timing of the Query forwarding)?

The minimum TTL to ensure all the other nodes in the Gnutella ring gets the message is 3, as seen in the above part if the question.

Part (c): If we add a 16th process that is directly connected to all the old 15 processes, what is the minimum TTL in a Query message to ensure all nodes in the system receive the Query?

The minimum TTL required is 2, as once the sender sends the message to the 16_{th} node with a TTL of 1, it can then still transmit it to all the remaining nodes in the ring in the next transmission cycle.

Question 2: Three wealthy industrialists, Lo Skum, Frank Bozo, and Sugar Mountain, all want to make a movie aboard a very unsafe spaceship called "Solargate" that travels to all 8 planets (go figure!). Anyway, you are responsible for managing the spaceship as long as it is active. Lo and behold, the different planets use BitTorrent! While working through an example, you find a case where a file has 7 shards: A, B, C, D, E, F, G. The querying node (peer, leecher) is on a spacecraft, connected to all the planets directly (the SBM group figured out a way to have high bandwidth, low latency communication that beats the speed of light!). The planet servers at the 8 planets each have the following shards: Mercury (ACEG), Venus (BDF), Earth (ABCDEFG), Mars(ABCDEF), Jupiter (ABCDE), Saturn (ABCD), Uranus (ABC), Neptune (AB).

Part (a): Which shard will be fetched first?

1. Initial condition: since the file has seven shards/blocks: A,B,C,D,E,F,G and there are 8 nodes/peers in the system currently, excluding the leecher node/peer.
2. The shard least replicated on the 8 peers will be fetched first, i.e. 'G'

Part (b): What is the order in which shards will be fetched by the querying node? (assume that no shards are being fetched by any other nodes). If there are ties, you can use the alphabetically lower one.

G -> F-> E -> D -> C -> A -> B

Part (c): If one introduced a new renegade planet called DeGrasseTyson (and the spacecraft was connected to it), assign a set consisting of just one shard to DeGrasseTyson that changes the first fetched shard by the spacecraft (ties broken by lower letter)?

If we assign the shard 'G'top this new planet, then as a leecher peer the spaceship has to first fetch the shard 'F'now due to same replication factor but coming first in alphabetical order.

Question 3: One of the Producers, Leo Bloom, and his start Orlando Bloom, both like Bloom filters. But being a producer, he wants to create a new Bloom filter-based data structure. A (regular) Bloom filter's false positive rate is given as $\left(1 - e^{\frac{-kn}{m}}\right)^k$ where k is the number of hash functions, n is the size of the input set and m is the size of the Bloom filter in bits. Leo says that instead of using a single Bloom filter B with 4096 bits and 2 hash functions, his new datastructure, called Leo Bloom filter, uses 4 Bloom filters B1, B2, B3, and B4, each with 1024 bits, and each using 2 hash functions (each hash function different from each other, and different from the above 2 hash functions). There are two variants: When checking for an item, it returns true only if the item is present in Variant A: all, Variant B: any of B1, B2, B3, and B4. When inserting an item, for both variants A and B, it is inserted into all of B1, B2, B3, and B4. Which of the above three approaches— original using Bloom filter B vs. Leo–any vs. Leo–all —gives the best (lowest) false positive rates? Answer this for two cases: (1) when there are typically 10 elements inserted into the datastructure, (2) when there are typically 1000 elements inserted into the datastructure. (We recommend though, that you solve the problem with the variables k, n, m, etc., and then apply these values. But solving with only these two values of 10, 1000, would be ok as well.)

1. Using the FPR = $\left(1 - e^{\frac{-kn}{m}}\right)^k$ we can say that the FPR is inversely proportional to the filter size, and directly proportional to m(hash functions) and n(size of input set).
2. For 10 elements: The best filter is Leo-all Bloom filter
   - Original filter: FPR = 2.3726×10^{-5}
   - Leo-any filter: FPR = Union of all 4 $P(1 \text{ Leo-filter}) = 0.001495$
   - Leo-all filter: FPR = Intersection of all 4 = product of all 4 FPR = $1.9565295 \times 10^{-14}$
3. For 1000 elements: The best filter is Original Bloom filter
   - Original filter: FPR = 0.1492
   - Leo-any filter: FPR = Union of all 4 $P(1 \text{ Leo-filter}) = 0.995$
   - Leo-all filter: FPR = Intersection of all 4 = product of all 4 FPR = 0.294

Question 4: The director, C. Nolan, likes to deal with time travel, which means he asks a lot of "What if?" questions. He asks you several questions about Cassandra – for each of these, give one advantage and one disadvantage:

Part (a): What if there were no Bloom filter (but the rest of the system works the same way)?

Advantage: Simplified System Design: Removing the Bloom filter from Cassandra simplifies the system's architecture. Bloom filters are probabilistic data structures used to test whether an element is a member of a set and are primarily used in Cassandra to prevent unnecessary disk reads on non-existent data. Without them, the system design becomes less complex, potentially making it easier to understand, maintain, and modify the database's internal mechanisms.
Disadvantage: Increased I/O Operations: The primary role of the Bloom filter in Cassandra is to reduce the number of disk reads by checking the presence of the requested key in the filter before accessing the SSTables on disk. Without Bloom filters, Cassandra would need to perform disk accesses for every read operation, even if the data does not exist. This results in a significant increase in I/O operations, leading to slower query performance and higher latency in data retrieval.

Part (b): What if Cassandra did not use Memtable at all, and instead wrote directly into the "latest" SSTable? (but the rest of the system works the same way)

Advantage: Immediate Durability of Writes: Writing data directly to an SSTable would ensure that all written data is immediately persisted to disk. This could enhance data durability, as there would be no data residing in volatile memory (like a Memtable) that could potentially be lost in the event of a power failure or system crash.
Disadvantage: Increased Write Latency and Reduced Throughput: One of the primary roles of the Memtable in Cassandra is to buffer write operations in memory, which allows for fast write performance because writing to memory is much quicker than writing to disk. Memtables are periodically flushed to SSTables in an efficient, batched manner. Without the use of Memtables, every write operation would need to be directly written to an SSTable on disk, significantly increasing the write latency due to slower disk I/O operations. This method would also cause more frequent compaction processes and potentially lead to greater wear on the physical disks. Additionally, because SSTables are immutable once written, this approach would necessitate frequent creation of new SSTables for each write or a complex mechanism to append to "latest" SSTables, complicating the storage engine architecture and potentially leading to fragmentation and inefficient use of disk space.

Part (c): What if SSTables were mutable, i.e., when a Memtable were flushed, instead of creating a new SSTable, it would go and modify the corresponding values in already-present SSTables? (but the rest of the system works the same way)

Advantage: Reduced Number of SSTables and Compaction Overhead: By allowing in-place updates, Cassandra could potentially reduce the number of SSTables stored on disk. This could lessen the overhead associated with compaction processes, as fewer SSTables would need to be merged. The reduction in compaction could also lead to improved overall performance by minimizing the I/O and CPU resources required for merging SSTables and maintaining read performance.
Disadvantage: Increased Complexity and Potential for Data Corruption: Mutable SSTables would greatly complicate the storage engine's architecture. In Cassandra's current design, immutability of SSTables simplifies the management of data, especially in distributed environments. It avoids issues such as concurrency conflicts and data corruption that can arise from in-place updates. With mutable SSTables, Cassandra would need sophisticated locking or concurrency control mechanisms to handle simultaneous writes and reads, which could increase the complexity of the codebase and the likelihood of bugs or data corruption. Additionally, mutable files might be more susceptible to corruption in cases of system crashes or hardware failures, especially if changes are being written in-place without atomic operations or sufficient transaction logging.

Question 5: Help, the stars are stuck in a Black Mirror episode. In a system of N processes (N large enough, etc.), k quorum sets Q1, Q2, Q3, Qk are selected in an arbitrary manner, each Qi of the same size M. k is a constant, much smaller than N. What should this minimum value of M be so that at least one (any) process belongs to ALL three of Q1, Q2, Q3,…Qk (that is, the intersection of all k sets is non-null)? If this problem is too abstract (and daunting), we recommend you start with k=2 (as discussed in class), and quantitatively reason/think about why it has to be N/2+1. Then solve for k=3, then solve k=4, and so on. And then notice the patterns

For the requirement that we have intersection of processes non-null in k-Quorum we can mathematically write the requirement as: $k * M > N$
This implies:
$k > N/M$ $k >= N/M + 1$

Question 6: One of the actresses, named Meryl, is consistently a good actress and consistently wins awards. It's no surprise that she is very interested when you tell her about consistency models. She asks you about the differences between linearizability, sequential consistency, 4 causal consistency, and eventual consistency (for key-value stores with get/put operations on keys).

Part (a): Can you say briefly, and clearly what the differences are between the above models?

Linearizability ensures operations are instantly consistent globally, appearing instantaneous. Sequential consistency preserves operation order across all processes. Causal consistency ensures order only for causally related operations. Eventual consistency promises data consistency at some future time, without immediate guarantees. Each model balances strictness and performance differently.

Part (b): Give an example (using 2-3 clients writing and reading objects), where, for a particular read, using each of these models above gives a completely different return value (so 4 different answers). While you can search the Web to clarify differences between these models, you cannot borrow an example from the Web. Be concise

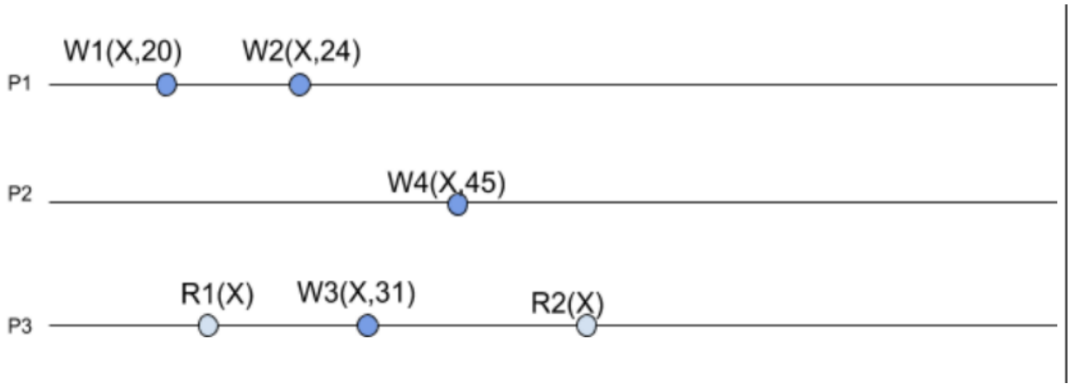

Figure 1: 3 Processes accessing key 'X'

Wi implies write Operation and Ri implies read operation in the above diagram.If we consider R2(X), it will return different values based on the consistency Model.

Linearizability - R2(X) returns 45 as the last write as per the global clock is W_4
Sequential Consistency - R2(X) can return 24(Since W_1 happens before W_2 in P_1, per process ordering is respected)
Causal Consistency - R2(X) can return 31(Causal Ordering Preserved by reading W_3)
Eventual Consistency - R2(X) could return any of the values, like let's say 20, since in an eventually consistent system,factors like network delays or timing of updates can cause the read operation to return different values

Question 7: An overly-religious person suddenly wants to fund the show, and they only worship Cristian's algorithm. They find that the round-trip time for one round of synchronization messages is exactly 0.78 ms. They are trying to calculate the error for this run, and so they calculate some minimum delays. On the server side, they find that there is a delay of at least 56.0 microseconds for a packet to get from an application to the network interface, and the delay on the opposite path (network interface to application buffer) is at least 0.12 ms. On the client side, they measure that the minimum time to get from the network interface to the application buffer is at least 0.34 ms, and the minimum time on the reverse path is X microseconds. But they forget to measure X. They figure that for the error in this Cristian's algorithm run to be zero, X should lie in an interval [A microseconds, B microseconds]. That is, A and B are respectively the minimum and maximum values possible for X so that the error for this run is zero. What should the values of A and B be, given these measurements?

For Chritian's algorithm, with know minimum latencies

$$App \rightarrow NWmin1 = 56micros,$$
$$NW \rightarrow Appbuffermin2 = 120microsecond,$$
$$RTT = 780microsecond$$

Client:

$$Appbuffer \rightarrow NWmin2 = Xmicrosecond$$
$$NW \rightarrow Appbuffermin2 = 340microsecond$$

For error to be zero, the error bound should be zero:

$$RTT - (56 + 340) - (X + 120)/2 = 0$$
$$780 - 396 - X - 120 = 0$$
$$X = 264microsecond$$

Client-side delay from application to the network interface for 0 error:

$$X \in [264microsecond, 264microsecond]$$

Question 8: Katherine Johnson was a Black computer mathematician and scientist who worked at NASA in building/writing the first NASA computer programs to do mathematical calculations for the geometrical trajectories of the Apollo moon spacecraft, being one of the first to write computer programs involving space and time. The astrophysicist consultant for your current film, a Dr. N. Tyson, is also a big fan of space and time. He is intrigued by the NTP algorithm. He poses the following question – suppose you are dealing with a network where latencies are symmetrical (e.g., an enhanced and controlled ATM network), i.e., latencies are identical in reverse directions, i.e., $A \rightarrow B$ message latency and $B \rightarrow A$ message latency (for same message) are known to be identical (except of course you don't know their absolute values), for every pair of processes A,B. From the equations, for NTP, can you derive the new error bound? Don't just show the last portion, please work through the entire derivation just like in the slides, from the beginning to the end.

The new error is o.
Let L be the 1-way latency, in both directions:

$$tr1 = ts1 + L + oreal$$

$$tr2 = ts2 + L - oreal$$

Subtracting, we get:

$$tr1 - tr2 = ts1 + L + oreal - (ts2 + L - oreal) = ts1 - ts2 + 2oreal$$

Because o = (tr1 - tr2 + ts2 - ts1)/2, we plug in to see that o = oreal, so the error is o.
Intuitively, the symmetry guarantee means the latencies in each direction cancel and we can isolate the skew.

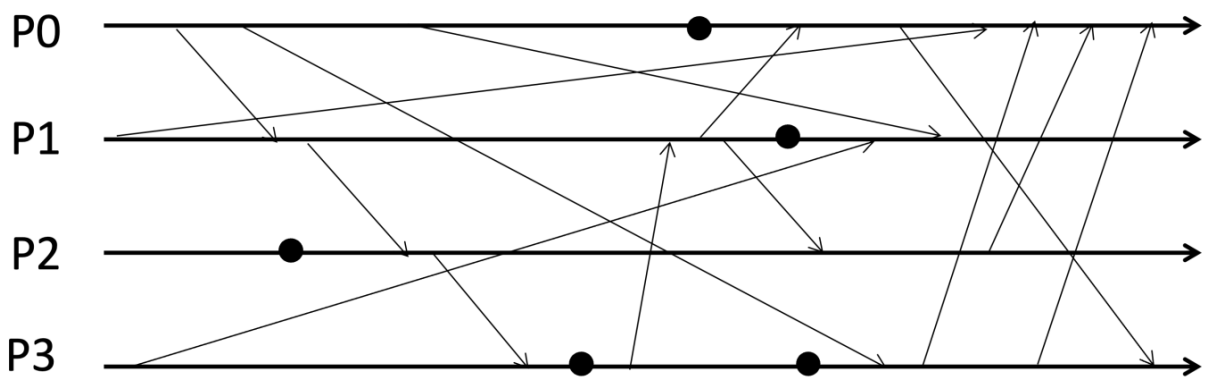

Figure 2: Q9

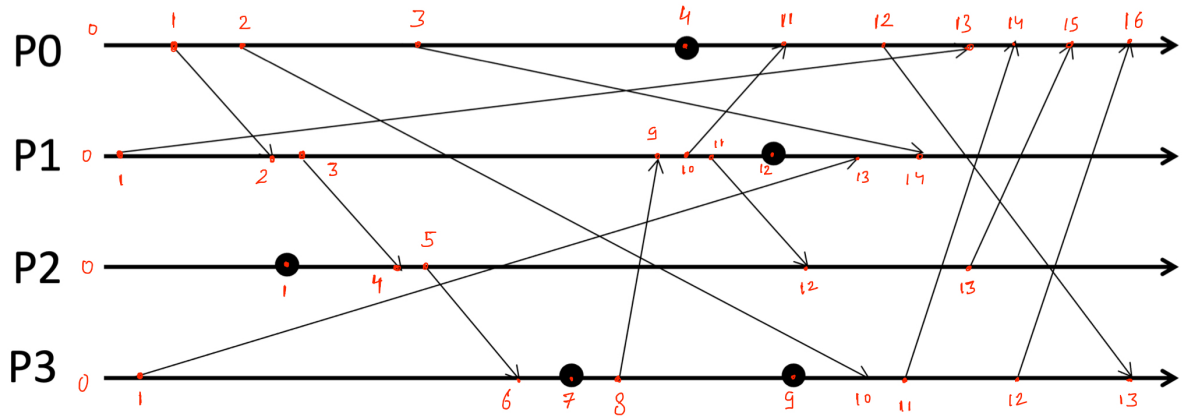

Figure 3: A9

Question 9: Spiderman and his doppelgangers in the metaverse are trying to figure out the communication among the different metaverses. They have the following 5 timeline of messages exchanged, where each dot represents an instruction. Can you mark Lamport timestamps on each event? It is ok to print out this and handdraw/write the timestamps (then scan or photograph your solution, and insert it into your solution doc).

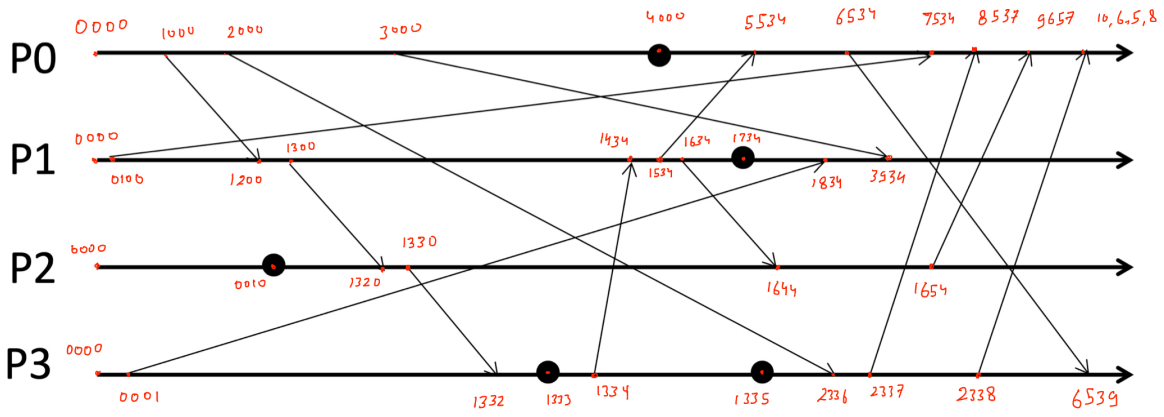

Figure 4: A10

Question 10: Whoops, the simultaneous and sudden arrival of all of the concurrent villains (Doc Ock, Green Goblin, Electro, Sandman, and the others) means Lamport timestamps are unusable for distinguishing concurrent events from causal events. Can you mark vector timestamps for the same timeline as in the previous question? It is ok to print out this and hand-draw/write the timestamps (then scan or photograph your solution, and insert it into your solution doc).

References