UNIVERSITY OF CENTRAL FLORIDA

SCALABILITY IN BLOCKCHAIN

Team 20

Pedro Roman

Dahlia La Pommeray

Sterling Downs

Weiyi Chen

Thalia La Pommeray

College of Engineering and Computer Science

2022

Abstract

We have determined a variety of things since beginning our work on this project. We had originally planned to run our blockchain off the Proof-of-Work consensus mechanism for processing transactions and creating new blocks. However, after further research, we realized that a Proof-of-Stake consensus mechanism would be more appropriate, which is explained further on. We had to figure out a way to parallelize our blockchain in a manner that would be effective. This is when we came to our second determination, which is the use of sharding, which will also be explained later.

Contents

0.1 Problem Definition

Blockchain transactions can be slow, sometimes taking hours to complete validation and verify integrity. Since a certain amount of work must be done to help prevent any abuse of the blockchain system and each block in the blockchain usually having a set limit for its transactions, scalability becomes an issue of blockchain. To help alleviate this issue, our idea is to create separate concurrent transaction chains that can handle transactions based on size, risk, and intention. For example, low risk transactions would be given priority to blocks that require less work to mine and therefore have a faster transaction rate.

0.2 How Blockchains Work

A blockchain is a type of database that stores the information together in groups, which are called blocks. Each block has a certain capacity of storage for the information, therefore when it is filled, it will be closed and added to the end of the blockchain. This data structure of compiling the data into chunks and stringing them together creates an irreversible timeline of data and makes it a great way of storing data about all kinds of transaction histories. The most widely known use is for storing monetary transactions.

The goal of blockchain is to allow the information to be recorded and distributed, but not edited. Therefore, other than the data a user wants to store, each block also contains its own hash, which identifies the block, as well as the hash of the previous block, and the hashes of sequential blocks can be compared to ensure the integrity of the blockchain. The hash is usually created by a mathematical algorithm which uses the information stored in the block and turns it into a string. This process is one-way and irreversible. If the information is modified, the hash of a block should be different as well, further invalidating the rest of the blockchain. The most common hashing algorithm is SHA-256 which is being used for Bitcoin's blockchain. However, with modern computers, hashes can be generated very fast and therefore, if a piece of information is being tampered with, the hash of the block which contains that information and the entire following blocks can be easily recalculated. To prevent that, blockchain uses Proof of

Work mechanism to ensure that each block can only be created after satisfying some conditions, which would cost a certain amount of computing power. The computation process of creating a block is also called mining. In Bitcoin's blockchain, it requires the hash of a block to begin with nineteen zeroes. And since the result of the hash function can not be influenced, computers will need to keep calculating until a hash that satisfies the condition comes up. This increases the difficulty of creating a block, as well as the cost of attempting to tamper with the data of an existing block.

Blockchain also ensures its security by allowing the data in the database to be spread out among different network nodes in multiple locations. In other words, it decentralizes the data so that it won't be easily tampered with. In the case of false information being stored, blockchain uses a fault-tolerant mechanism called consensus mechanism which requires the consensus of a majority of the network in order to modify the content of a block.

0.3 Challenges of Traditional Blockchains

Most blockchains depend on a proof-of-work consensus mechanism for validating transactions. This works by machines competing with each other to generate a hash usually beginning with a number of zeros. Each block gets processed one at a time and must be consistent with the rest of the blockchain. Such an approach works fine for a limited number of transactions. But as the transaction rate increases over time, the approach struggles to scale to meet demand. This issue with scalability is one of the biggest issues that modern blockchain implementations are facing. Currently, several popular cryptocurrencies such as Ethereum are in the process of researching a solution for this exact problem.

0.4 Benefits of Parallelizing Blockchain

Blockchain technology is a serial data structure, where each block must be processed one by one. This limits the speed of the technology because transactions occur mostly sequentially. Parallel processing is a very effective technique used to solve large-

scale problems, in the case of blockchain, this would mean unrelated transactions could occur concurrently which would yield more confirmations at a time. There are risks and costs associated with every technology, and with proof-of-work blockchains, we see a generally high computational power need. Ethereum, one of the largest cryptocurrencies in the world, uses around 113 terawatt-hours per year- the equivalent to about as much energy as the average US household uses in one week used for every single transaction. With the mechanism only being able to handle 15 transactions per second, it's also slow. Additionally, cryptocurrencies relying on a proof-of-work system are inherently difficult to scale.

Proof-of-stake systems are more performant due to its preference in capital over computing power, and being more compatible with parallel processing. This allows us to employ the help of a technique known as sharding. Sharding helps enable faster throughput by splitting a blockchain into several different instances that run in parallel. These shard chains increase the transactions per second rate since executions won't occur sequentially. With a parallel blockchain, one can expect the energy problem associated with proof-of-work to decrease substantially and transaction speed to soar. It lessens the environmental impact that proof-of-work has, but of course has its own risks and costs.

0.5 Importance of Security

TODO

0.6 Researching a Viable Solution

TODO

0.7 Consensus Mechanisms

Proof of Work

Proof-of-work is the original crypto consensus mechanism, which was first used by Bitcoin. The idea behind this consensus is that the blockchains are secured and verified by virtual miners around the world which are racing to be first to solve a "math puzzle." The solution to this puzzle requires a certain amount (typically being large) of computational power. The winner gets to update the blockchain with the latest transaction and is then rewarded by the network with a predetermined amount of crypto. This consensus mechanism has its advantages, such as the fact that it is a proven, robust way of maintaining a secure decentralized blockchain. As the value of the cryptocurrency increases, more miners are incentivized to join the network, which in turn increases the power and security of the blockchain. However, it also has its disadvantages. Because Proof-of-Work can become very energy-intensive considering such a large amount of transactions that can occur, it does not scale very well to accommodate that. Another flaw is a security issue. If a sub-blockchain within the network becomes bigger than the main blockchain, it will replace it. This is extremely unlikely to happen, especially for larger cryptocurrencies, but for smaller ones it can happen. This can pose a serious security issue.

Proof of Stake

The developers of the Proof-of-Stake consensus mechanism knew from the beginning that the Proof-of-Work mechanism would post limitations in scalability that would eventually need a solution to. This is why Etherium is investing so many resources into making the conversion for Etherium 2.0. In a Proof-of-Stake system, "staking" is different from the proof of work's "mining," in that it's the process by which a network participant gets selected to add a batch of transactions to the blockchain and earn some amount of crypto in exchange, instead of all machines competing for a single block in an arms race. Instead of having a math puzzle to solve, a "validator" can contribute, or stake, their own crypto in exchange for a chance of getting to validate a new transaction,

update the blockchain, and earn a reward. The whole process can be summarized in three steps:

1. The network selects a winner based on the amount of crypto that each validator has in the pool and the length of time it has been there. (rewarding the most invested participant)
2. Once the winner has validated the latest block of transactions, other validators can then come in and verify that the block is accurate, and once a threshold number of those attestations has been met, the network updates the blockchain.
3. All participating validators receive a reward in the native crypto, which is generally distributed in proportion to each validator's stake.

Not everyone can become a validator. It is a major responsibility and requires a fairly high level of technical knowledge. The minimum amount of crypto that is required to stake is typically relatively high, and users can lose their stake if their node goes offline or if they validate a "bad" block of transactions.

Proof-of-Stake can pose a disadvantage, however. A threat to this consensus mechanism is known as the 51% attack. When a validator controls 51% of a cryptocurrency, they can potentially have the ability to alter the blockchain. It is very expensive for someone to own 51% of the staked cryptocurrency, however, so this is not common at all. Additionally, if foul play were to be detected, they would lose their stake. While unlikely to happen, this is still important to note as a potential security issue.

0.8 Parallel Execution Model

For this project, we will be attempting to design our own parallel execution model for blockchain. Currently, most cryptocurrencies rely on a linear execution model which struggles with scalability and provides very limited throughput for transactions. While a linear execution model may be used for smaller cryptocurrencies, it is no longer a feasible solution for the larger and more active ones.

To support parallel execution, we will be making use of the Proof-of-Stake consensus mechanism. Due to its nature, as previously defined, it is much more compatible with

parallelism than the Proof-of-Work consensus mechanism. Additionally, we will be making use of a distributed network of nodes known as shards in a process called sharding. Such a design would allow our cryptocurrency to scale as needed.

0.9 Sharding

Sharding is a parallelization method that can help blockchains improve their efficiency and scalability–a big issue faced with blockchains. Sharding involves dividing an entire network into smaller partitions called shards. Each shard only needs to work on transactions relevant to their own. Different shards therefore can work in parallel to share the workload. The more shards there are, the better the performance.

0.10 Implementation

We will be implementing our solution with the Java programming language. While not as performant as a language such as C++, Java is better suited for the rapid prototyping of our proposed design. In our newly defined cryptocurrency, we will also be including several layers of security against more well-known blockchain attacks. However, security will not be our main focus for this project.

TODO Algorithms (FUTURE)

0.11 Performance Metrics

To ensure that progress is being made as the project develops, we will be keeping track of multiple versions of the project. Doing so will allow us to compare performance between implementations to ensure that performance does not diminish.

TODO Chart Comparison of Metrics (FUTURE)

[1]

Bibliography

[1] Mike Ross. “Review of Contracts”. In: vol. 2340. Aug. 2021, pp. 70–90.