Bitcoin project

Viola Renzi
renzi.1943828@studenti.uniroma1.it
Sapienza, University of Rome
Rome, Italy

Antonio Cordeiro
cordeiro.1999975@studenti.uniroma1.it
Sapienza, University of Rome
Rome, Italy

Abstract

Blockchain technology has garnered significant attention in recent years due to its potential to revolutionize various industries, including finance, supply chain, and healthcare. This paper explores a Java-based blockchain simulation, similar to Bitcoins, which provides insights into the fundamental concepts of blockchain networks, decentralized transactions, and the mining process. This paper discusses the key components of the simulation, including agents, miners, transactions and the blockchain structure. Additionally, it delves into the synchronization mechanisms used to ensure the integrity of the simulation in a multi-threaded environment.

Keywords: bitcoin, blockchain, block, peer-to-peer, multithread, transaction

1 Introduction

Bitcoin, introduced in 2008 by an anonymous entity known as Satoshi Nakamoto, revolutionized the financial landscape by pioneering the concept of decentralized digital currency. This groundbreaking cryptocurrency relies on blockchain technology to enable secure, peer-to-peer transactions without the need for intermediaries. Bitcoin's success is attributed to its robust security features, transparent ledger system, and fixed supply, making it a digital store of value and a medium of exchange. In essence, Bitcoin transactions are digital instructions that move bitcoins from one user's control to another. They are cryptographically secured to ensure the authenticity of the transfer and are recorded on the public ledger, the blockchain. Understanding the intricacies of these transactions is fundamental to comprehending how Bitcoin operates as a decentralized digital currency. The ledger is immutable, transactions are verified by a network of participants (miners) rather than a central authority. In this paper, we present a Java-based simulation that emulates the core features of a blockchain network, providing a hands-on experience of its mechanics.

2 Bitcoin Transactions

Bitcoin transactions are the core activity on the Bitcoin network, allowing users to send and receive bitcoins. Understanding how these transactions work is crucial to comprehending the inner workings of Bitcoin. Each transaction is uniquely identified by a transaction ID. Transaction IDs are used to track and reference transactions on the blockchain. Ownership of bitcoins is determined by controlling the private key associated with a Bitcoin address. When a user initiates a transaction, they sign it with their private key to prove ownership. This signature is critical for transaction validation. Bitcoin transactions are not considered valid until they are included in a block and added to the blockchain. Miners play a crucial role in validating transactions by verifying that the sender has the necessary bitcoins to spend and that the transaction has a valid digital signature. Transactions go through a confirmation process, where they are added to blocks in the blockchain. Each block represents a set of confirmed transactions.

3 Double spending

Double spending is a fundamental problem in the world of cryptocurrencies, most notably in the case of Bitcoin. This issue arises because digital currencies like Bitcoin exist solely in the digital realm and lack a physical form, making it difficult to prevent someone from spending the same Bitcoin more than once. Double spending occurs when a user attempts to send the same Bitcoin to multiple recipients simultaneously, exploiting the decentralized nature of the blockchain technology. To address this problem, Bitcoin relies on a consensus mechanism called proof-of-work, where miners compete to validate transactions by solving complex mathematical puzzles. Once a transaction is confirmed by a majority of miners, it becomes a part of the immutable blockchain ledger, making double spending practically impossible. However, the constant evolution of technology and potential vulnerabilities mean that the issue of double spending remains a key concern for the ongoing development and security of cryptocurrencies.

4 Agent Class

This class represents an individual agent or node within the blockchain network. Each agent has a unique public ID (which is made from a private ID), a private ID, and a balance of bitcoins. An agent is initialized with a constructor. The public key serves as an identifier, allowing others to send bitcoins to the wallet. The private key, on the other hand, is kept secret and is used to sign transactions, providing proof of ownership and authorizing the transfer of bitcoins. When a Bitcoin user initiates a transaction, they create a digital signature using their private key. The bitcoins will slowly increase because of the miner's rewards until they arrive at 21 million. Agents make transactions that have to be valid, so, for example, the sender's balance must be as big as the amount of money involved in the transaction. The agents' actions are mimicking the real world.

5 Miner Class

Miners, represented by the Miner class, are responsible for validating transactions, creating new blocks, and solving cryptographic puzzles (mining). To add a block to the blockchain, they must find a unique solution, known as a "nonce", which, when combined with the block's data, produces a hash value that meets specific criteria set by the network's difficulty level. This process, called "proof-of-work," is intentionally computationally intensive and energy-consuming, making it challenging and resource-intensive to find a valid solution. Miners are rewarded with bitcoins when the successfully mined block is inserted in the blockchain.

6 Transactions to Blockchain

A transaction represents the fundamental process through which the transfer of value occurs. It is a digital record of the movement of bitcoins from one party to another within the decentralized ledger. A transaction includes vital information such as the sender's Bitcoin address (public key), the recipient's address, the amount of bitcoins being sent, and a digital signature generated using the sender's private key. This digital signature serves as proof of ownership and authorization for the transaction. Once initiated, the transaction is broadcast to the Bitcoin network, where it enters a pool of pending transactions. Miners in the network select transactions from the pool, validate their authenticity, and group them into blocks. These blocks are then added to the blockchain, forming an immutable record of all transactions. A block is a fundamental component of the blockchain, serving as a container for a set of transactions. The structure ensures the integrity of the transaction history and prevents double spending.

7 Synchronization

We implemented miners and agents as threads. Thus, since we have a multi-threaded environment, we need synchronization to avoid race conditions and data corruption. The SynchroQueue class provides a synchronized queue for safely storing and accessing transactions. It ensures that transactions are processed in an orderly and synchronized manner by agents and miners, contributing to the reliability of the simulation.

8 Implementation Details

As already mentioned above, we decided to model agents and miners as threads. More specifically, the Agent class extends the Thread class, and miners are a subclass of agents. All agents have both a public and a private ID, and for simplicity they start with an initial balance of 50 bitcoins. After they start running, agents randomly create transactions and put them in a pool. In these transactions, they can either be the senders or the receivers (like asking someone for money in the real world), in which case they need to get the validation from the sender they are asking money from. The role of miners is to select transactions and place them into a block. They then begin the mining process to find a nonce that results in a block hash with a specified number of leading zeros (the difficulty level). While the real Bitcoin adjusts this difficulty periodically, in our project we have made it a user-settable constant. The Blockchain is implemented using a synchronized queue, and we have defined a method for adding blocks. This method checks the block's validity by recomputing its hash and verifying that the previous block's hash is correct. This is necessary because it's possible for two miners to mine a block simultaneously, but the one added to the chain first determines the correct previous block's hash. Finally, the World class is used to define all global variables and methods required to run the simulation. Users can configure several parameters, such as the number of agents and miners, the miner's reward amount, and the block mining difficulty level (i.e., the number of leading zeros in the block's hash). Additionally, there is a list to keep track of agents and a synchronized queue to manage the pool of transactions.

9 Conclusion

In conclusion, the project presented in this paper offers a practical and educational tool for understanding the concepts of blockchain technology. It provides insights into how transactions are validated, how miners compete to add blocks to the chain, how to work with double spending and how synchronization mechanisms ensure the integrity of the simulation. By exploring the code and its components, individuals can gain a deeper understanding of the decentralized, transparent, and secure nature of blockchain networks, which have the potential to transform various industries in the digital age.