Illinois Institute of Technology

Advanced Operating System (CS-550)

HW-3

Submitted By:

Batkhishig Dulamsurankhor (#A20543498)
Nitin Singh (#A20516824)
Vaishnavi Papudesi Babu (#A20527963)

Instructor: Professor Ioan Raicu
TA: Mr. Lan Nguyen
TA: Ms. Sonal Gaikwad
TA: Mr. Adarsh Agrawal

CS550 Homework #3

Instructions:

- *Assigned date: Tuesday October 24th, 2023*
- *Due date: 11:59PM on Monday October 30th, 2023*
- *Maximum Points: 100%*
- *This homework can be done in groups up to 3 students*
- *Please post your questions to BB*
- *Only a softcopy submission is required; it will automatically be collected through GIT after the deadline*
- *Late submission will be penalized at 20% per day; an email to the TA with the subject "CS550: late homework submission" must be sent*

1. **(5 points)** Outline an efficient implementation of globally unique identifiers.
   For an efficient implementation of GUIDs, we need to consider a few factors:

   GUIDs generation and comparison should be efficient, fast, and scalable, especially when we deal with lots of requests. We should use algorithms and techniques that are optimized for rapid generation.

   We need to implement a strategy for GUIDs to make sure that they are unique across the system.

   We need to implement randomness or pseudo-randomness to avoid generating predictable or sequential identifiers that could compromise security or cause collisions.

   To generate a unique identifier, use a strong hashing method such as SHA-1 or SHA-256. These techniques are used to generate secure and reliable IDs by producing unique hashes for various inputs.

2. **(5 points)** Name at least three sources of delay that can be introduced between broadcasting the time over the network and the processors in a distributed system setting their internal clocks.
   Clock Drift:
   Over a period of time, the internal clocks of processors will have slight variations in the tick rates, as a result, they will accumulate and cause a time drift.
   Thus, it will create a difference between the clocks of different processors in the system.

   Network Latency:

The time it takes for a packet of data to travel from one point to the other over a network can cause a delay. As a result, there will be inconsistencies in timekeeping across different nodes. It can be caused due to multiple reasons, like distance between two points, the speed of the network, and amount of traffic in the network.

Clock Skew:
The difference in time between the sender and receiver clocks. It can be caused due to multiple reasons like, the age of the clock, the temperature, and the type of the clock.

Message processing time:
Time synchronization delays are also caused by the processing time it takes for the receiving nodes to receive the time synchronization message. The processing time can vary depending on the receiving node's load and compute power, which can lead to time synchronization differences.

3. **(5 points)** Consider the behavior of two machines in a distributed system. Both have clocks that are supposed to tick 1000 times per millisecond. One of them actually does, but the other ticks only 990 times per millisecond. If UTC updates come in once a minute, what is the maximum clock skew that will occur?

In this case, One machine's clock ticks as expected at 1000 ticks per millisecond(ms).
On the other hand the other machine will tick slightly slower and actually ticks at 990 ticks per millisecond(ms).
Thus, the difference in ticks per millisecond will be: 1000 - 990 i.e. **10 ticks/ms.**
In a minute this error will be 600 ms

Hence, the max clock skew will be 600ms.
We can also conclude that second clock is one percent slower.

4. **(5 points)** One of the modern devices that have (silently) crept into distributed systems are GPS receivers. Give examples of distributed applications that can make use of GPS information. Take one example and explain in detail how it uses the GPS information.

One example of distributed application that uses GPS information is a fleet management system (FMS). It is a software program that helps businesses manage their fleets of vehicles.
It usually has functions like vehicle monitoring, route planning, and fuel management.

GPS receivers are used by fleet management systems to track the location of vehicles in real time. This data is then sent to a centralized server, where it is processed and evaluated.
The fleet can use this information and provide many features:

1. **Optimized delivery route**:

The fleet management can use the location of the vehicle to optimize the delivery routes.
Thus, it can reduce the fuel cost and also reduce the delivery time. The application may recommend the fastest and most efficient routes by dynamically updating the route information based on real-time GPS data.

2. **Optimize fleet's operation**:

The fleet management system can also use GPS data to determine each vehicle's distance traveled, projected arrival time for deliveries, and fuel usage. This data can subsequently be utilized to develop reports and optimize the operations of the fleet.

3. **To create Geofences**:

A geofence is a map-based virtual barrier that can be drawn on the ground. When a vehicle enters or leaves a geofence, an alert is sent to the fleet management system. This alert can be used to monitor vehicle movement, confirm that the vehicle is on route, and alert the fleet management system if a vehicle passes or leaves a restricted area.

4. **Real time tracking**:

Vehicles' locations can be tracked in real time. This data is then presented on a map, allowing the fleet management to see where all of their vehicles are.

5. **(5 points)** Consider a communication layer in which messages are delivered only in the order that they were sent. Give an example in which even this ordering is unnecessarily restrictive.

   For example, let's take the context of transferring a large image that is divided into consecutive blocks, there is no use of strict FIFO(First In First Out) ordering for message delivery. The receiver can paste the incoming block in the correct position when it arrives. The positioning of the block depends on their respective identifiers, which represents its position in the original image as well as their height and width.

   Thus, the correct position of the block depends on their respective identifiers, so the order in which the block arrives is not so important.Therefore, the rigid FIFO order requirement can be relaxed to provide more flexibility and speed of message delivery.
   The communication layer can reduce delays, improve overall transmission speed and improve image rendering at the recipient's end by adopting a more flexible approach.

6. **(5 points)** Explain in your own words what the main reason is for considering weak consistency models. It is often argued that weak consistency models impose an extra burden for programmers. To what extent is this statement true?

   There is no one-size-fits-all answer to the question I believe, While some argue that weak consistency models add complexity, others argue that programmers are accustomed to protecting shared data through synchronization techniques like locks or transactions.
   Read-only and write-only technologies often require more coarse-grained concurrency than read-only parallelism.
   Despite this, programmers often expect synchronization variable operations to follow sequential consistency because it is essential for the predictability and consistency of concurrent operations.

7. **(10 points)** What kind of consistency would you use to implement an electronic stock market? Explain your answer.

   When constructing an electronic stock market, choosing the right consistency model is essential to ensure that all transactions and operations are handled correctly. In this situation, sequential consistency is the best.

   Sequential consistency is based on the idea that all activities happen in a certain order.

Therefore, stock market data must accurately reflect the chronological order of transactions and modifications.

Sequential consistency makes it possible for all participants in the electronic stock market to observe changes in stock prices and their subsequent movements on a consistent basis. Sequential consistency allows investors and traders to see a consistent and unified view of market activity.
By following the principle of sequential consistency, the electronic stock market is able to maintain transparency and confidence among investors by creating a stable and reliable trading environment.

Let’s take an example:
1. Trader “Robert” places a buy order for 100 shares of Tesla.
2. Trader “Peter” places a sell order for 100 shares of Tesla.
3. The stock exchange will match the two orders and execute the trade.
4. The stock exchange will update the account of Trader “Robert” and Trader “Peter”.
5. The stock exchange replicates the updated account to all its servers.

Sequential consistency requires all replicas of the stock exchange database to be updated sequentially.
Therefore, even if you have multiple traders on different servers, all of them will have the same view on the market.

8. **(10 points)** Consider a personal mailbox for a mobile user, implemented as part of a wide-area distributed database. What kind of client-centric consistency would be most appropriate?
Answer: For a personal mailbox designed for mobile users within a wide-area distributed database, the most suitable form of client-centric consistency is **Monotonic Read Consistency**. This model ensures that once a client retrieves a value, any subsequent read operation on that value will consistently provide the same or a more recent version. This is important for a personal mailbox as it guarantees users always have access to the latest version of their mailbox, irrespective of the server they connect to within the distributed database. When a user switches servers, the new server ensures that they receive at least all the updates from their previous server, maintaining the integrity of their mailbox, and allowing them to read incoming mail seamlessly while on the go.

9. **(10 points)** We have stated that totally ordered multicasting using Lamport's logical clocks does not scale. Explain why.
Answer: Totally ordered multicasting using Lamport's logical clocks does not scale primarily due to the absence of a global clock, which results in local clocks on different machines being potentially unsynchronized. This lack of global time reference makes it challenging to order events accurately across different machines, leading to the need for extensive all-to-all communication between participants. This inefficient communication pattern becomes increasingly impractical as the number of participants grows, resulting in significant overhead and potential network congestion. Furthermore, this approach is susceptible to message delays and losses, which can disrupt the total order of the multicast and cause inconsistent outcomes for participants. In summary, the scalability issues stem from the inherent challenges of maintaining synchronized logical clocks and the extensive communication required to achieve totally ordered multicasting in large and dynamic systems.

10. **(10 points)** For active replication to work in general, it is necessary that all operations be carried out in the same order at each replica. Is this ordering always necessary?
    Answer: No, all operations need not carry in same order at each replica in active replication. There are few scenarios where there is no need to follow an order.
    For example, if replicas are independent of each other, they can perform operations in any order without affecting systems consistency.
    Another scenario is the read-only or write-only operations, where each replica can operate independently without requiring a specific order. Hence, whether or not it is necessary to order all operations in the same order at each replica depends on the specific needs and usecase.

11. **(10 points)** Define what a hashing algorithm is. What is the difference between a hashing algorithm and a cryptographic hashing algorithm? Give the pseudocode of a simple hashing algorithm. What is the time complexity of your algorithm?
    Answer: A hashing algorithm is a function that converts any input data of arbitrary size into a fixed-length output known as a hash. Hashing algorithms are often used in computer science to index and retrieve data, identify duplicate data, and verify the integrity of data.

    Cryptographic algorithm is also a hashing algorithm that is designed to be resistant to collision, to ensure the integrity of data and to protect against tampering. A collision attack is a type of attack in which an attacker attempts to find two different data inputs that produce the same hash value.
    The main difference between hashing algorithm and a cryptographic hashing algorithm is that cryptographic hashing algorithms are designed to be one-way functions, meaning that it is computationally infeasible to determine the input message from the hash output whereas with a hash value generated using hashing algorithm, we can always find the original input data.

    Below is a simple hashing algorithm, it is adding the ascii character value of each character of the given string to form a numerical value.

```
Hash(string input):
  hash = 0
  for each c in input:
    hash = hash +ASCII( c )
  done
  return hash
```

    The time complexity of the above algorithm is O(n) where N is the size of input, as each character has to be processed, converted to ASCII and appended to hash value. .

12. **(10 points)** Explain in your own words what a blockchain is. What are the two main problems blockchains solve? Explain how these problems are solved in blockchain technologies.
    Answer: A block chain is a distributed, secure, and transparent digital ledger that records information in a way that it cannot be tampered. It is a decentralized system where data is

stored in blocks, each block is linked to next block and a block in the chain contains a cryptographic hash of the previous block, a timestamp, and transaction data. So the data tampering is very difficult. A consensus mechanism like proof-of-work or proof-of-stake is used to agree on the valid state of the blockchain among distributed nodes.

The two main problems that blockchain solves are security and transparency.
Security and trust has been a problem in traditional systems due to data centralization which is vulnerable to corruption, fraud, single point of failure. In the context of digital currency, double spending is one of the concerns where a unit of currency is spent twice. Blockchains address these problems by offering a decentralized and transparent system using cryptographic hashes and linking blocks in a chain. Each block contains a hash of the previous block. Trying to tamper with a previous block will cause the hashes to become invalid and break the chain. This makes the data practically unalterable once recorded. The network will reject any blocks that are tampered with. This provides transparency and trust in the data.
In traditional systems, data inequality, corruption and mistrust are major concerns due to lack of transparency. Blockchains provide a solution to the challenge of reaching an agreement among a group of participants who might not trust one another. In a blockchain, transactions go through a process of validation by various participants, often called nodes, using a consensus mechanism like Proof of Work or Proof of Stake. This mechanism guarantees that all participants reach a consensus on which transactions are valid and the order in which they are added to the blockchain. This shared transparency and consensus effectively thwart the ability of any single entity to manipulate the system for their own gain. Furthermore, in the context of digital currencies, this approach prevents the issue of double-spending. It ensures that all participants in the network have access to the same information, and the rules for adding new blocks are collectively and democratically agreed upon.

13. **(10 points)** There are 3 main consensus algorithms in blockchain technologies. Proof of Work, Proof of Space, and Proof of Stake. Define each one, and discuss 2 advantages and 2 disadvantages of each approach (compared to other approaches).

Proof of work is one of the popular consensus algorithms in public blockchain. PoW involves the nodes/miners solving complicated mathematical puzzles to produce new blocks. This process is called mining. It takes 10 minutes to solve one puzzle. The first miner to solve the puzzle will get a new block added to it. Although it is complex to solve the problem ,they are easy for the network to verify.
Advantages:
1. Robust way of maintaining a decentralized blockchain, any minor can participate with appropriate hardware
2. Provides strong security against fraudulent transactions and attacks unlike other consensus algorithms. Mining is difficult and requires high computing power.
Disadvantages:
1. Energy intensive, as mining requires large amount of energy to constantly compete to solve the mathematical problems required to mine new blocks.
2. Performance implications as miners need time to solve mathematical problems and add new blocks to blockchain.
Proof of Space is a consensus mechanism that uses hard disk space instead of computational

power (like PoW) to validate transactions. Each validator compete to prove that a certain amount of disk space is available, the more disk space allocated, the higher chance they have of being selected to create new blocks.
Advantages:
1. Energy efficient than PoW as they do not require solving puzzles like PoW
2. PoSpace blockchains can process transactions very quickly, as validators can simply provide more disk space to the network.
Disadvantages:
1. PoSpace may be more vulnerable compared to PoW and PoS for attacks such are timelords and plotsteal
2. PoSpace blockchains can be more centralized than PoW blockchains, as farmers with more disk space have more influence over the network.

Proof of Stake is an algorithm that randomly selects validators for block creation based on the amount that token holders stake from their crypto-asset ownership. The more cryptocurrency a validator stakes, the more likely they are to be selected to validate the next block.
Advantages:
1. It is extremely energy efficient and have greater transaction throughput as it does not require minors to solve problems or mining.
2. It is scalable as PoS does not rely on physical machines to generate consensus
Disadvantages:
1. Potential for centralization around wealthy validators
2. PoS blockchains are more vulnerable to certain types of attacks, such as stake grinding and slashing.

2 Where you will submit

You will have to submit your solution to a private git repository created for you at https://classroom.github.com/a/3GUBhnWx. You will have to first clone the repository. Then you will have to add or update your source code, documentation and report. Your solution will be collected automatically after the deadline. If you want to submit your homework later, you will have to push your final version to your GIT repository and you will have let the TA know of it through email. There is no need to submit anything on BB for this assignment. If you cannot access your repository, contact the TAs. You can find a git cheat sheet here: https://www.git-tower.com/blog/git-cheat-sheet/

3 What you will submit

When you have finished implementing the complete assignment as described above, you should submit your solution to your private git repository. Each program must work correctly and be detailed in-line documented. You should hand in:

1. **Report:** A written document (typed, named hw3-report.pdf) describing the overall assignment, clearly numbered with questions answered.

Submit report through GIT.

Grades for late programs will be lowered 20% per day late.