Raed A. Hemeed, #SID A20539655
CS 550
FALL23, WA3
Due: Oct 30, 2023

1. **Storage and Serialization**: Decide how to store and serialize UUIDs in your application.
   Performance Considerations: Optimize the generation of UUIDs for performance, especially in high-throughput systems.
   Testing and Verification: Regularly test the generated UUIDs to ensure their uniqueness and validity.
   Distribution and Scalability: Consider how UUIDs will be distributed and managed in a distributed or clustered system.
   Documentation and Compliance: Document the UUID generation process, especially if you are following a specific UUID version standard.

2. First, signal propagation delay in the atmosphere. Second, packet propagation delay on the LAN. Third, delay in each processor after the packet arrives, due to interrupt processing and internal queueing delays.

3. Let's break down the scenario and calculate the maximum clock skew:
   Clock A (Ticks 1000 times per millisecond)
   Clock B (Ticks 990 times per millisecond)
   Now, let's calculate how much time each clock will advance in each minute:
   Clock A: $1000\text{ticks/milliseconds} \times 1000\text{milliseconds/seconds} \times 60\text{seconds/minute}$
   $= 60{,}000{,}000\text{ticks/minute}$
   Clock B: $990\text{ticks/milliseconds} \times 1000\text{milliseconds/seconds} \times 60\text{seconds/minute}$
   $= 59{,}400{,}000$ ticks/minute

   The difference in tick count between Clock A and Clock B in a minute is
   $60{,}000{,}000 - 59{,}400{,}000 = 600{,}000$ ticks.

   However, this difference doesn't directly translate into seconds because the clocks are running at different rates. We can calculate the maximum clock skew by converting the tick difference into time on Clock A, which is running at the correct rate:

   Maximum Clock Skew =(Ticks per Millisecond/Tick Difference) seconds
   Maximum Clock Skew $=600{,}000/1000=600$ milliseconds
   So, the maximum clock skew that will occur between these two machines is 600 milliseconds. This means that Clock B will be up to 600 milliseconds behind Clock A due to their different tick rates.

4. IEEE 802.11 Networks for Path Loss Exponent (PLE) Estimation.
   Wirless Video Processing & Monitoring using mobile.
   Localization of Automobile Tacking System.

Tracking Pace while Exercising:

One typical example that comes to mind is sports and health care. There are now GPS-based body-area networks that allow a person to keep track of his pace while exercising an outdoors sport. These networks are often augmented with heart rate monitors and can be hooked up to a computer to download the sensored data for further analysis.

5. Suppose that a large image is being transferred and, for that purpose, it has been split into successive blocks. The position of each block in the initial image, as well as the width and height of each block, can be used to identify each incoming block. In such a case, there is no need to FIFO order the incoming blocks, as each incoming block can simply be inserted into the appropriate position.

6. Weak consistency models need to be replicated for performance. But, efficient replication only works if we can avoid global synchronization, which we can do by reducing consistency constraints.

7. Causal consistency is probably enough. The issue is that the effects of changes in stock levels should be uniform. Independent changes in stocks show up in different order.

8. Types of client centric consistencies:
   i. **Monotonic Read Consistency:** Monotonic read consistency is when a process reads a data item and then every time it reads that same data item again, it always returns the same value or a newer one.
   ii. **Monotonic Write Consistency:** In monotonic write consistency, a process does a write operation on a data item which is completed before any successive write operation on that data item by the same process.
   iii. **Read-your-writes consistency:** In read-your-writes consistency, a value written by a process on a data item will always be available to a successive read operation performed by the same process on that data item.
   iv. **Writes-follows-reads consistency:** writes-follows-reads consistency, a write operation by a process on a data item following a previous read operation on the same data item by the same process is guaranteed to take place on the same or a more recent value of that data item that was read.

   Basically, depending on the types of client-centric consistencies mentioned above, the best client-centric consistency for a mobile mailbox that's part of a WADD can be any of the four or all of them. The main goal is for the mobile user to always see the mailbox, no matter if they're updating or reading. So, you can just use a primary-based local-write protocol where the primary is always on the user's phone.

9. Lamport's logical clocks is the way of totally ordered multicasting which requires that all the servers are up and running. When one of the servers have turned out to be slow or

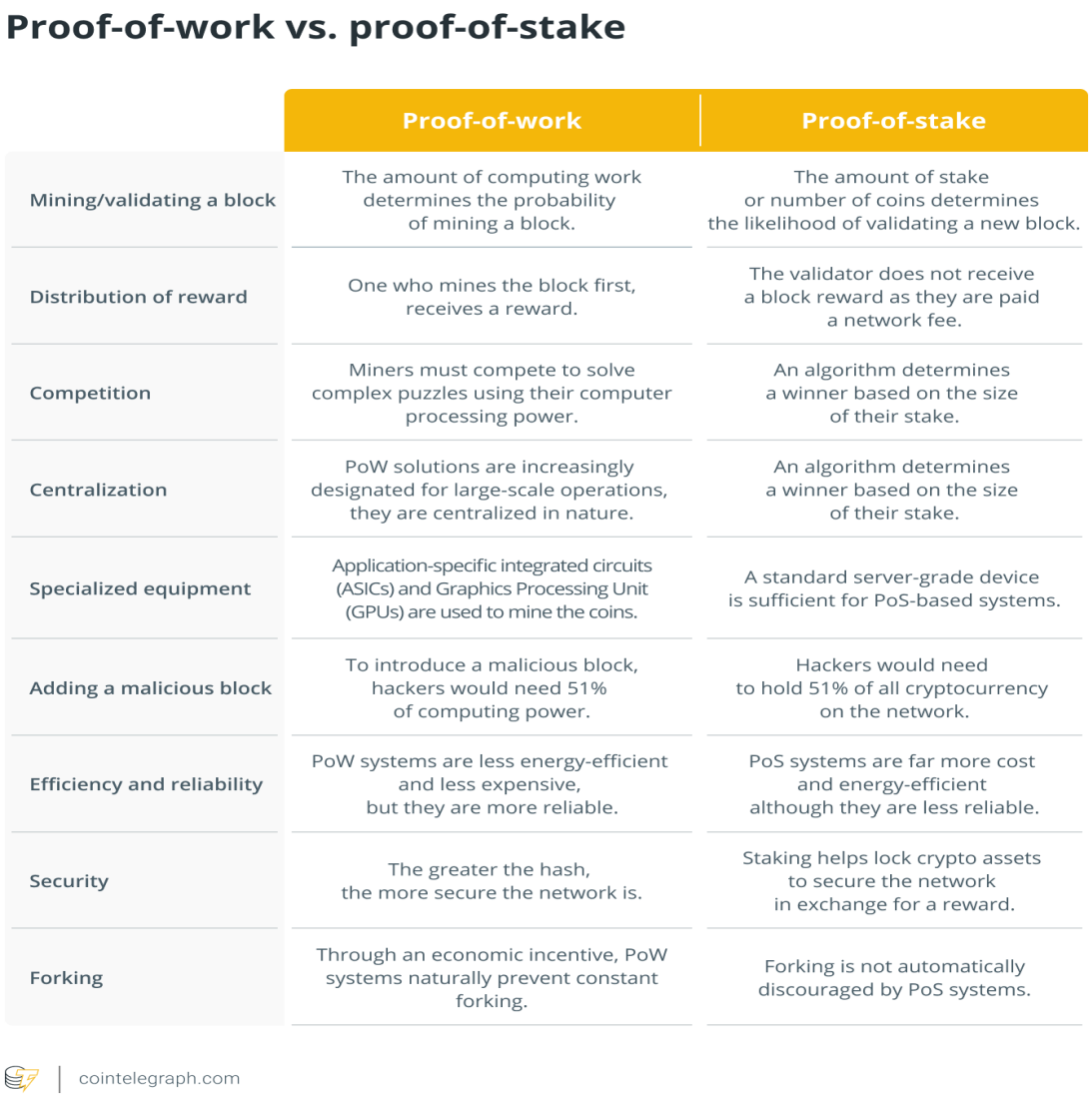
crashed then this effectively hinders the performance. Thus, this condition should and have to be detected by all the other servers. The growth in the number of the servers aggravated and gives a raise to the problem.

10. In general, for active replication to work correctly, it is necessary that all operations are carried out in the same order at each replica. This ensures that all replicas reach the same state, and the system remains consistent.
    However, there are certain scenarios where relaxed ordering, such as causal consistency or eventual consistency, may be acceptable depending on the specific requirements of the distributed system and the trade-offs between consistency and performance.

11. **Hashing Algorithm Definition**: A hashing algorithm is a mathematical function that maps an input to a fixed-size output, commonly used in computer science for data integrity and retrieval.
    Cryptographic Hashing Algorithm: A cryptographic hashing algorithm is designed for security and meets specific resistance criteria, making it suitable for secure applications.
    Simple pseudocode representation of a hashing algorithm:
    ```
    function simpleHash(input):
    initialize a variable hashValue to some initial value
    for each character in input:
      hashValue = (hashValue * 31 + character) % some_large_prime_number
       return hashValue
    ```
    The time complexity of the algorithm is O(n), linear in the length of the input string.

12. **Definition:** Blockchain is a method of recording information that makes it impossible or difficult for the system to be changed, hacked, or manipulated. A blockchain is a distributed, immutable, and decentralized ledger at its core that consists of a chain of blocks and each block contains a set of data. The blocks are linked together using cryptographic techniques and form a chronological chain of information.
    Problem solved by Blockchain:
    i. Supply Chain/Logistics
       Companies from large-scale industries will benefit from blockchain. Since they constantly purchase and sell goods to different parties, utilizing the technology will help them keep track of their operations.
       Being able to monitor orders, track invoices, and enter payment details seamlessly from start to end will ensure that all responsibilities are accomplished, and verifiable from the single source of truth that is the blockchain.
    ii. Security
       Blockchain technology can eliminate insider and cybersecurity threats within organizations. Since the entries in a ledger can only be accessed by authorized employees, this can greatly reduce the risk of a database leak. In case an unauthorized individual attempts to manipulate an entry, everyone in the network will be immediately notified of this change.

13. **Proof-of-work**: PoW is based on network users' capacity to prove that a computational task is accomplished. To answer a mathematical equation, some computing power known as a node is employed, and once the equation is solved, a new block on the chain is validated. A node is any physical device like a personal computer that can receive, send, or forward data within a network of other tools.
Proof-of-space: is a type of consensus algorithm achieved by demonstrating one's legitimate interest in a service (such as sending an email) by allocating a non-trivial amount of memory or disk space to solve a challenge presented by the service provider.
Proof-of-stack: To validate transactions on the crypto network, a user only needs to show that they own a particular quantity of cryptocurrency tokens that are native to the blockchain. This type of consensus mechanism used by blockchain networks to achieve distributed consensus is called the proof-of-stake consensus mechanism.



Proof-of-work vs. proof-of-stake

	Proof-of-work	Proof-of-stake
Mining/validating a block	The amount of computing work determines the probability of mining a block.	The amount of stake or number of coins determines the likelihood of validating a new block.
Distribution of reward	One who mines the block first, receives a reward.	The validator does not receive a block reward as they are paid a network fee.
Competition	Miners must compete to solve complex puzzles using their computer processing power.	An algorithm determines a winner based on the size of their stake.
Centralization	PoW solutions are increasingly designated for large-scale operations, they are centralized in nature.	An algorithm determines a winner based on the size of their stake.
Specialized equipment	Application-specific integrated circuits (ASICs) and Graphics Processing Unit (GPUs) are used to mine the coins.	A standard server-grade device is sufficient for PoS-based systems.
Adding a malicious block	To introduce a malicious block, hackers would need 51% of computing power.	Hackers would need to hold 51% of all cryptocurrency on the network.
Efficiency and reliability	PoW systems are less energy-efficient and less expensive, but they are more reliable.	PoS systems are far more cost and energy-efficient although they are less reliable.
Security	The greater the hash, the more secure the network is.	Staking helps lock crypto assets to secure the network in exchange for a reward.
Forking	Through an economic incentive, PoW systems naturally prevent constant forking.	Forking is not automatically discouraged by PoS systems.

cointelegraph.com

References:

Q12: https://coingeek.com/blockchain101/from-digital-to-real-life-7-problems-blockchain-can-solve/
Q13: https://cointelegraph.com/learn/proof-of-stake-vs-proof-of-work:-differences-explained