


Article

Improving Transaction Speed and Scalability of Blockchain Systems via Parallel Proof of Work [†]

Shihab Shahriar Hazari * and Qusay H. Mahmoud 

Department of Electrical, Computer and Software Engineering, Ontario Tech University, Oshawa, ON L1G 0C5, Canada; qusay.mahmoud@uoit.ca

* Correspondence: shihab.hazari@ontariotechu.net

[†] This paper is an extended version of our paper published in the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019. This version includes additional ideas, algorithms, experimental results, as well as comparison tables and further discussion.

Received: 3 July 2020; Accepted: 24 July 2020; Published: 27 July 2020



Abstract: A blockchain is a distributed ledger forming a distributed consensus on a history of transactions, and is the underlying technology for the Bitcoin cryptocurrency. Its applications are far beyond the financial sector. The transaction verification process for cryptocurrencies is much slower than traditional digital transaction systems. One approach to scalability or the speed at which transactions are processed is to design a solution that offers faster Proof of Work. In this paper, we propose a method for accelerating the process of Proof of Work based on parallel mining rather than solo mining. The goal is to ensure that no more than two or more miners put the same effort into solving a specific block. The proposed method includes a process for selection of a manager, distribution of work and a reward system. This method has been implemented in a test environment that contains all the characteristics needed to perform Proof of Work for Bitcoin and has been tested, using a variety of case scenarios, by varying the difficulty level and number of validators. Experimental evaluations were performed locally and in a cloud environment, and experimental results demonstrate the feasibility the proposed method.

Keywords: blockchain; scalability; bitcoin; cryptocurrency; proof of work; nonce; transactions; bitcoin mining

1. Introduction

In conventional financial systems, a third party is constantly required to verify transactions. For example, if a person wants to buy a product from a market using a credit or debit card, the transaction is verified by a bank or other financial institution. If s/he wants to use cash for the purchase, s/he first needs to withdraw money from the bank, which means that the third party is always involved directly or indirectly for validating or verifying a transaction. In this sense, transactions are centralized through a third party. As a result, there is always a probability of a single point of failure. The objective of blockchain, which can be either permissionless or permissioned, is to build up a decentralized framework [1]. A cryptocurrency uses public or permissionless blockchain so that everyone can participate in performing the transactions. In contrast, permissioned blockchain networks allow the network to appoint a group of participants who are given authority to take part in a block validation process. This can be applied within a private organization or network. For transactions, this provides a disseminated record which contains the history of each affirmed transaction. It also offers a shared system where the clients themselves can check the exchanges of different clients without the incorporation of any outsider association. Moreover, this blockchain also keeps all the transactions

and user information anonymous and provides a copy of the continuous growing ledger to every user of the system.

However, as with all other systems, blockchain presents some concerns [2,3], one of which is the major issue of scalability—the rate at transactions are processed on the Bitcoin network. Hundreds of cryptocurrencies on the market currently use the blockchain network for transactions, mining and maintaining ledgers. All cryptocurrencies face scalability issues, but VISA, a traditional transaction provider, has already reached a peak of 10,547 transactions per second [4]. The transaction speed for different cryptocurrencies is different due to their respective protocols. Table 1 shows the transaction speed and confirmation time of different cryptocurrencies which is adapted from [5].

Table 1. Transaction speed of various cryptocurrencies.

Cryptocurrency	Transactions per Second	Average Transaction Confirmation Time
Bitcoin	3–7	60 min
Ethereum	15–25	6 min
Ripple	1500	4 s
Bitcoin Cash	61	60 min
Stellar	1000	2–5 s
Litecoin	56	30 min
Monero	4	30 min
IOTA	1500	2 min
Dash	10–28	15 min

Blockchain offers some unique components or features which differentiates it from traditional systems. To understand blockchain, it is necessary to understand those properties.

1.1. Mining and Miners

A cryptocurrency needs some sort of system to keep one decision party from manhandling it. A decentralized system has no expert to designate this assignment; hence blockchain set a protocol by which miners need to contribute some work to meet all the requirements for this task [6]. Any individual with the required computation power and processor can be a miner. Fundamentally, there are three obligations of a miner: to verify the transactions; to create a new block containing the transactions; and to immediately verify the block which has been created. To create a new block, miners have to find a hash, which is the result of a cryptographic calculation that interfaces the new block with its antecedent. Subsequent to finding a hash or a solution, a miner can create a block and add it to the blockchain. Other miners then verify the solution. As an impetus, the miner will receive a particular amount of cryptocurrency as a reward. However, the miner should have enough computational power to solve the hash within the time period.

1.2. Proof of Work

The algorithm that is used to confirm the transaction and add new blocks to the chain is called Proof of Work [7]. With Proof of Work, miners go up against each other to finish exchanges on the system and be compensated. A decentralized ledger accumulates every one of the exchanges into blocks. A block is added to the blockchain when any miner solves the hash for that block. The goal of Proof of Work is to find a possible solution for a complicated mathematical puzzle. The puzzle consists of many elements such as puzzle protocol and hash function. The complexity of the puzzle increases with the growth of the network.

For different types of cryptocurrency, different types of techniques are used as proof of work. For example, Bitcoin uses the SHA-256 cryptography technique [8]. Litecoin also follows a similar type of system, known as script [8] while Ethereum uses the Ethash algorithm [8]. Elements such as transaction time, complexity, and hash power differ in different cryptocurrencies due to dissimilar algorithms.

1.3. Decentralized System

A decentralized protocol empowers saving assets in a platform that can be found on the Internet. Through a decentralized protocol, the owners have absolute authority over their resources and have the right to exchange assets with anyone at any time [9]. The innovative nature of blockchain has found a way to form a decentralized system in the web. This system will allow owners to process their property any time they want without the participation of a third party. Individuals can specifically enjoy the exchange for a minimum charge. Moreover, a decentralized system such as this has no single point of failure, unlike a centralized system.

To this end, in this paper we present a method for improving the transaction speed and scalability of blockchain systems by extending our previous work in [10] and results from [11]. In the proposed method, all miners will use the same transaction data except for the nonce for a certain block, thus ensuring that no multiple miners perform the same work, accomplished through a manager. This model differs from traditional Proof of Work or the Bitcoin pool mining [12] in several aspects, such as the responsibilities of the manager, contribution of active miners, and the reward system.

The remainder of this paper is organized as follows: discussion of related work is presented in Section 2, while Section 3 presents details of the proposed method, along with an explanation of its features. Section 4 discusses the implementation and evaluation results of the proposed system. Challenges and solutions are discussed in in Section 5. Finally, conclusions and ideas for future work are presented in Section 6.

2. Related Work

The process of coordinator selection is extremely useful for improving the performance of a distributed system. In this approach, which was first implemented by Gerard Lelann [13], a consensus protocol is proposed with a coordinator election for a partially synchronous processor [14]. The coordinator divides and distributes the portion of work to peers in a network, where the final decision is taken by using a consensus protocol.

A similar type of work for leader election in the Bitcoin platform was conducted in Bitcoin-NG [15]. This accomplishes an execution change by decoupling Bitcoin's blockchain task into two planes: leader selection and exchange serialization. It also partitions time into the period, where every period has a solitary leader.

In Bitcoin-NG, there are two types of blocks: the key block and the microblock. The key block contains the leader information as well as information about the previous block. The microblock contains the transaction information. Thus, to generate the key block, a proof of work needs to be performed. Once elected, a leader is able to issue microblocks using his/her private key which contains the transaction information. The amount of microblock issued to the leader is dependent on signing speed and delay network propagation. The microblocks have no proof of work; therefore do not affect the chain weight.

A framework for parallel mining has been proposed by Boyen, Carr, and Haines [16]. Here, each transaction is connected to at least two other verified transactions and miners verify all new transactions in parallel. The network is graph-structured rather than linear structured, similar to Bitcoin and to the Tangle network used by IOTA [17].

Proof of stake, which is an alternative to proof of work, is used to create a new block in the blockchain network [18]. In proof of stake, a validator is chosen for each block based on the amount and duration of the stake. The validator is responsible for validating a block in this system, in which

there is no need to solve a puzzle. Moreover, the validator cannot mine any currency, but instead receives only a transaction fee as a reward.

In the Bitcoin pool framework (mining pools), many miners work together in parallel within a pool and use their hash energy to identify a solution for the block. The result is that a considerable amount of hash power is used to solve the mathematical puzzle which is found by a combination of all the miners' computational energy within the pool. This platform increases the possibility of solving the hash problem. If a block is solved, the block reward is distributed to all the miners who contributed to creating that block. Block awards are provided to the miners depending on their effort to create the block. Several methods, such as Shared Maximum Pay Per Share (SMPPS), Capped Pay Per Share with Recent Backpay (CPPSRB), and Equalized Shared Maximum Pay Per Share (ESMPPS), exist for distributing rewards [19]. However, the process of solving the proof of work, mining and rewards are different in mining pools.

To summarize, Bitcoin miners work separately to process transactions and create the next block in the network. As a result, for every block, the efforts of all miners except the successful miner, become useless and hence the need for massive amounts of energy. The existing mechanism where miners and/or validators cooperate with each other (Practical Byzantine Fault Tolerance or Pool Mining) to create a block brings centralization to the network. The proposed parallel Proof of Work motivates the miners to solve the puzzle by distributing the amount of work. Also, it maintains the decentralization and anonymity in the network, and along the way reduces the amount of energy required.

3. Method of Proposed Solution

To perform the proof of work, some of the data used by the miners are identical, including the Bitcoin index, the hash value of the previous block, and the timestamp. However, the content of transactions and the nonce value chosen by the miners may differ. The proposed method is designed in such a way that all miners will use the same transaction data but a different nonce. This means that all miners will use the same data except for the nonce for a certain block, thus ensuring that no multiple miners perform the same work.

To provide such an environment, a manager is required to ensure that no two miners use the same nonce value and that all miners use the same transaction data. The manager, who will be chosen from the miners, will be different in every epoch. Here, an epoch contains the time interval between two blocks. In this case, the manager rather than the miner will choose the nonce to compute. In this way, the manager can ensure that no two miners use the same nonce value. The manager is also responsible for creating the transaction hash for a certain block for which s/he is responsible, and which will be provided, along with the nonce value, to the miners. Again, unlike nonces, the transaction hash should be the same for all miners. In a traditional system, all nodes are connected to each other directly or via another node. In the proposed system, they will still be connected to each other and will also be directly connected to the manager.

There should be a genesis block at the start of the blockchain with no transactions. While a miner is randomly chosen as the manager for the next block (Block 1), for the remainder of the blocks, the manager selected will be the one who solved the block before the previous block. All the miners will now compete with each other to solve the genesis block, following the traditional method. When the genesis block is solved by a miner, the epoch for the next block will begin. The proposed solution will be effective at this point.

3.1. Distribution of Data

At the outset, as depicted in Figure 1, the manager will create a transaction hash with the unconfirmed transactions and, at the same time, will generate several groups of nonces. Each group will contain a range of nonce values; no same nonce value should be in multiple groups. If m numbers of miners are active in the network, the manager must initially generate and register at least m number of groups. The manager will then distribute the transaction hash and groups of nonces to each active

miner. The system will ensure that no two miners have the same group. With the exception of the manager, all miners will now try to find a solution for the next block with the available transaction data and the range of nonces allocated to each of them.

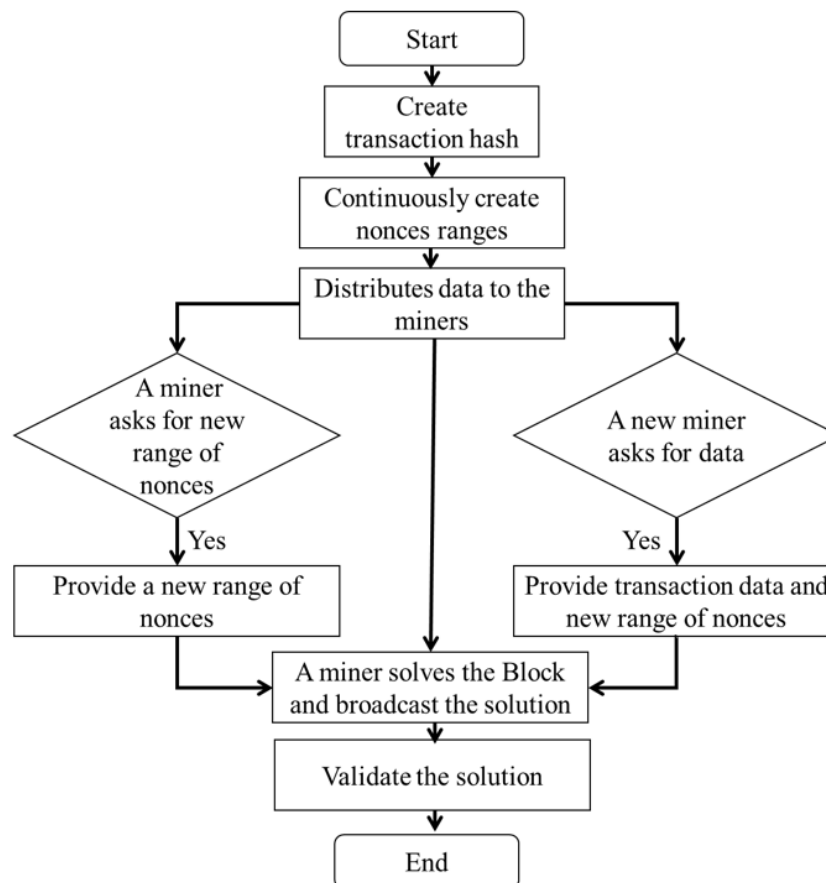


Figure 1. Workflow of a miner as a manager.

At the same time, the manager will generate and register more groups of nonces. Once a miner has used all of the nonce values of the allocated range, the miner will ask the manager for a new nonce range. The manager will then provide an unused range to that miner. Again, if a new miner enters into the network and asks the manager for required data, the manager will provide him/her with the same transaction data and a new group of nonces. For this reason, the manager should generate as many groups of nonces as possible. The process will continue until a designated solution for the current nonce is found.

3.2. Selection of a New Manager

In the proposed method, there will be a change of manager for each block. The validity of a manager will only remain for a certain block for which s/he is responsible. Only a miner who solves a block can be a manager. Upon solving a block, a miner will be the manager for the subsequent block. The genesis block has no manager as it contains no transactions while the manager of block 1 will be randomly chosen. For the remainder of the blocks, the manager selected will be the one who solved the block before the previous block. Therefore, having solved block number n , a miner will be the manager of $(n+2)$ block. In Figure 2, M5 has solved the genesis block, hence will be the manager for the 2nd block. After solving the genesis block, M5 will still act as a regular miner for first Block. When the first block is solved, M5 will act as manager for the second block and cannot compete with other miners as would a regular miner.

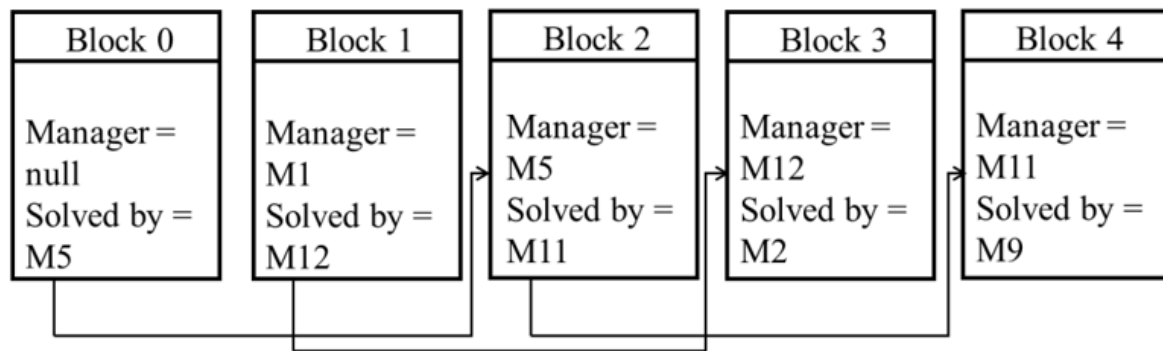
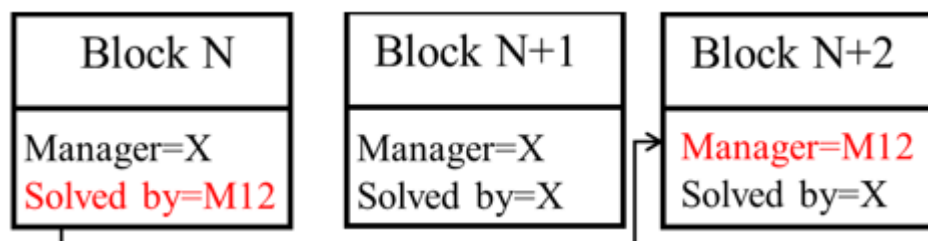


Figure 2. The process of manager selection.

3.3. Reward System

As a reward, a miner receives a transaction fee for all the transactions for the block s/he created. The miner can also mine a certain amount of cryptocurrency which, at present (2018) in Bitcoin is 12.5 BTC for each block. In the proposed method, having created a block, the miner will be able to mine a certain amount of cryptocurrency. However, the miner will not receive all the transaction fees for all the transactions. Instead, the fees will be split with the manager, who will receive 65% of the transaction fee while the remaining 35% will be awarded to the miner who solved the block. An example is provided in Figure 2, where Block 3 is solved by M2 with M12 as the manager of that block. M2 and M12 will receive 35% and 65%, respectively, of the transaction fee (Figure 3). The transaction data is created by the manager, who therefore receives greater reward than the miner who solved the block. A miner will receive both the reward (mining and transaction fees) on completing his/her responsibility as manager.



Reward for M12 =
 35% of the transaction fees of Block N
 + Mining coin
 + 65% of the transaction fees of Block N+2

Figure 3. Reward system.

The key features of the proposed solutions are: transaction speed, fairness to miners, and decentralization. Details are provided in the following subsections.

3.3.1. Transaction Speed

The goal of parallel mining is to increase the scalability of the system. Through parallel mining, the miner can more quickly reach consensus and so the transaction will be verified sooner. This will be beneficial for the general user who makes the transactions. According to evaluation test results, when compared to solo mining, this method registered a significant improvement.

3.3.2. Fairness to the Miners

In this system, every miner has an equal opportunity to be a manager. Furthermore, the reward system is considered in such a way that every contributor to a block (the manager and the miner who solved the hash) can obtain a portion of the reward. In terms of processing power, the miner who invests more in increasing the processing speed will have a higher probability of becoming a manager. Although everyone will work in parallel, the miner with more processing power will have the ability to calculate more nonce value, thus increasing the probability of becoming a manager. Algorithms 1 and 2 show the block solving and block validation techniques, respectively.

Algorithm 1. Block solving technique

1. Initialization

Asks for nonce range and transaction hash to the manager.

Receives transaction hash T from the manager.

Receives nonce range N from the manager.

2. Create record

$Record = Sha256(\text{Block index} + \text{Previous block hash} + \text{timestamp} + T \dots \dots)$

3. Solve puzzle

for $i = \text{initial nonce value to } N$ **do**

if $\text{length}(\text{Blockchain}) > \text{new block.index}$ **then**

 Block already solved

 Validate the Block solution

Break

$Solution = SHA256(Record+i)$

if $Solution$ satisfies the target **then**

 Solution found

 Broadcast the solution

Break

end if

end for

if solution is not found **or** Block not already solved **then**

 Asks for new nonce to the manager

 Receives nonce range N from the manager

Repeat step 3

end if

Algorithm 2. Block validation technique

if $\text{Previous Block Index}+1 \neq \text{New Block Index}$

return false

else if $\text{Previous Block Hash} \neq \text{New Block Previous Hash}$

return false

else if $\text{Hash}(\text{New Block}) > \text{target}$

return false

else

return true

end if

3.3.3. Decentralization

Both the current system and the proposed technique increase the probability of a miner with more processing power solving the puzzle. In the current system, it is theoretically possible for the miner with the highest computational power to solve all the blocks in the network. However, this is not allowed in the proposed system. Upon solving a block, in order to receive a reward, a miner has to act as a manager for the subsequent block. This allows for more decentralization in the system.

Bitcoin pool mining is a process where many miners work together to solve a block combining their hash resources. Parallel mining also encourages them to combine their mining resources. Though there

are many differences between these two processes, the major differences between these two processes are discussed in Table 2.

Table 2. A comparison between pool mining and parallel mining.

Attribute	Pool Mining	Parallel Mining
<i>Centralization</i>	In pool mining there is a fixed central coordinator who is responsible to provide mining resources to the miner.	There is no fixed central authority in parallel mining. The manager changes in every Block which keeps the system decentralized.
<i>Difficulty target</i>	Traditionally, the difficulty target assigned in a pool mining is less than the actual target in the Blockchain main stream.	The target in parallel mining is same as the target in Blockchain main stream.
<i>Rewards</i>	The rewards split to all participant based on the contribution of the miners.	The reward does not split. Only the miner who solved the Block gets all mining rewards. Transaction fees splits between the manager and the successful miner only.
<i>Responsibility of coordinator/manager</i>	The coordinator responsibility involves the assignment distribution to the miners, split of rewards, checking the contribution of each participant.	The manager responsibility includes distribute of transaction hash and nonces ranges.
<i>Pool fee</i>	Pool mining coordinator may take a small amount of reward from each participant. Also, there may be a participation fee for the miners.	There is neither reward fee nor participation fee for the miners.
<i>Contribution to the network</i>	The contribution to the network for each individual miner is assigned based on individuals mining resources.	The contribution is independent based on the mining resources of the peer.

The proposed method is developed for public blockchain where transaction can be done with necessary rewards. Different types of community can influence the network by providing service and accepting rewards. Table 3 compares the impact of different community in the network.

Table 3. Influence of different communities in the network.

Community	Service to the Network	Rewards Achieved	Influence on the Network
<i>Miners</i>	<ul style="list-style-type: none"> ○ Verify transactions. ○ Create the Block by performing PoW. ○ Validate the Block. 	<ul style="list-style-type: none"> ○ Get mining reward. ○ Get transaction fee. 	<ul style="list-style-type: none"> ○ The scalability of the network depends on the number of active miners. ○ It also depends on the processing machine used by the miners.
<i>Individual users</i>	<ul style="list-style-type: none"> ○ Provide transactions. ○ Pay transaction fees. 	<ul style="list-style-type: none"> ○ Transaction are verified and completed. ○ Get secured transaction environment. 	<ul style="list-style-type: none"> ○ They are the key community of the network. They can stop transaction and make the system worthless.
<i>Trading platforms or exchanges</i>	<ul style="list-style-type: none"> ○ Provide liquidity to the market. ○ Provide a fiat denominated value to the cryptocurrency. 	<ul style="list-style-type: none"> ○ Make profit from trading. ○ Hold the cryptocurrency. 	<ul style="list-style-type: none"> ○ Control the supply and price of cryptocurrency to the market.
<i>Businesses</i>	<ul style="list-style-type: none"> ○ Encourage the users to use the network by providing product and services. ○ Pay transaction fees. 	<ul style="list-style-type: none"> ○ Achieve secured business model. 	<ul style="list-style-type: none"> ○ Move the users to another transaction platform. ○ Offer developers to upgrade the network based on limitations.
<i>Developers</i>	<ul style="list-style-type: none"> ○ Upgrade the network. ○ Propose new features. 	<ul style="list-style-type: none"> ○ Get paid by developing the network. 	<ul style="list-style-type: none"> ○ Improvements are implemented by developers.

4. Results of Experimental Evaluation

The proposed method has been developed using the Go programming language, and the code is available from [20]. Specifically, a peer-to-peer network has been developed by using the GX library of Golang [21]. This is a decentralized package manager that is used to distribute the same program to different nodes. In order to perform the Proof of Work, a SHA-256 cryptographic hash algorithm has been used. The genesis block, which has no transaction record and no previous hash value, has been core coded. The miner who first connects to the system will be the manager for the next block as default.

A ring-structure peer-to-peer network [22] has been developed to implement the proposed solution. Each node can connect to maximum of two nodes. When a node is connected to a network, it can open a new connection by which a new node can connect to the network. Each node address contains of unique IP and id. The id is random and different for each node. The IP is the network IP of the node through which a new node can connect. When a node establishes a connection with a new node, it cannot accept new connections. Figures 4 and 5 represent diagrams of the network with different IP and unique id. Here, the green highlighted peer is the first peer of the network. The blue highlighted peer is waiting for an incoming connection as it is the last peer which is connected to

the network. In the following figure it is shown that how a direct connection is established with all other peers when a miner acts as a manager. Here, the peer with id 10005 is acting as the manager. The evaluation of the proposed system has been done both locally and in cloud.

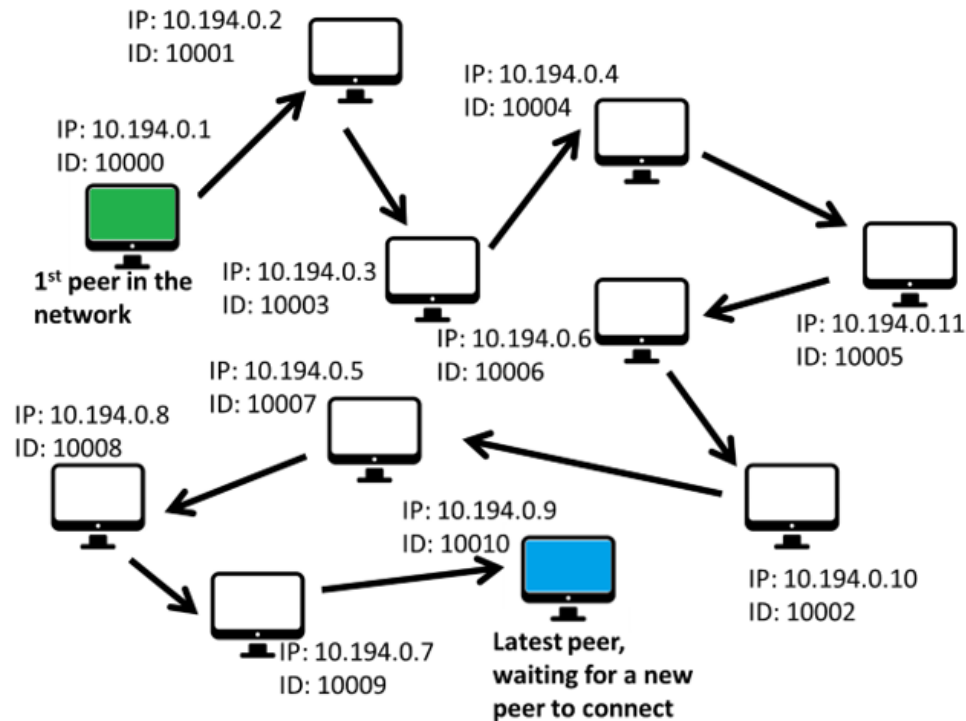


Figure 4. Peer-to-peer network diagram to implement the solution.

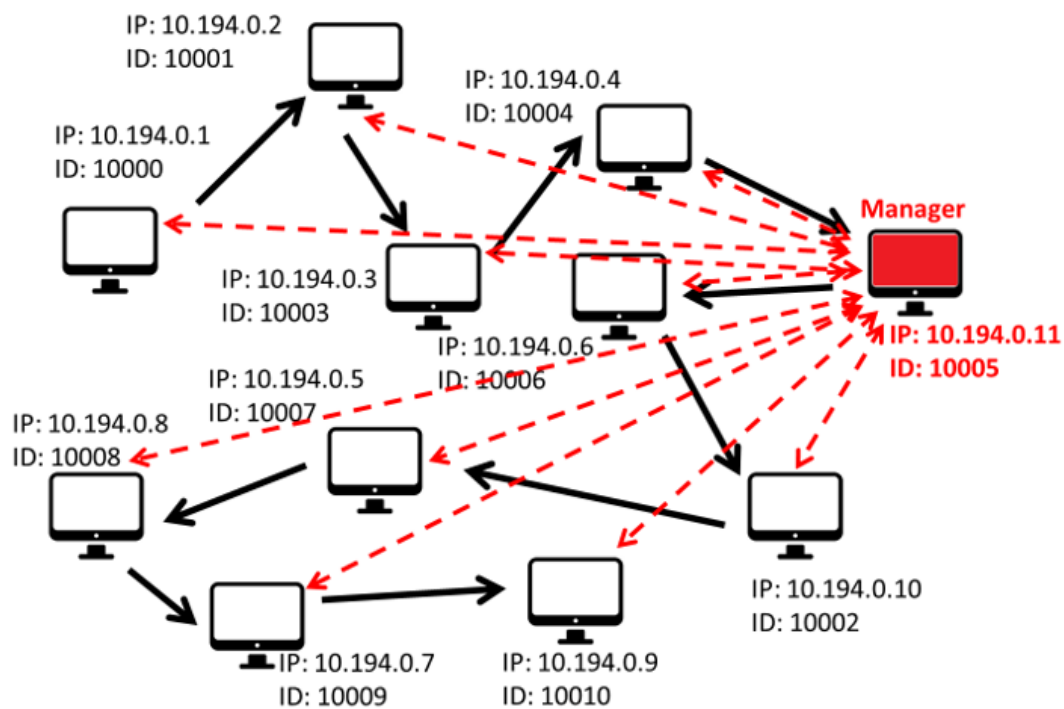


Figure 5. Peer-to-peer network diagram with a manager.

4.1. Evaluation in Local Environment

It is very important to distribute the resources equally to all peers to evaluate the solution. To implement this approach, we used a Docker container (docker.com). Docker provides a Linux-based container with its own network interface. A dedicated network has been created in Docker where all peers will be connected. The implementation has been performed in an Ubuntu operating system with Core i5-5200U CPU 2.2 GHz. The installed RAM is 4.00 GB. To ensure each miner has equal processing power, every miner has been allocated with 10% of the total resource. To compare the test result with the existing system, another similar environment has been developed using the same resources and components. In this system, the miners work solo. They compete with each other, as in the existing system, and a successful miner receives all the reward.

4.1.1. Experimental Setup

The test has been conducted based on different numbers of peers, both in solo and parallel mining, using different difficulty levels. Here, the difficulty level denotes the least number of consecutive zeros required at the beginning of an acceptable hash. Figures 6 and 7 represent the test result based on solo and parallel mining. Here, the average time(s) refers to the average time required to solve a block in seconds. This is calculated after conducting several tests under the same conditions and taking the average of all results. To identify the solution, the index, timestamp, transaction hash, previous hash and nonce are taken as input. Here, for the solo mining index, the timestamp and previous hash are the same for a certain block for all miners. In parallel mining along with these data, the transaction hash is also the same for all miners for a certain block.

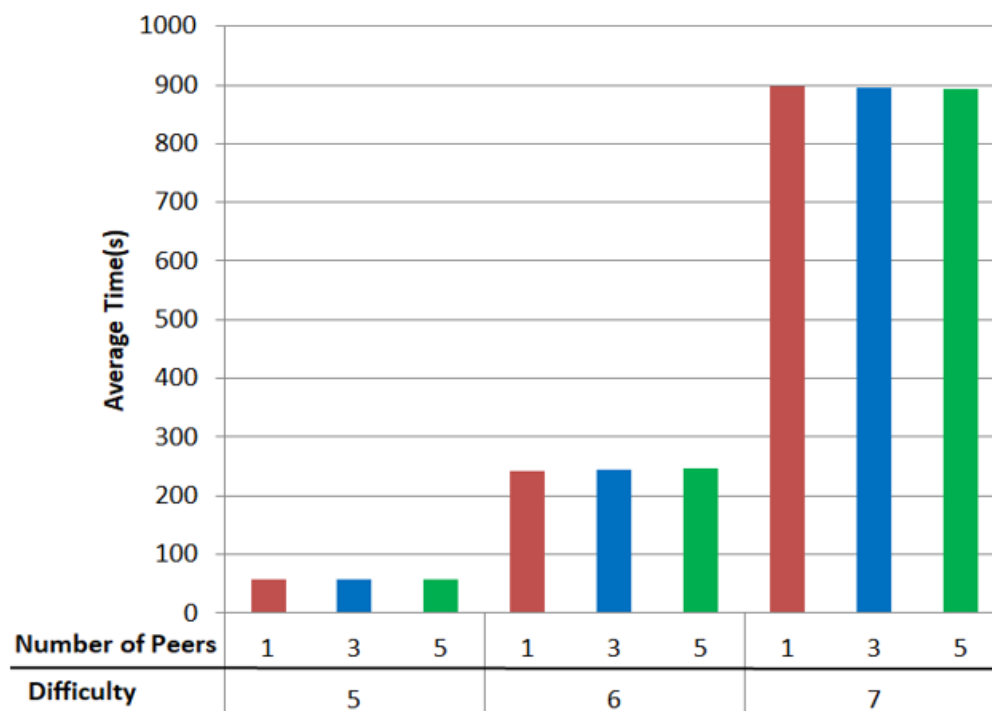


Figure 6. Test results for solo mining.

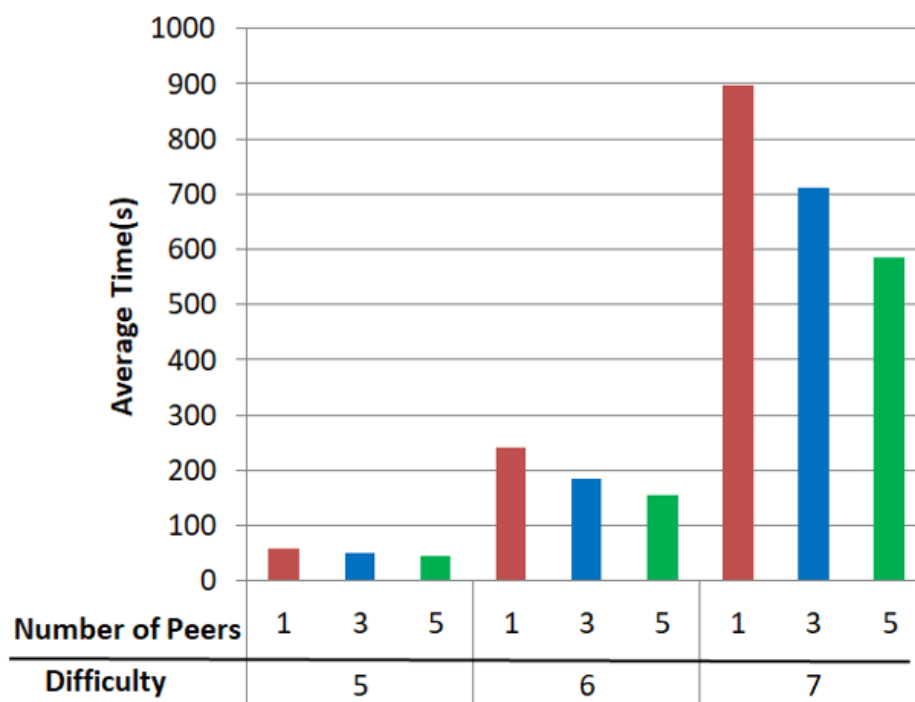


Figure 7. Test results for parallel mining.

4.1.2. Results

For difficulty levels 1, 2, 3 and 4, there is no significant difference between solo mining and parallel mining. However, for difficulty levels 5, 6 and 7, there is improvement in parallel mining, and, as Figures 6 and 7 depict, this improvement becomes significant with the increase in the difficulty level and the number of miners. In solo mining, the average time depends only on the level of difficulty, but, in parallel mining, the average time depends on both the difficulty level and the number of peers. If the level of difficulty increases, the average time required increases. Again, if the number of peers increases, the average time decreases because the miners are working in parallel and no two miners perform the same work. Another important aspect to notice is that the average time taken for one peer in parallel mining is almost the same as that in solo mining regardless of the number of peers. This is because, when there is only one miner in parallel mining, no parallel work is taking place. The improvement reaches 34% for five miners compared to one miner. It should be noted that the results may vary based on the processing power allocated to the miners.

4.2. Evaluation in Cloud Environment

The resources that can be provided in a local environment are very limited. As a result, both difficulty level and peer number cannot be increased. To address this limitation, the solution was implemented in a cloud platform. Google cloud platform (GCP) was used to implement the solution. GCP is a collection of cloud computing services which is provided by Google that runs on the same infrastructure that Google uses internally for its end-user products, such as Google Search and YouTube. Thirty-two virtual machines with equal resources were set up to do the experiment. Each machine contains 6.25 GB of memory with 1 virtual CPU. The operating system of every machine was Ubuntu 16.04 LTS with 10GB of allocated hard disc. The CPU platform was equivalent to Intel Skylake. No GPU was provided to any of the machines. The physical location of each virtual machine was in different region with different zone. Thus the network IPs are also from different groups.

4.2.1. Experimental Setup

To perform the experiment, different types of difficulty level were chosen. The targets for different difficulty levels are 0x1dffffff, 0x1d0fffff, 0x1d00ffff, 0x1c0fffff and 0x1bffffff. Each target has 6,7,8,9 and 10 leading 0's in the target respectively. We will represent the target as 6D, 7D, 8D, 9D and 10D respectively for the rest of the paper. The test has also been done for different numbers of peer in parallel mining. The numbers of peers were 2, 8, 14, 20, 26 and 32. In every experiment for parallel mining, there was always one miner in every epoch who acted as the manager. To compare the result in solo mining, a similar environment was created where every peer had the same resource configuration as parallel mining. In solo mining, the same difficulty level was used.

4.2.2. Results

For parallel mining, the test was done with different difficulty levels for different peers. The test was conducted for a large number of blocks continuously. Figures 8 and 9 represent the time required to solve any 15 consecutive blocks by a different number of peers in 6D and 10D difficulty levels, respectively. For 6D difficulty, the block solving time differences were not that significant compared to different number of peers. For example, the highest time required to solve one block was around 13.5 min when the test has been done for one peer. Again, the highest time required to solve one block was around 14.25 min for 19 peers. If the lowest block-solving time is considered, the time required to solve one block was almost same for one peer, seven peers and 19 peers. However, when the average time is considered for different numbers of peers, a small but significant result has been found. It seems that when the number of peer increase the average block solving time decrease. The average block solving time for one peer, seven peers, 13 peers, 19 peers, 25 peers and 31 peers were 7.92, 8.41, 8.79, 9.39 and 9.72 min respectively. With increase of 6 peers, the block solving time decreased to 0.45 min on average.

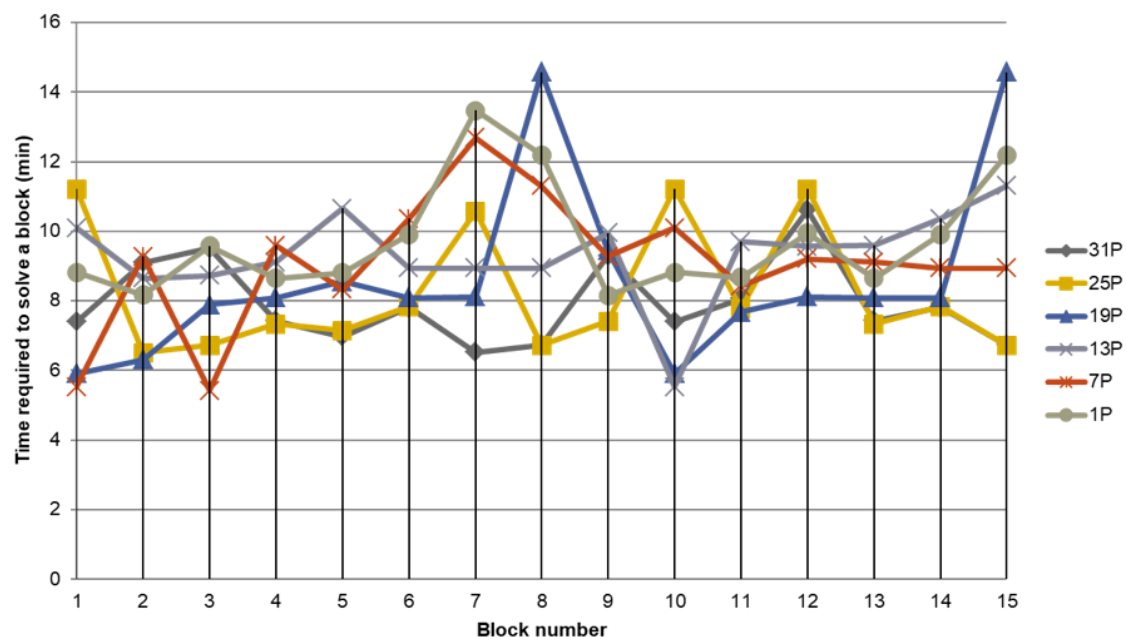


Figure 8. Time to solve 15 consecutive blocks by varying number of peers in 6D difficulty.

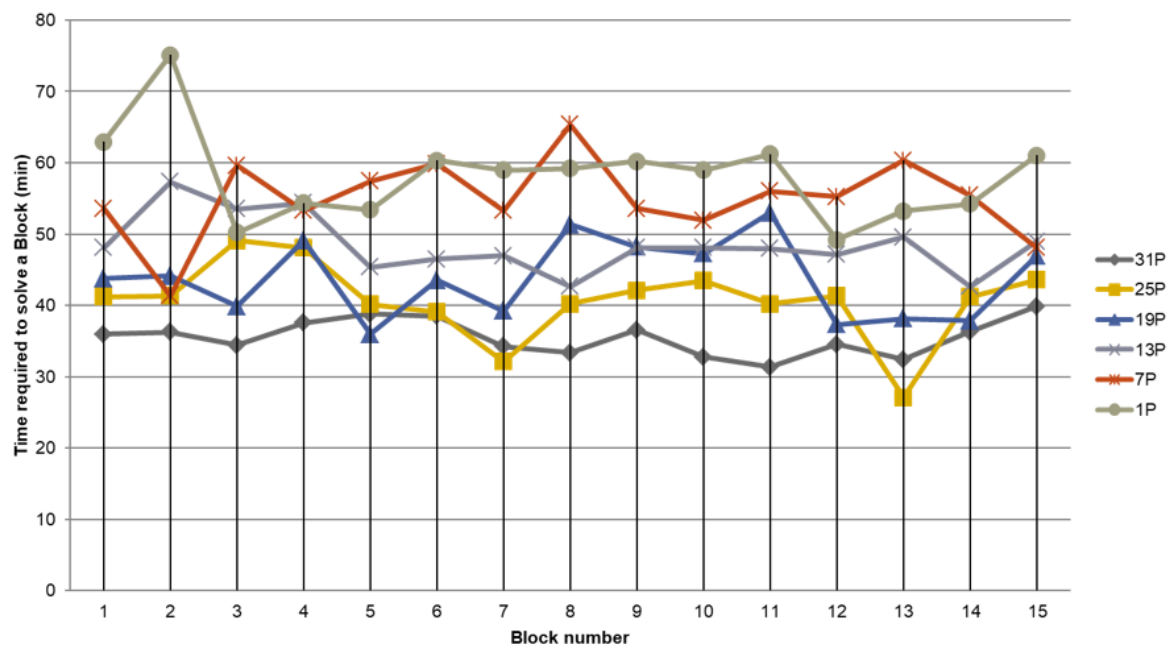


Figure 9. Time to solve 15 consecutive blocks by varying number of peers in 10D difficulty.

The next test increased the difficulty level to 7D, 8D, 9D and 10D, and this increased the block creation time. For example, the average block solving time difference for 7D was 0.55 min. It was 1.33 min for 8D level of difficulty. The average time differences were 2.65 min and 5.51 min for 9D and 10D respectively with the increase of number of peers. Figure 9 represents the time required to solve any 15 consecutive block by different number of peers in 10D difficulty level. It shows more significant result compared to the 6D difficulty. The block-solving time difference between one peer and 31 peers parallel working environment was significantly more compared to the result in 6D difficulty level. For example, the highest and lowest time required to solve a block in case of one peer were around 75 min and 50 min respectively. For 31 peers, those were 40 min and 31 min respectively. If the average times are considered, those are 58.84, 53.63, 48.48, 40.03 and 36.78 min for one peer, seven peers, 13 peers, 19 peers and 31 peers, respectively, with an average of 5.51 min' time difference to create one block.

Figure 10 represents the average time required to solve a block in a different difficulty level with a different number of peers. The required time increased with the increase of number of peers in the same difficulty and increased with the increase of difficulty for the same number of peers. Also, the difference of average time required increased with the increase of the difficulty level. For example, in 6D, the time difference was almost 2.2 min compared with one and 31 peers. In the case of 10D difficulty, it was 22.07 min. This shows that the proposed algorithm was more efficient with the increase of both peers and difficulty level.

To compare parallel mining with traditional solo mining, a similar environment was created. The experiment was done for different difficulty levels with different numbers of peers. The block creation time does not depend on the number of peers, but on the level of difficulty. Figure 11 shows the solo mining for different difficulty levels for any 15 consecutive blocks. The highest and lowest time required to solve one block in 10D difficulty was around 75 and 49 min. For the 6D difficulty it is around 12 min and nine minutes respectively. The average time increased with the increase of difficulty level.

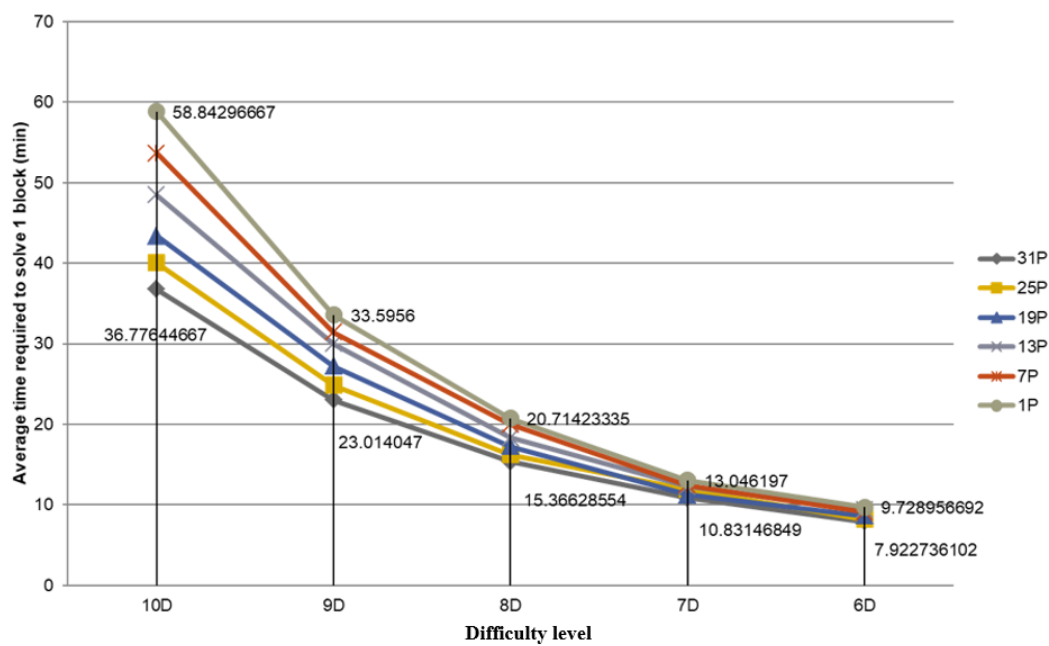


Figure 10. Average time required to solve a block by varying number of peers in different difficulty.

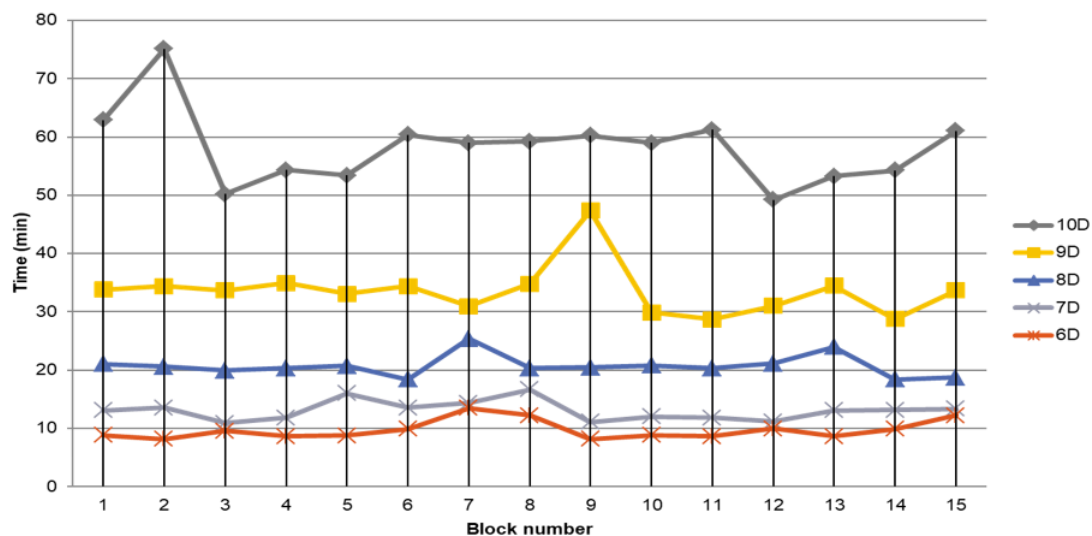


Figure 11. Time required to solve any 15 consecutive block in different difficulty levels in solo mining.

When we compared the solo mining average block creation time in respect to parallel mining for 31 peers, we found similar results for the difference between one peer and 31 peers in parallel mining (Figure 12). The number of peers did not affect the block creation time in solo mining. Here, for 6D difficulty, the time difference was not that significant in 6D difficulty compared to the time difference in 10D difficulty. Thus, we can again conclude that the proposed algorithm was more efficient with increasing difficulty level.

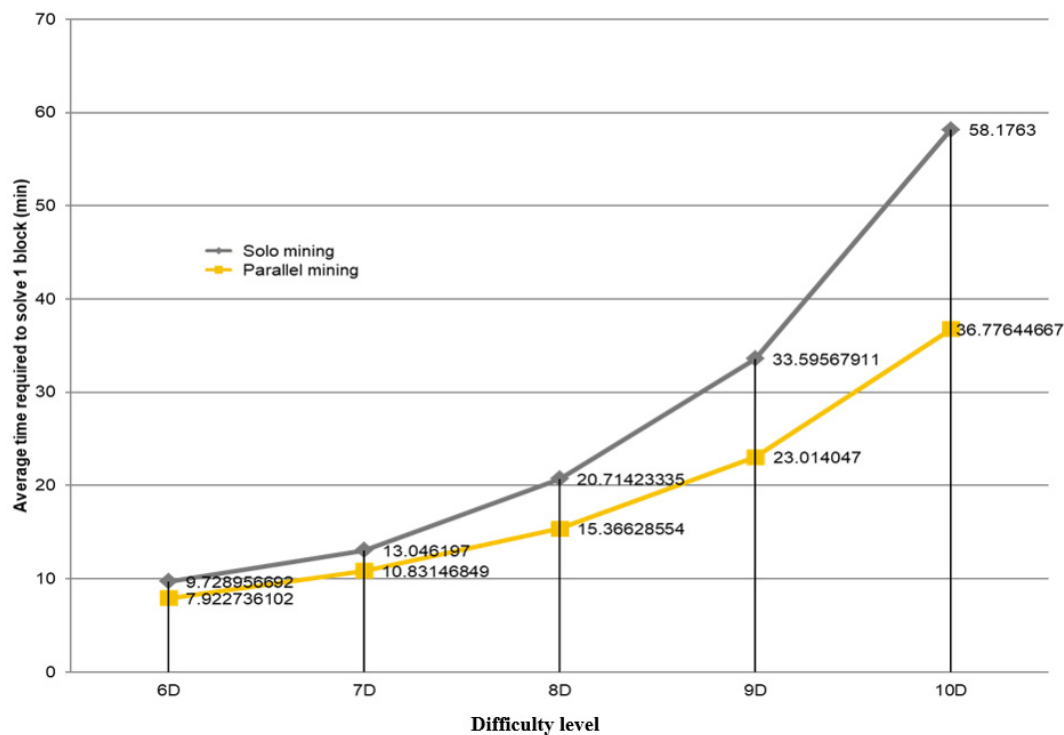


Figure 12. Solo mining vs. parallel mining.

5. Discussion

There are several challenges and case scenarios regarding manager, peer and network behavior, which we discuss in this section.

5.1. Single Point of Failure

In the proposed method, at the beginning of each epoch, all miners have to depend on the manager to obtain a transaction hash and nonces. If the manager goes offline or fails to respond, there can be a single point of failure. However, due to the proposed reward system, this is a very unlikely scenario. Upon fulfilling responsibility as manager, a miner will receive a reward. If unable to fulfill this responsibility, the miner will forfeit the reward as a penalty.

The proposed method is designed in such a way that the duration of the single point of failure will remain only for the one epoch where it happens. If a manager fails to respond, the miner can create the transaction hash and can also generate the nonces. In permissionless blockchain systems, every miner has access to all the transaction records. As a result, for that block, the miner will follow the traditional system with a different nonce and a different transaction hash. This type of epoch will take longer as the miners will perform solo rather than parallel mining. However, the next block will again follow the proposed system since the manager has been decided by the previous block.

5.2. Multiple Miners Solve the Hash at the Same Time

This is a major issue in the current Bitcoin validation process. Bitcoin clients always trust the longest chain. Therefore, if two miners solve the hash at the same time, the block is accepted by most of the miners (at least 51%) who will be added to the blockchain network. The efforts by the other miners will be worthless. This situation may arise in a parallel chain in the network for a certain amount of time. For this reason, clients need to wait for enough confirmed blocks. In Bitcoin, the standard waiting time is six blocks.

In the proposed method, this waiting time decreases. When two miners solve a hash at the same time, one of their solutions will be selected by the manager for the next block as the previous hash.

That data will also be broadcast to all miners by the manager, along with transaction data and range of nonces. The miner whose solution will be selected by the manager will be the manager for the next block. Additionally, the system will not allow more than one miner to be a manager for a certain block. Thus, in comparison to the current system, a parallel chain in the network is not likely.

5.3. Malicious manager

A malicious manager may try to harm a miner by supplying a used range of nonces. In effect, this cannot happen because the manager needs to register each nonce range with the system. The system will not allow the use of the same nonce range by multiple users. Additionally, until he finds a solution, miner information is unknown to the manager. Thus, it is very unlikely that a specific miner will be harmed by the manager.

5.4. New Peers

It is not possible for a manager to know how many peers work at the same time. Thus a manager should continuously create and register nonce range to the network. When a new peer arrives it has to ask for new nonce range and the transaction data. Then he will get the transaction data and a new nonce range which is not used yet by any miner.

5.5. Peer Leaves the Network

A peer can leave the network at any time. It is also possible for a peer in the middle of the processing of proof of work. There is a possibility that the nonce range containing by the peer which is left, may have a solution. However, a block has multiple solutions for different nonce. Thus another peer can provide a different solution for the same block. Thus, the network or the manager will have no impact if any peer leaves the network.

5.6. Peer Asks for New Nonce Range before Finishing the Previously Allocated Range

A peer may ask for new nonce range before checking all the nonces of his current range. In that case, the manager is not able to know if the peer checked all nonces or not. Thus the manager will provide a new range to the peer. However, it is unlikely to do such things by a peer. Because, the speed of checking nonces depends on the processing power of the peer. If he gets a new nonce without finishing the previous nonce range, he can start checking with the new nonce range. However, there is a possibility that the unfinished nonces may have the desired solutions. Thus it is unlikely for a peer to try new nonces range before finishing the previous one.

6. Conclusions and Future Work

In this paper, we proposed a method to improve the transaction speed and scalability of permissionless blockchain networks that are driven by proof-of-work consensus mechanisms. The proposed method introduces parallel Proof of Work in which all miners can together solve the puzzle by taking part in the competition. We have implemented and evaluated the proposed method in a local as well as cloud environment, with results showing significant promise for parallel proof of work as the difficulty level and number of miners increase.

For future work, we plan to evaluate the proposed solution against the 51% attack, and deploy it on the Bitcoin testnet. In addition, we plan to evaluate the reward system and the energy consumption used by the proposed method. Since it improves transaction speed, the proposed method will eventually consume less energy per block compared to the current system. In future work, we will evaluate this aspect, based on a real-time network.

Author Contributions: Writing—original draft preparation, S.S.H.; supervision and writing—review and editing, Q.H.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yli-Huomo, J. Where is current research on blockchain technology?—A systematic review. *PLoS ONE* **2016**, *11*, e0163477. [CrossRef] [PubMed]
2. Scherer, M. *Performance and Scalability of Blockchain Networks and Smart Contracts*; Umea University: Umea, Sweden, 2017; Available online: <http://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf> (accessed on 15 May 2020).
3. Joseph, B.; Andrew, M.; Jeremy, C.; Arvind, N.; Joshua, A.; Kroll, E.; Felten, W. Research perspectives and challenges for bitcoin and cryptocurrencies. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 17–21 May 2015.
4. Cryptocurrency Transaction Speeds: The Complete Review. The Daily Hodl. 2018. Available online: <https://dailyhodl.com/2018/04/27/cryptocurrency-transaction-speeds-the-complete-review> (accessed on 3 November 2018).
5. Understanding Cryptocurrency Transaction Speeds—Coinmonks—Medium. Medium. 2018. Available online: <https://medium.com/coinmonks/understanding-cryptocurrency-transaction-speeds-f9731fd93cb3> (accessed on 12 June 2020).
6. Nakamoto, Satoshi. Bitcoin: A Peer-To-Peer Electronic Cash System. 2017. Available online: <https://bitcoin.org/bitcoin.pdf> (accessed on 25 July 2020).
7. Johansen, S.K. *A Comprehensive Literature Review on the Blockchain Technology as an Technological Enabler for Innovation*, 2nd ed.; Mannheim University: Mannheim, Germany, 2017.
8. Types of Cryptocurrency Hashing Algorithms—Bitcoinlion.Com. Bitcoin Lion—Your Gate to Cryptocurrency. 2018. Available online: <http://www.bitcoinlion.com/cryptocurrency-mining-hash-algorithms> (accessed on 5 June 2020).
9. Crosby, M.; Nachiappan, P.; Pattanayak, S.; Verma, V. Kalyanaraman Blockchain Technology. In *Sutardja Center for Renessereneurship & Technology Technical Report*; Sutardja Center for Entrepreneurship & Technology: Berkeley, UC, USA, 16 October 2015.
10. Hazari, S.S.; Qusay, H.M. A parallel proof of work to improve transaction speed and scalability in blockchain systems. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 916–921.
11. Hazari, S.S. Design and Development of a Parallel Proof of Work for Permissionless Blockchain Systems. Master's Thesis, Ontario Tech University, Oshawa, ON, Canada, 2019.
12. Cong, L.W. Decentralized Mining in Centralized Pools. *SSRN Electron. J.* **2018**. [CrossRef]
13. Le Lann, G. Distributed Systems-Towards a Formal Approach. *IFIP Congress*. **1977**, *7*, 155–160.
14. Dwork, C.; Lynch, N.; Stockmeyer, L. Consensus in the presence of partial synchrony. *J. ACM* **1988**, *35*, 288–323. [CrossRef]
15. Eyal, I.; Gencer, A.; Sirer, E.; Renesse, R. Bitcoin-NG: A Scalable Blockchain Protocol. In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16), Santa Clara, CA, USA, 16–18 March 2016; pp. 45–59.
16. Boyen, X.; Carr, C.; Haines, T. Blockchain-Free Cryptocurrencies. A Rational Framework for Truly Decentralized Fast Transactions. *IACR Cryptol. ePrint Arch.* **2016**, *2016*, 871.
17. Popov, S. The Tangle. 3 April 2016. Available online: http://www.tangleblog.com/wp-content/uploads/2016/11/IOTA_Whitepaper.pdf (accessed on 12 June 2020).
18. King, S.; Scott, N. Ppcoin: Peer-to-peer crypto-currency with proof-of-stake. 19 August 2012. Available online: <https://www.peercoin.net/whitepapers/peercoin-paper.pdf> (accessed on 25 July 2020).
19. How Do Mining Pools Work? Is It Better Than Solo Mining? Captainaltcoin. 2018. Available online: <https://captainaltcoin.com/what-is-pool-mining> (accessed on 20 October 2018).
20. Implementation Code for Parallel Mining. 2019. Available online: https://github.com/shihab2555/Parallel_mining (accessed on 20 July 2020).

21. The Pioneer's Guide to GX—Decentralized Dependency Management on IPFS. Hacker Noon. 2018. Available online: <https://hackernoon.com/the-pioneers-guide-to-gx-decentralized-dependency-management-on-ipfs-90064858f4c2> (accessed on 4 November 2018).
22. Shaker, A.; Douglas, S.R. Self-stabilizing structured ring topology p2p systems. In Proceedings of the Fifth IEEE International Conference on the Peer-to-Peer Computing, Konstanz, Germany, 31 August–2 September 2005; pp. 39–46.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).