Design and Development of a Parallel Proof of Work for Permissionless Blockchain Systems

by

Shihab Shahriar Hazari

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

Masters of Applied Science

in

Electrical and Computer Engineering
University of Ontario Institute of Technology
Oshawa, Ontario, Canada
April, 2019

THESIS EXAMINATION INFORMATION

Submitted by: **Shihab Shahriar Hazari**

Masters of Applied Science in **Electrical and Computer Engineering**

Thesis title: Design and Development of a Parallel Proof of Work for Permissionless Blockchain Systems

An oral defense of this thesis took place on April 3, 2019 in front of the following examining committee:

Examining Committee:

Chair of Examining Committee	Dr. Walid Morsi Ibrahim
Research Supervisor	Dr. Qusay H. Mahmoud
Research Co-supervisor	N/A
Examining Committee Member	Dr. Akramul Azim
Examining Committee Member	N/A
University Examiner	N/A
External Examiner	Dr. Patrick Hung

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

Abstract

Design and Development of a Parallel Proof of Work for Permissionless Blockchain Systems

Shihab Shahriar Hazari
University of Ontario Institute of Technology, 2019

Advisor:
Dr. Qusay H. Mahmoud

Blockchain, which is the underlying technology for the Bitcoin cryptocurrency, is a distributed ledger forming a decentralized consensus in a peer-to-peer network. A large number of the current cryptocurrencies use blockchain technology to maintain the network and the peers use a consensus mechanism called Proof of Work to verify and confirm the transactions. However, the transaction speed in this process is significantly slower than traditional digital transaction systems such as credit cards or PayPal. In this thesis, a parallel Proof of Work model is proposed in order to increase the scalability of the processing of the transactions. The goal of this model is to ensure that, no two or more miners put the same effort into solving the same block. This model differs from traditional Proof of Work or the Bitcoin pool mining in many aspects, such as the responsibilities of the manager, contribution of active miners, and the reward system. A proof of concept prototype of the proposed model has been constructed based on the attributes of Bitcoin. The prototype has been tested in a local as well as in a cloud environment and results show the feasibility of the proposed model.

Keywords: Blockchain, Bitcoin, scalability, parallel computing.

Acknowledgment

First and foremost, I would like to thank Almighty for giving me such an opportunity to work in this field. After that, I am thankful to my respected thesis supervisor Dr. Qusay H. Mahmoud. He provided me a continuous guidelines and suggestions during all the time. I would also like to extend my gratitude to all of my course instructors for their valuable guidance in every step of my learning stage of my graduation period. Finally, I would like to thank my friends and family for their valuable support that has helped me to complete the research successfully.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

PoW = Proof of Work

PoS = Proof of Stake

BFT = Byzantine Fault Tolerance

P2P = Peer to Peer

DPOS = Delegated Proof of Stake

PBFT = Practical Byzantine Fault Tolerance

DBFT = Delegated Byzantine Fault Tolerance

BTC = Bitcoin

ETH = Ethereum

LTC = LiteCoin

GoLang = Go Programing Language

PoI = Proof of Importance

GCP = Google Cloud Platform

CPU = Central Processing Unit

GPU = Graphics Processing Unit

PoC = Proof of Capacity

PoA = Proof of Activity

FPGA = Field Programmable Gate Array

ASIC = Application Specific Integrated Circuit

PoB = Proof of Burn

Prop. = Proportional

SMPPS = Shared Maximum Pay Per Share

PPS = Pay Per Share

Chapter 1

Introduction

In conventional financial systems, a third party is constantly required to verify transactions [1, 3]. 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. Such a mechanism of performing a transaction is derived from the triple-entry bookkeeping [2].

The centralization in transaction brings two major issues. First, it can cause a single point of failure. Banks or financial institutions handle millions of transactions each day from different types of community. If these organizations fail or delay to operate for even a small amount of time, all of its users will be affected. Another major issue is about trust. The transactions or business information are often sensitive. To perform these transactions, the third party handler gets access to that information. If the third party is dishonest, it may leak the confidential information about any business deals.

Peer to peer network is an alternative of the centralized network to transfer data [4]. Here, a two-party can perform a transaction without the inclusion of any third party. However, such a process is not practical for the fiat currency system. As an

example, a bank is not only responsible for verifying a transaction but also for storing the currency. This enables individuals to perform digital transactions without the exchange of any fiat currency. Also, without verification, there is always a chance of fraud or double spending. These issues can be solved in Blockchain. Blockchain provides the fundamentals of a peer to peer network. The objective of Blockchain, especially the permissionless Blockchain, is to build up a decentralized framework [5]. A cryptocurrency uses public or permissionless Blockchain so that everyone can participate in performing the 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.

1.1 Motivation

Though Blockchain can provide a secure and decentralized platform for transactions, it presents some concerns [6, 7]. One of the major issues is scalability. Currently, hundreds of cryptocurrencies on the market currently use the Blockchain network for transactions, mining and maintaining ledgers. All of them are facing scalability issues. On the other hand, VISA, a traditional transaction provider, has already reached a peak of 10,547 transactions per second [8]. The transaction speeds of the cryptocurrencies are much less compare to VISA. The speed for different cryptocurrencies is different due to their respective consensus protocols. For example, Bitcoin uses the Proof of Work technique for block validation whereas

Ripple uses Proof of Correctness technique. Table I shows the transaction speed and confirmation time of different cryptocurrencies which is adapted from [8].

Table 1.1 Transaction speeds of different cryptocurrencies

Cryptocurrency	**Transactions per second**	**Average transaction confirmation time**
Bitcoin	3-7	60 min
Ethereum	15-25	6 min
Ripple	1500	4 sec
Bitcoin Cash	61	60 min
Steller	1000	2-5 sec
Litecoin	56	30 min
Monero	4	30 min
IOTA	1500	2 min
Dash	10-28	15 min

The scalability issue raises another major issue in Blockchain. The transaction verification process for the cryptocurrency is more complex compared to the fiat currency. As the verification process is complex, it needs a vast computational and electrical power. According to [9], Bitcoin, the most popular cryptocurrency, processes approximately 110,000 transactions per day. The total amount of electric power used by the miners to process these transactions is of 215 MW per day on average. This same amount of power can cover the electricity usage of 173,000 American households. In other words, the power required to process each transaction is equal to the daily power required for 1.15 households. This is an excessively large amount of electric resource usage for one cryptocurrency.

1.2 Contributions

The goal of this thesis is to design a consensus mechanism which will increase the scalability and transaction speed in Blockchain network. It was also considered that the core characteristics such as decentralization or security of Blockchain cannot be disrupted. To this end, the following are the main contributions of this thesis.

- Comparative analysis of available consensus mechanisms.
- A framework of the parallel Proof of Work mechanism.
- Proof of Concept prototype of the proposed model.
- Evaluation of prototype from local network and cloud network based on the different case scenario.

1.3 Thesis Outlines

The remainder of the paper is organized as follows. In the next chapter, an overview of our thesis related terminologies and contains a brief discussion on previous works that are already implemented. Chapter 3 describes the working procedure of our proposed system. This chapter also discusses the incentive analysis, case sensitive scenario, and comparative analysis. In Chapter 4, we have illustrated our implementation of the thesis in details. Chapter 5 focuses on the experimental result of the proposed system. The thesis concludes with a summary of research contributions and future plans of our work in chapter 6. This thesis contains two appendices intended for the persons who wish to explore certain topics in greater depth. Appendix A presents the source code of the implementation framework. Appendix B contains the sample output of the distributed ledger.

Chapter 2

Background and Related Work

Blockchain can be compared to a common platform where every participant has equal liabilities, guarantees no data can be missing due to any human or computational faults. Moreover, Blockchain has some unique properties that make it completely different from traditional systems. In this chapter, we present the studies on the terminologies related to the thesis which are important to understand. This chapter also contains a brief discussion about the related works.

2.1 Peer to Peer Network

A peer-to-peer network or P2P network is a group of nodes, where each node is connected to others directly or indirectly without having a central node. Here, the nodes are dedicated to serve each other instead of a central server. The nodes can transfer data among each other maintaining a protocol. When a peer to peer network is established over the internet, the data storing can be done in two ways. Either a central server can be used or a distributed network can be created to share the files among all nodes having a certain protocol.

2.2 Parallel Computing

In parallel computing different parts of a problem is solved simultaneously. This is an efficient way to solve a large problem in a relatively small amount time. The parallel computation can be done in different levels. Such as, data level, resource

level, device level. Currently it is very popular in perspective to energy consumption and heat level increase. Also, it is able to gain high performance in less time.

The efficiency of parallel computing depends mainly on several factors. Such as the distributed search, allocation of resources and election process of a coordinator. To do these, one of the popular algorithms is election algorithm. Here, a coordinator can be changed after a certain period or the coordinator failed to perform his/her responsibility. Our proposed algorithm is based on similar type of technique which is discussed in the following section.

2.3 Blockchain

Blockchain, which is the underlying technology of Bitcoin, provides a peer to peer decentralized network that brings cryptocurrency into play. The blockchain contains the fundamentals that cryptocurrencies require to perform and verify transactions. Other than cryptocurrencies, it can also be used in a decentralized data exchange system. The blockchain can be classified into three main kinds [12]: private, public and federated. In a private (permissioned) Blockchain, the power to change a ledger belongs to a central authority. This type of Blockchain is used within a private organization. In public (permissionless) Blockchain, every node has equal authority to update the ledger. Updating is completed when all or a certain number of nodes in the network reach a consensus. Most cryptocurrencies use this type of Blockchain. Federated Blockchain is a hybrid version of both private and public Blockchain. Here, a number of individuals, rather than a central authority, are responsible for modifying the ledger. These individuals need to reach a consensus in

order to make any changes. Other peers may validate, but only if the individuals responsible have given the authority.

Currently, Blockchain technology is used in a different field. Cryptocurrencies are the major users of public Blockchain. Besides cryptocurrency, internal data communication of an industry, IOT devices data transfer, financial institutions, smart contract, the business community, supply chain, and several other several communities use Blockchain technology.

2.4 Bitcoin

A cryptocurrency is a purely digital currency with no physical state and which is used as a medium of transaction. Verification of a cryptocurrency transaction is achieved by using a cryptography technique. This technique is used not just for the verification, but also for mining or the creation of a new cryptocurrency. The idea of cryptocurrency was invented in the early 80s. A paper published by Chaum [27] which has given the idea to perform transactions without the inclusion of any third parties. This can be done by doing a blind signature in a cryptographic format. The practical use of this theory has implemented in Digicash [28] in 1990. Though, it did not succeed to get people's attention. After that in 1998, the idea of implementing two similar cryptocurrencies has emerged. One is B-Money, which was invented by Wei Dai [29]. It has implemented Proof of Work by solving a complex mathematical puzzle. Another is Bit-Gold which was invented by Nick Szabo [30]. It used the similar idea of using Proof of Work where the solution of the puzzle of a transaction is used as a data to solve the puzzle for the next transaction in order to create a link similar to Blockchain. The goal of both of cryptocurrency is to perform the

transactions without any central authority. Though, none of them has been qualified to implement in the real world.

The first successful cryptocurrency, Bitcoin, was invented in 2009 by an unknown programmer or group of programmers using the name Satoshi Nakamoto [10]. This currency successfully got to introduce a peer to peer system with no central authority for making a transaction. The concept is similar to B-Money and Bit-Gold. As there is no centralized authority, every peer of the Bitcoin network maintains a distributed ledger. This ledger contains all transactions occur in the network. The transaction records the sender and recipient's public keys as their identity and the amount of currency to be transferred. Before performing a transaction, the sender needs to input his/her private key, similar to the basic cryptography protocol. When a transaction is initialized, it is verified by any other peer using the public ledger. After verification, it is broadcast in the network and other peers update their public ledger. Once a transaction is broadcast, it cannot be modified.

Cryptocurrencies have solved many issues, including centralization, double spending, and security concerns. The value of a traditional currency is not fixed but can change over time according to several factors, such as the global political situation, environmental impact, availability of natural resources, and the stock exchange. In contrast, cryptocurrency value does not depend on these factors; rather, the change in the value of cryptocurrency is fixed and pre-defined for a specific period of time. For these reasons, the popularity of cryptocurrency is increasing regularly. In today's market, 2,072 cryptocurrencies are available. The total cryptocurrency capitalization is more than USD 114 billion [11]. The top 20

cryptocurrencies contain almost 89% worth of the market, where the capital is Bitcoin is almost half of the capital of all cryptocurrencies.

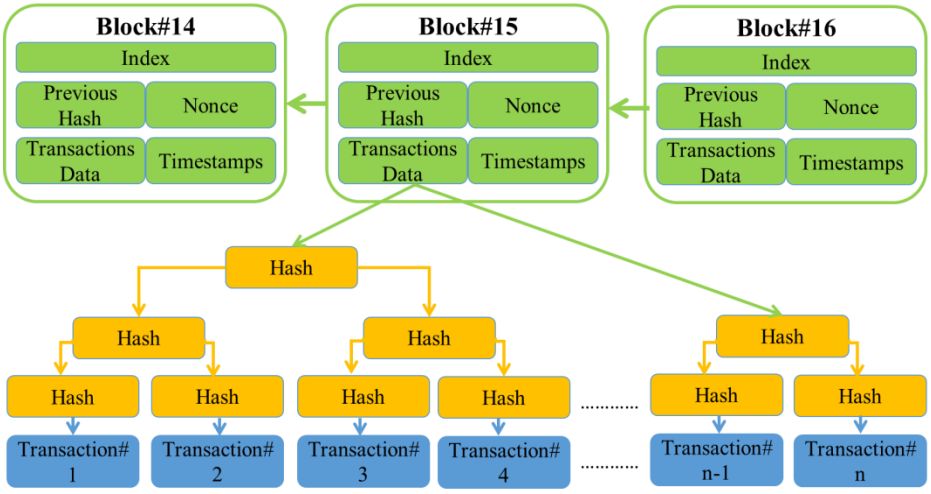

Figure 2.1 Blockchain network of Bitcoin

2.5 Distributed Ledger

A Blockchain is an open ledger that gives data related to all members and every digital exchange that is completed. Blockchain consists of several Blocks that contain information about all the transactions that occur in the Blockchain network [31]. Every Block contains two hash values along with the transactions. One value is its own while the other is from its previous Block. The process of creating a new Block starts immediately when a block data is entered into the Blockchain. Every participant in the system has the right to verify the information and receive a copy of every activity that has occurred via the Blockchain network. This copy is called a distributed ledger. Any progressions to the ledger are instantly notified to all the participants with access to the ledger. Using a cryptography method by means of digital signatures and keys, the security and precision of transactions are maintained

cryptographically and are controlled by the members. The distributed ledger keeps the system from malicious attacks such as double spending, deletion or modification of the block record. It also assures the ownership of the property or assets.

2.6 Consensus Mechanism

Public Blockchain maintains a decentralized system on a global scale. Here, thousands of peers contribute in order to verify or validate transactions. In such a powerfully changing status of the Blockchain, these openly shared ledgers require an effective, reasonable, ongoing, practical, dependable, and secure instrument to guarantee that every one of the transactions happening on the system is real and that all members concur on the status of the ledger. To ensure all this, it is very important to maintain a consensus mechanism, which is similar to the Byzantine Fault Tolerance mechanism [13]. Byzantine Fault Tolerance (BFT) denotes how many failures a system can consider. It allows no restrictions or cannot make any assumption on the decision taken by a single node of the system. A node has the ability to generate distinct data in this type of situation. As Blockchain has no central authority, by using such a mechanism, all or a majority of peers can settle on a decision following certain rules. As a result, all peers can maintain a common public ledger. A decentralization network must maintain the BFT mechanism. Without having that, the node can upload malicious data to the network, which can destroy the reliability of the system. Moreover, there is no central authority which can take over the control of the system to repair the damage. Thus, the system will lose its credibility.

The mechanism for reaching consensus in a decentralized system can be classified into two major parts. Those are, election based and proof based [14]. In an election based system, the node or group of nodes responsible for reaching a consensus are mostly pre-elected. Thus, the number of such nodes is limited, which makes it easier to reach consensus. In contrast, all nodes have equal rights in most proof based systems. Here, all or a majority of peers must reach consensus. Thus, a considerable number of messages need to be exchanged among all the nodes. In order to verify a transaction, the nodes have to do some work as proof of verification, which increases the complexity of a decentralized system.

2.6.1 Proof of Work

Proof of Work (PoW), the most popular consensus protocol in cryptocurrency, first came into the practical use of play with the invention of Bitcoin [10]. This consensus mechanism is used to verify and validate a transaction as well as to mine the currency. In order to perform a Proof of Work, a miner has to create transaction data with one or multiple unconfirmed transactions to create a block. A miner is responsible for verifying and validating transactions. Any peer in the network can be a miner. After creating the transaction data, the miner has to solve a cryptographic puzzle. Here, the cryptographic puzzle is a hash problem with a given difficulty. This difficulty regulates how much time is required for a miner to solve a block. Along with transaction data, the miner must also take the hash of the previous block as an input. In this way, every block is connected to the next block, thus forming a chain. The miners compete with each other with their transaction data to solve the puzzle for a certain block. When a miner finds a solution, s/he broadcasts it to the network

and other miners then validate it. Following validation, the block is added to the network and the miner who solved that block is rewarded. Bitcoin and Litecoin use Proof of Work as their consensus mechanism. Most of the cryptocurrencies use the PoW method to maintain the consensus around the peers in the network.

Completion of a Proof of Work requires considerable computational power as well time, based on the level of difficulty. As an example, Bitcoin requires an average of 10 minutes to solve each block. As the network grows, the puzzle becomes more complex. These aspects make the network very secure since significant computational power, more than half of the total combined computational power of all the miners, would be needed to attack it. Even if an attack were possible, it would cost too much. All of these factors make an attack futile. Another positive outcome of this consensus is the possibility of becoming a miner. Each peer can be a miner with the required computational power.

However, this consensus mechanism faces scalability issues due to the considerable time needed to solve a block, and for which an extensive amount of computing power is also needed. Multiple miners compete with each other using their computational power, where only one miner will be able to succeed. As a result, with the exception of the winner, the efforts of all the other miners will be wasted. This represents considerable misused energy.

2.6.2 Proof of Stake

Proof of Stake (PoS) involves the creation and validation of a new block with no competition amongst miners. In fact, there are no miners in Proof of Stake. Here,

validation is achieved by a ‘validator’. One major difference between Proof of Work and Proof of Stake is that in the latter, rather than mining currency, the validator receives only a transaction fee for creating a new block. This means that the amount of total currency in the network always remains fixed.

Table 2.1 Differences between Proof of Work and Proof of Stake

Criteria	**Proof of Work**	**Proof of Stake**
The probability of being a validator for the next block	Based on CPU power.	Based on the amount of stake and coin age.
Block reward	Yes	No block reward; validator receives a transaction fee.
Cost-effective	No	Yes
Need to solve a complex mathematical puzzle	Yes	No
Security	Potential 51% attack based on the hash power.	51% attack has the low possibility as the security does not depend on the hash power.

In Proof of Stake, the validator is elected in a pseudo-random way before starting the validation. Only the elected validator can validate a subsequent block. Each time before creating a block, a validator is randomly selected. In order to be elected, the user has to put some of his/her own currency at stake. The user who puts more

currency at stake has more chance of being elected. Once elected, a user can create a new block and is rewarded with the amount of currency staked along with the transaction fees. The other users receive back the amount of currency they put at stake. In Lisk and Nxt, Proof of Stake is used to create a new block [10].

2.6.3 Other Consensus Mechanisms

Majority of cryptocurrencies use either PoW or PoS. Besides, there are some other popular consensus mechanisms which are used in different cryptocurrencies [50]. Some of them are briefly discussed below.

- **Proof of Burn:** The Proof of Burn (PoB) consensus mechanism was invented by Ian Stewart [32]. Here, miners send some coins to a random invalid unknown address before creating a block. The address changes after each block is created. As it is an invalid address, the coin which is sent to that address is unusable or burned. This address is also known as an 'eater address'. Among the miners, only one is able to create the next block and receive a reward. Slimcoin uses PoB as consensus mechanism.

- **Proof of Capacity:** The Proof of Capacity (PoC) algorithm privileges on the capacity of a miner's storage rather than hashing power. The goal of this mechanism is to decrease the usage of computational energy, as is the case in Proof of Work. Instead of calculating the hash in every block, Proof of Capacity allows storing the list of possible solutions, even before mining the block. The miner who has more space can store more solutions, which provides the miner with an advantage to solve the block [33]. This mechanism is used in Burstcoin.

- **Proof of Activity**: The Proof of Activity (PoA) is a hybrid solution from both Proof of Work and Proof of Stake. At the beginning of the mining process, all miners start to compete with each other, similar to Proof of Work. The process changes to Proof of Stake when a block is mined. At that time, the block contains the header with the miner's reward information [34].

- **Proof of Importance:** Proof of Importance (PoI) is an advanced consensus mechanism similar to Proof of Stake [35]. To eliminate the drawback of the rich becoming richer, which exists in Proof of Stake, the Proof of Importance mechanism introduces some new regulations, including a score-based protocol known as the Proof of Importance score. A participant with a higher score has an increased possibility of being selected as a validator. This score is calculated according to three factors: vesting, transaction partner and number and size of transactions in the previous 30 days.

- **Practical Byzantine Fault Tolerance:** The Practical Byzantine Fault Tolerance (PBFT) is a real-world replication of BFT consensus mechanism. In general practice, in the case of cryptocurrency, a group of individuals is predefined to validate the transactions in a PBFT model [32, 36]. When a new transaction arises, the predefined group receives the transaction and reaches a consensus. Among the nodes, one node is considered as a leader node and other nodes as the backup node.

Various other consensus mechanisms are used in different cryptocurrencies. Among them, the most significant consensus mechanisms are Proof of Membership; Proof of Luck; Proof of Elapsed Time; DBFT; Proof of Authority; and Delegated

Proof of Stake [36, 45 - 47]. All of these mechanisms have their own unique properties, with different applications.

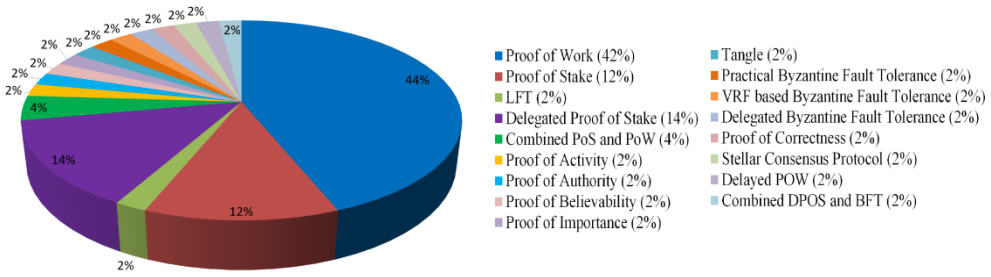

Figure 2.2 Consensus mechanisms of the top 50 cryptocurrencies based on the current (December 2018) market capital

Based on their techniques and characteristics, different Blockchain based consensus mechanisms can be divided into four major groups: Proof of Work; Proof of Stake; a hybrid or combination of both PoW or/and PoS and Byzantine Fault Tolerance with different versions. A short evaluation of them is discussed below. However, the evaluation of a hybrid or combined mechanism of both PoW or/and PoS is not mentioned in the following. Because the features of different cryptocurrencies are different which follow the hybrid or combined protocols. Thus, it cannot be generalized.

Table 2.2 Evaluation of consensus mechanisms

Consensus mechanism	**Proof of Work**	**Proof of Stake**	**Byzantine Fault Tolerance**
Energy consumption	Wastes considerable energy.	Less energy consumption.	Less energy consumption.
Advanced hardware requirement	Required.	Not required.	Not required.
Centralization	Decentralized.	Partially centralized.	Centralized.
Scalability	Not scalable.	Scalable.	Scalable.
Security	The attack is possible with 51% hash power, which is impractical in the real world.	Removes 51% attack threat	May have a single point of failure.

2.7 Existing Scalable Solutions

Selection of a coordinator is extremely useful for improving the performance of a distributed system. In this approach, which was first implemented by Gerard Lelann [15], a consensus protocol is proposed with a coordinator election for a partially synchronous processor [16]. 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 [17]. 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 micro block. The key block contains the leader information as well as information about the previous block. The micro block 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 micro blocks using his/her private key which contains the transaction information. The amount of micro block issued to the leader is dependent on signing speed and delay network propagation. The micro blocks 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 [18]. 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.

Lewenberg, Yoad, Yonatan Sompolinsky, and Aviv Zohar [41] proposed the Directed Acyclic Graph network for consensus mechanism. This mechanism was picked by IOTA [19] later to do the Proof of Work parallel. In this case, many peers validate the transactions in parallel and the scalability increase with the increase of peer. However, IOTA is facing the security concern issue where the network can be attacked with 34% computational power of the whole network [42].

Sharding is another effective solution for increasing the scalability of Blockchain [39]. The concept of sharding in Blockchain arose from horizontally dividing a vast

database into portions and processing all portions in parallel. Sharding in Blockchain uses the same concept where there are different chains that will run in parallel containing transactions. Each chain will have different unique properties and will contain several nodes. A transaction can only occur between the nodes in the same channel. If a transaction needs to occur in a different channel, it needs to maintain a certain protocol. Gao, Yuefei, and Hajime Nobuhara [40] proposed a combined model of sharding and Proof of Stake to make the consensus more scalable. Besides, there are some other significant scalable solutions such as SegWit [43] and Lightning Network [44], in which the data load in the main chain is decreased to increase the transaction speed of the network.

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. Here, they use game theory to distribute the work [48]. In game theory, the action of a participant depends on the action of other participants. The work load in pool mining for the miners follows the similar way. 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. Many methods, such as SMPPS, PPS, and Prop., exist for distributing rewards [21]. They are different from each other based on the distribution of reward technique. For example, PPS provide a fixed amount of reward

to all contributors, regardless of the amount of total pool earning. On the other hand, in SMPPS, each contributor gets the maximum profit based on pool earning. In Prop., the reward distribution is proportional based on the contribution but not maximum. However, the process of solving the Proof of Work, mining and reward are different in the mining pool than the actual Bitcoin.

2.8 Gaps in the State of the Art

Blockchain provides a trustless network with anonymous nodes. In the case of cryptocurrency, the miners work separately to create the Block. The existing solutions increase the scalability of the network preserving this property. Such as, decreasing the size of the transactions or the transactions is classified in parallel chains. However, in all those solutions the miner works separately. As a result, for every Block, the effort of all miners except the successful miner, become useless. The existing mechanism where every miners or validators cooperate with each other (Practical Byzantine Fault Tolerance or Pool mining) to create a certain Block brings centralization to the network. This may bring a single point of failure or security concern. The proposed parallel Proof of Work [49] motivates the miners to solve the puzzle by distributing the amount of work. Also, it maintains the decentralization and anonymity in the network. Besides, it also can save a lot of energy consumptions which became a great concern for Bitcoin or other Proof of Work based cryptocurrency. The miners co-operate each other by participating in the competition similar to game theory.

2.9 Summary

Blockchain is a vast concept with many resources and use cases. A brief description of different parts of Blockchain has provided which are related to our thesis. This includes different types of Blockchain technology, uses of Blockchain technology, especially the public Blockchain, current status of cryptocurrencies in the market with their processing algorithm. The basic idea of consensus mechanism framework has also discussed, along with the elaborate discussion of PoW with its use cases, limitation and challenges. The proposed framework will be presented in the next chapter.

Chapter 3

Parallel Proof of Work

In this thesis, we have focused on developing a scalable consensus mechanism to increase the transaction speed. In order to do so, the proposed model was designed based on the consensus mechanism currently used by Bitcoin. Bitcoin provides the secured and decentralized framework comparing to the other Blockchain applications. Currently, it is the most popular cryptocurrency in the world. That's why the attributes of the consensus mechanism of Bitcoin has been chosen to design the model and build the prototype. The proposed model was designed in such a way that, it can still be decentralized and secured with increased scalability. This chapter elaborately discusses the architecture of the proposed system. Besides, it also discusses the challenges and solution of different case scenario which may arise due to the proposed algorithm.

3.1 Network Architecture

To perform the Proof of Work, some of the data used by the miners are identical, including the block 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. As the miner works separate, it may happen that multiple miners can use the same transaction data and nonce to create the next block. As the miners do not share the data they are using to find the cryptographic solution, there is no way to know that if they are using the same data. Again, as the miner competes with each other to

create the same block and only one miner can be the successful miner (who find the solution first), the effort of all other miner become completely worthless. That decreases the scalability of the network and wastes a lot of energy. To get rid of this scenario, the proposed model is designed in such a way that, all miners can work parallel and no multiple miners do the same work. In order to do so, 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.

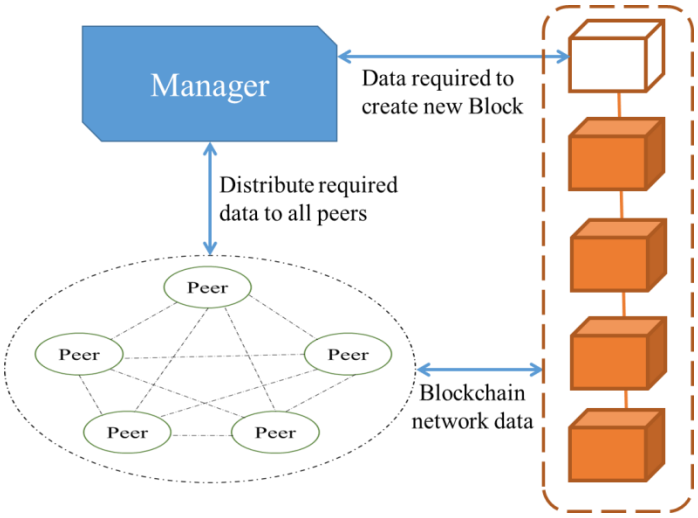

Figure 3.1 Network architecture for parallel Proof of Work

3.2 Architecture Breakdown

The proposed architecture includes a new component which differs it with PoW consensus mechanism followed in Bitcoin. Following are major changes with the traditional method.

- A manager is required which is absent in the traditional model.

- The manager needs to distribute the data to all other miners maintaining the given protocol which is absent in the traditional method.
- The manager changes in every block based on the performance of the miners.
- Reward system is different to defend the single point failure and malicious attack in the system.

3.2.1 Role of a Manager

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 duplication can be checked in two chunks by using linear comparison. The implementation prototype used a similar technique. 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 the traditional way, every miner chose the transaction data and nonce value on their own. In the proposed method, 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.

3.2.2 Distribution of Data

At the outset, the manager will create a transaction hash with the unconfirmed transactions and, at the same time, will generate several chunks of nonces. Each chunk will contain a range of nonce values. In each chunk, the nonces can be random or certain. However, no multiple chunks can have the same nonce value. If m numbers of miners are active in the network, the manager must initially generate and register at least m number of chunks. The manager will then distribute the transaction hash and chunk of nonces to each active miner. The system will ensure that no two miners have the same chunk. 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.

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 or the system 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 chunks of nonces as possible. The process will continue until a designated solution for the current Block is found.

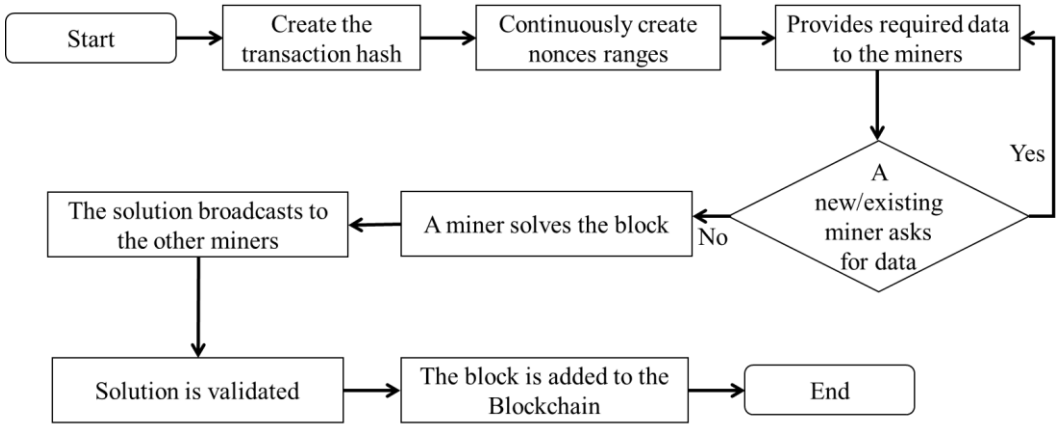

Figure 3.2 Workflow of a miner as a manager

3.2.3 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. The manager of block 1 can be chosen in two ways. Either randomly or the node which connects to the network first. 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 $(n+2)^{th}$ Block. In the following, the process of selecting a new manager is shown. Here, M5 has solved the Genesis block; hence s/he will be the manager for the 2^{nd} block. After solving the Genesis block, M5 will still act as a

regular miner for 1st Block. When 1st block is solved, M5 will act as manager for the 2nd block and cannot compete with other miners as would a regular miner. In the same way, Block 1 is solved by M12. Thus s/he will be the manager for 3rd block. In 2nd Block, s/he will act as the regular miner who will compete with other miners to solve the block. If a miner solves two consecutive blocks, s/he will be the manager for the next two consecutive blocks.

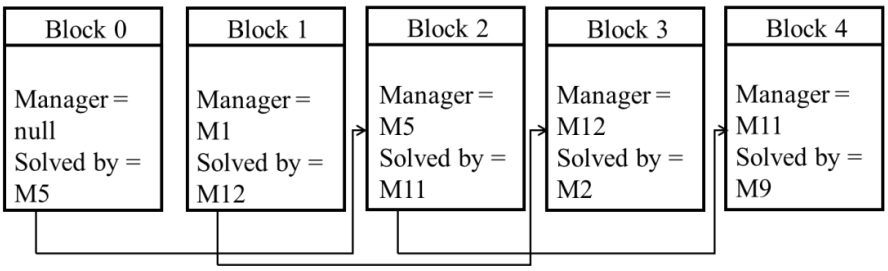

Figure 3.3 The process of manager selection

3.2.3 Reward System

In the current 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 system, having created a block, the miner will be able to mine a certain amount of cryptocurrency, similar to the current system. 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 the following figure. Here, we illustrate the total reward (Transaction fees and mining currency) for a miner. For the given figure it is M12. M12 solves the block

N. Thus s/he will get the 35% transaction fees of all the transaction in block N. After that s/he will become the manager of block N+2. After solving that block s/he will get 65% transaction fees of that block. Additionally, s/he will get the corresponding mining reward to solve Block N.

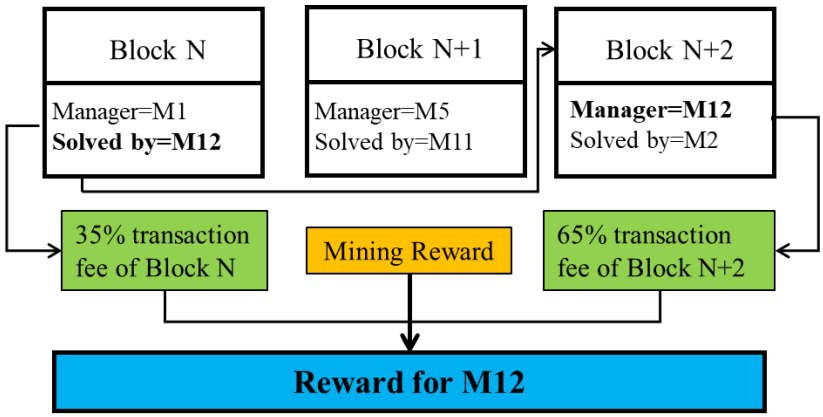

Figure 3.4 Proposed reward system

3.3 Features and Attributes

The goal of the proposed model was to design a scalable solution maintaining the core properties of Blockchain technology. How we can get that from the proposed model and how it does not violate any core features are discussed in the following.

3.3.1 Transaction Speed

The transaction speed depends on the time required to create a Block. Again the Block creation time depends on the processing power of a miner by which they can find the solution of the cryptography puzzle after several trial and errors. The time of solving the puzzle can decrease with the increase in processing power. That's why the miners invest a lot for the processing power to get more processing speed. Initially, at Bitcoin, CPU was used to solve the puzzle. Then the machine upgraded

to GPU, FPGA and currently, they are using ASIC machines as puzzle solving processors. Though, the average Block creation time is still the same (10 minutes) because they adjust the difficulty in every two weeks to keep the Block creation time unchanged. If we consider the constant difficulty, we can reduce the Block creation time without adding more processing machine in parallel mining. Traditionally, all miners work in separate. Here, the miners will work in parallel to find the solution. As the miners are working in parallel, their combined computational power will be used to find the solution of a certain block. Eventually, it will decrease the block solving time and transaction speed. Because the more processing power we can add for a certain Block, the more we can be able to find the solution for that Block. 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 Energy Consumption

The consumption of energy is dependent on the block solving time. The more block takes time to solve, the more energy consumptions occurs. For PoW based cryptocurrency it is a major issue and it is increasing day by day. According to Digiconomist [37], the estimated current energy consumption (October 2018) of Bitcoin and Ethereum is 73.121 TWH and 18.98 TWH per year. It was 19.625 TWH and 4.242 TWH in October 2017 respectively. The proposed model can decrease the energy consumption for each block by decreasing the bock solving time. The energy consumption is directly proportional to the Block solving time for a single miner. Again, in PoW, all miners compete with each other and the only one can win. Thus

the energy consumption of all other miner becomes waste. Here, the combined energy of all miners will be used to solve a particular block. This can save a lot of energy consumptions.

3.3.3 Fairness to Miners

In this system, every miner has an equal opportunity to be a manager. A miner can be a manager only by solving a Block. Thus a miner can be a manager only by contributing to the network. This makes the process fair. For example, in PoS the validator can be selected with the amount of stake. That brings an issue of rich being richer. Again, the question may arise that the process is not fair to the miner who invests more in mining machine. Currently, the miners who can have more processing power have more probability to be the winner. If parallel mining is done, the probability will be equal to all. This won't be fair to the miners who invest more in their processing power. However, that is not practically true. The miners who have more processing power can still calculate more nonces than the other miner. Eventually, they can finish their allocated range earlier than others and ask for a new range to the manager (If the Block is not solved by that time). Thus s/he always has more probability to solve the puzzle earlier than others but it does not provide any guarantee like current mining system of Bitcoin.

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. The manager gets the 65% reward of transaction fee and the miner who solved the Block gets 35% reward. The manager is getting more than the miner because the transaction hash, for which the rewards are splitting, is created by the

manager. The miner who solved the Block will get his remaining 65% reward in the next Block after. Additionally, each manager will get the mining reward which does not split, after finishing his/her responsibility as the manager.

3.3.4 Decentralization

Both the current system and the proposed technique increase the probability to solve the puzzle to a miner with more processing power. 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. Again, any miner can be the manager and the manager changes in every epoch. This allows for more decentralization in the system.

3.3.5 Security Concern

In the proposed system, the manager is responsible for distributing the transaction data and the chunk of nonces. However, the process for validating a Block is the same as followed in the current Bitcoin consensus mechanism. In Bitcoin, to fabricate a Block data, the hacker needs to acquire at least 51% computational power of the whole network. By doing that, the Block validation can be delayed; the confirmation for the new transaction can be prevented. Also, this type of attack can reverse the transaction data. However, this type of attack is impractical in the real world. Because the combined hash power of the cryptocurrency such as Bitcoin or Ethereum is very large. Bitcoin consumes almost 41,483,931 terahashes. To do a 51% attack, an attacker needs to invest at least 1.4 billion USD to perform such an

attack. The cost is based on the full efficiency of current ASIC mining devices, 22% infrastructure cost, and 10% labor cost. However, if an attacker can able to perform such an attack s/he needs at least 194 days to get the investment back. This is possible only the attacker can solve the entire block created in these days. This is impractical in the real world. Also, the other miner will leave the community in such condition which will decrease the value of Bitcoin. Thus the platform is secured based on the economy. As the same process is followed in the proposed system, it also brings the same security strength in the network.

3.4 Comparison with Bitcoin Pool Mining

Bitcoin pool mining is a platform where many miners works together to solve a Block combining their hash resources. Here, the miners who cannot afford to have powerful computing machines come together and combine their individual computing machine to create a large powerful one. They can also take part directly in Bitcoin mining. Currently, there are many mining pools available such as BTC.com, Antpool, Slushpool, F2pool and so on [22]. Pool mining and proposed parallel mining encourage the miners to combine their mining resources. Though, there are many differences between these two processes when these are deeply considered. The major differences between these two processes are discussed in the following table.

Table 3.1 Comparison between pool mining and parallel mining

Attributes	Pool Mining	Parallel Mining
Centralization	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.
Difficulty target	The difficulty target assigned in a pool mining is less than the actual target in Blockchain mainstream.	The target in parallel mining is the same as the target in Blockchain mainstream.
Rewards	The rewards split to all participant based on the contribution of the miners.	Only the successful miner gets all mining rewards. Transaction fees split between the manager and the successful miner.
Responsibility of coordinator or manager	The coordinator responsibility involves the assignment distribution to the miners, split of rewards, checking the contribution of each participant.	The manager responsibility includes distributing of transaction hash and nonces ranges.
Pool fee	Pool mining coordinator may take a small amount of reward from each participant. There may be a participation fee for the miners.	There is neither reward fee nor participation fee for the miners.

3.5 Incentives in Diverse Spheres

The proposed algorithm is developed for public Blockchain where the transaction can be done with necessary rewards. Different types of the community can influence the network by providing service and accepting rewards. The community is divided into four major types. These are developers, miners, general users, and traders.

Table 3.2 Influence of different community in the network

Community	**Service to the network**	**Rewards Achieved**	**Influence on the network**
Miner	Miners verify the transactions, create the Block and validate the Block.	They are rewarded by mining fee and transaction fee.	The scalability depends on the number of active miners and processing machine used by the miners.
Individual users	They create transactions and pay transaction fees.	They get faster transactions in a safe environment.	They can stop the transaction and make the system worthless.
Trades	Provide liquidity to the market and a fiat denominated value to the cryptocurrency.	They can make a profit by successful trading and hold the cryptocurrencies.	The can control the supply and price of the cryptocurrency to the market.
Developers	They can propose new features by upgrading the network.	They get to be paid by developing the network.	The improvement of the network can be implemented by them.

3.6 Challenges and Solutions

The proposed mechanism can face different challenges regarding manager, peers and network behavior. The possible case scenario and the solution of these challenges are discussed following

3.6.1 Single Point of Failures

In the proposed solution, 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, this is very unlikely due to the proposed reward system. A manager can only get a reward by appointing as a manager and finish his/her responsibility as the manager. If s/he fails to do that, s/he will not get any reward which includes the transaction fee reward and mining reward, in spite of solving a Block.

Let's assume the manager goes offline. In that case, the proposed system does not face a complete breakdown due to network architecture. Every miner has access to the Blockchain mainstream. Thus if the manager goes offline, the miners always have access to get the data from the network. In that case, the miner can solve the puzzle separately like the traditional system. For this scenario, the Block solving time will take the same time solo mining. However, this solo mining will continue only for one Block. The manager for the next Block has been decided in the previous Block. When the epoch of the current Block will be over, the miner who solved the Block will be selected for the manager of the next Block after. In the next Block, the system will again switch into parallel mining from solo mining.

3.6.2 New Peer Arrives

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

3.6.3 Multiple Miners Solve the Puzzle at the Same Time

This is a major issue in the current Bitcoin validation process. Bitcoin clients always trust the longest chain. Therefore, if multiple miners solve the hash at the same time (usually two), 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 form a parallel chain in the network for a certain amount of time. The miners who accepted the solution from the first miner will try to create the next Block based on that. The other miners will do the same thing based on the other Block. If the next Block is solved based on the top of the first miner, this becomes the longer chain and the miner turn into this chain. If the next Block is solved on the top of the second miner, all of them turn into that chain. Either case only one miner can get the reward.

In the proposed system, this can be solved in two ways. 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, this problem can be solved immediately.

Another solution to this issue can be solved as a traditional way. In that case for the next Block, the miners will get divided into two groups. One group will use the previous hash data obtained from the first miner and another group will use the previous hash data from the second miner. In that case, the combined computational power will be divided into two groups. This may decrease the scalability of the network. However, this scenario will remain for only one epoch. After that, every miner will turn into any one chain. For this type of scenario, a miner can be the manager of for the next block after.

3.6.4 Malicious Manager

A malicious manager may try to harm a specific miner by supplying a used range of nonces. In that case, the target miner cannot able to find the solution as the allocated range is already been used. However, in a Blockchain network, the identities of all peers are anonymous. Thus it is not possible for a manager to harm a certain miner. Again, according to the proposed consensus mechanism, the manager needs to register each nonce range with the system. The system will not allow the same nonce in two different chunks. While a nonce range is chosen by a miner, only s/he will be able to use that range. Once a range will be chosen, the system will not allow any other miner to pick that range for that certain Block.

3.6.5 Peer Leaves the Network

A peer can leave any time from the network. It is also possible for a peer to leave in the middle of the processing of Proof of Work for a certain Block. There is a possibility that the nonce range containing by the peer, may have a solution for that certain Block. If s/he leaves the network without using every nonce allocated to him/her, the network will not get that solution. However, a Block can have multiple solutions. A solution for a certain Block should be same or less than the target. There are other solutions with different nonce value. Eventually, any other miner can find a different solution for the certain Block and can create that Block. Thus, leaving a peer from the network does not affect the network or the consensus mechanism.

3.6.6 Malicious Peer

A malicious peer may ask for a new nonce range before checking all the nonce of his current range. In that case, the manager or the system is not able to know if the peer checked all nonces allocated to him/her. If a peer asks for a new nonce range, s/he will get the new range immediately from the system. However, it is unlikely to do such things by a peer. If a malicious peer does not finish all nonces allocated to him, it may possible that the required solution may have in those unchecked nonces. Thus it is a bit risky for a peer to do such things. The speed of checking nonces depends on the processing power of the peer. As the processing power of the peer is fixed, s/he has to leave some nonce unchecked. This scenario does not affect the network. The network will not get that solution (If the unchecked nonces have a solution), but there are multiple solutions for a certain Block. Any honest miner can find the desired solution and create the Block. Again, the number of a chunk of

nonces created by the manager is a lot more than the number of a miner. Thus, if a malicious miner acquires extra nonce ranges, it will not affect the other honest miners.

3.7 Use Cases

The proposed model is designed for public Blockchain platform. Also, the mechanism uses PoW for the verification of data. Based on that, it may have the following use cases.

- **Transactions:** The model can be used for making transactions through cryptocurrency. A new or existing cryptocurrency can follow the model to get a scalable network.
- **Store of Value:** The model contains a secured distributed ledger. Thus, it can be used for storing value or data.
- **Smart Contract:** Currently, smart contract is very popular for cryptocurrency. A trade can be made with the mechanism through smart contract.
- **Data Management:** Important data can be stored and verified using the model. In this case, the reward system can be modified.
- **Supply Chain:** The model can be used on the platform similar to supply chain, where a mass communication needs to be done. Also, in this case, reward mechanism can be modified or removed.

3.8 Summary

In this chapter, the proposed model is discussed elaborately with architecture breakdown. Besides, the feature and properties of the proposed model, comparison with the current model and different case scenario are also discussed. The following chapter will discuss the implementation technique of the proposed model.

Chapter 4

Implementation

This chapter describes the implementation technique for the proposed system. The cryptocurrency model has two types of community. These are general users and miners. The proposed model only impacts the miner community in case role play. Thus to build the prototype, the network architecture regarding the miners are considered.

4.1 Prototype Overview

A peer to peer network has been developed to implement the proposed solution. The implementation is done in local and cloud system. In both cases, the same network architecture has been considered. The network has been created with a logical ring structure [22]. Thus, each node can connect to maximum of two nodes except the first node. It can only connect to the next node. 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 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. The following figure represents the network architecture with different IP and a 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. If a new node wants to connect the

network, it can connect through the last peer only. When a peer leaves the network, the nodes were connected to that node, connect with each other.

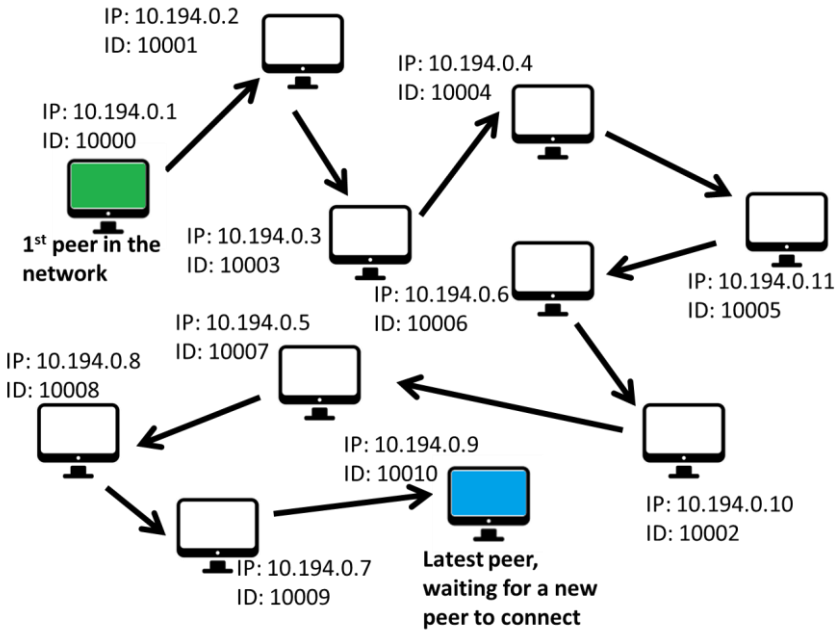

Figure 4.1 Peer to peer network diagram to implement the solution

When a peer acts as a manager, a temporary connection is established with him/her to all other peers. Through this connection, the manager can send the transaction data and nonces data to other peers. This connection changes in each epoch. The peer to peer connection remains unchanged. The validation of a new block is done by the peer to peer network using the gossip protocol. When a consensus is reached, the temporary connection breaks and it established with the new manager. In the following figure, it is shown how a direct connection is established with all other peers when a miner acts as a manager. In this scenario, a

peer can maximum connects to three other nodes. These are the previous node, the following node, and the manager.

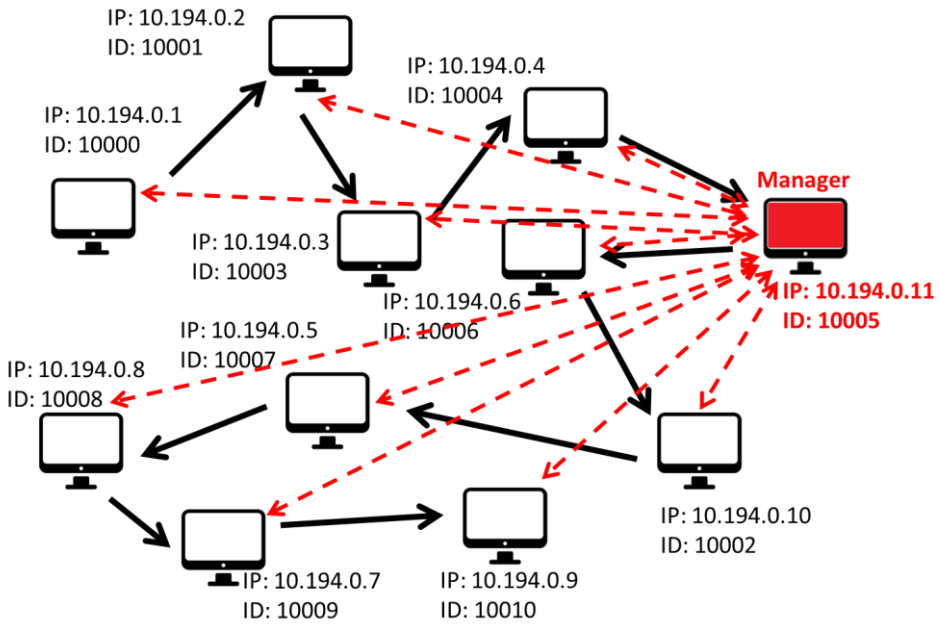

Figure 4.2 Peer to peer network diagram with a manager

4.2 Prototype Deployment

The prototype has been deployed in both local and in the cloud. To deploy in local, Docker has been used to create the P2P network. To deploy in the cloud, Google Cloud Platform (GCP) has been used for the same reason.

4.2.1 Local

Docker provides a Linux based container with its own network interface. Docker container is an open source application which is similar to a virtual machine. However, unlike virtual machines, Docker container installs the related application and dependencies to run a certain application. A Docker engine can have several

Docker containers, which can share the same application. Also, it is easy to distribute the resource to resources every container in the network.

To run an application, Docker engine creates a Docker file. In that file, there are three major components. These are the source code of the application, the dependencies of the application and the operating system. After creating a Docker file, it is deployed in a Docker engine.

For implementing our prototype, a Docker file has been created using the prototype source code. The source code is written using GoLang. After deploying the source code, the file has downloaded and installed all dependencies of the application. After completing the Docker file, it has been implemented into the Docker engine. Inside Docker engine, a network has been created with some Docker Containers. Every container has given different IP and different ID. Each container is considered as a different peer and they can communicate over the internet in the network.

Docker engine uses the processing power from the PC. It can be both dynamic and static. For our deployment, the processing power allocated to Docker was static. This resource will be distributed to all the container of the engine. Again, the resource allocated for each peer can be dynamic or static. If the dynamic resources are allocated, the resource provided to each peer will change based on the number of peers. Thus if a peer leaves the network or a new peer joins the network, the combined CPU resources remain the same. Thus, the allocated resource for each peer kept fixed.

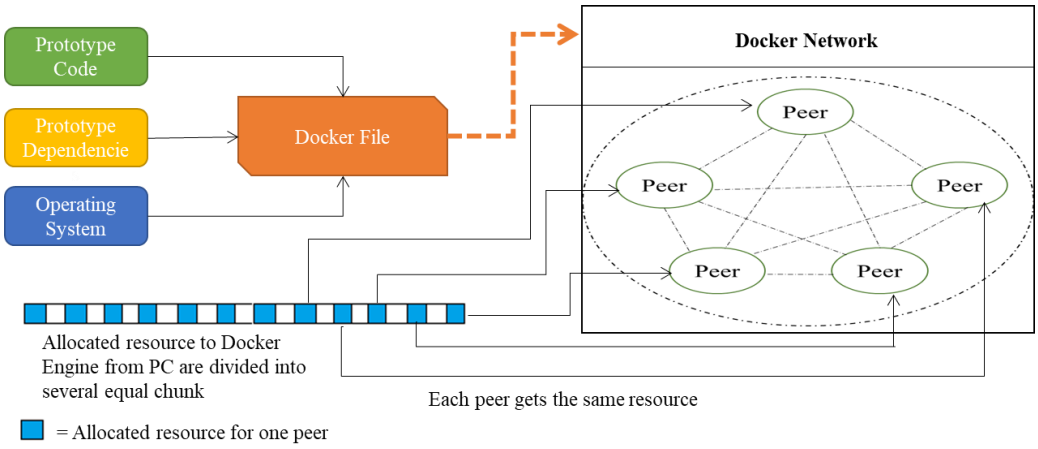

Figure 4.3 Prototype deployment for local resource

4.2.2 Cloud

The deployment in local brings some resource limitations. The Docker engine uses the resource from the local PC. Also, the allocated resources are distributed to peers. Thus every peer gets only a small amount of resource. The prototype cannot be evaluated with such a small amount of resources. Thus, the cloud platform needed to use to get a more powerful machine. Google Cloud Platform (GCP) is used to get the cloud environment.

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. GCP provides many services to the developers such as computing, hosting, big data, machine learning, networking, and storage.

To deploy our prototype, 32 compute engine has been used as a virtual machine. The physical locations of those virtual machines are in a different zone. A zone can contain maximum of 8 virtual machines for a single project. For our prototype

implementation, 6 zones have been chosen randomly and 32 virtual machines have been created in different zones.

Each virtual machine is considered as a peer. As the locations of these machines are different, the code and the required environment for the prototype have been built in each machine. Also, customized resource allocated to each peer, such as operating system, CPU type, number of CPU, RAM, storage. However, each peer has been allocated an identical resource.

No network needed to configure to communicate among the peer. Each virtual machine generated an IP while created. The peer could communicate over the internet using that IP. No firewall has been applied to any peer so that they can communicate without any blocking.

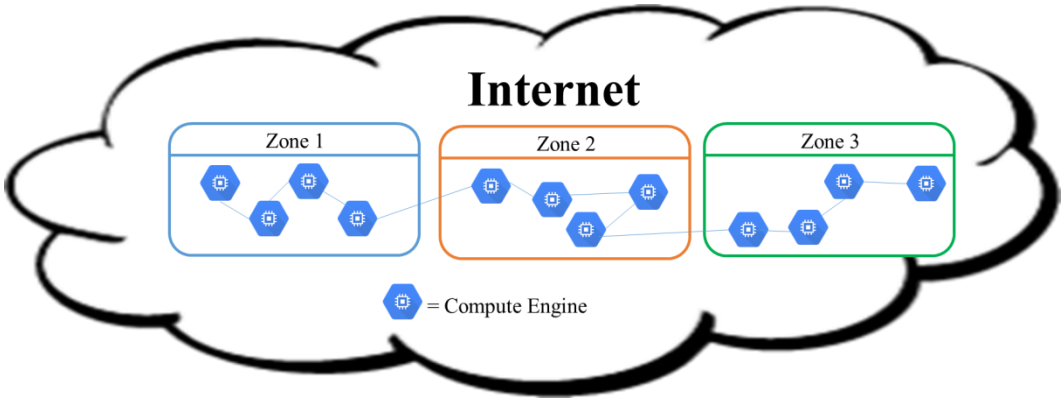

Figure 4.4 Prototype deployment for cloud resources

4.2 Block Data

The block contains the almost all data of the Bitcoin block such as transaction data, index, timestamp, solution of the previous block. Besides, it also contains the

corresponding manager id in a block. During solving a block, all data except transaction data and nonces are combined as a record and encrypted.

Algorithm 1. Block Solving Technique by the Miners

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 (Block index + Previous block hash + timestamp + transaction data + nonce)

3. Solve puzzle

for *i* = initial nonce value to *N* **do**

  if length(Blockchain)>new block.index **then**

    Block already solved

    Validate the Block solution

    Break

  Solution = SHA256(*Record* + *T* + *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

Here the transaction data is considered as a bunch of random data for each block and the nonce is taken in hexadecimal form. When the transaction data and nonce become available to a peer, s/he combines those with the record and encrypt again. The process continues with different nonce until the desired solution is found by any miners.

4.4 Block Validation

When a solution is found it validated by the other nodes. After validating, the block is created and the ledger is updated. To validate a block three parameters need to be considered. First, if the validated block index is the corresponding index of the current block; second, if the hash of a current block is the previous hash of validated block; third, if the hash of a validated block is less or equal to the target. If all of them are true, then the block is considered to be uploaded in the Blockchain. If any of them does not satisfy, the block considered as invalid.

Algorithm 2. Block Validation

```
if Previous Block Index+1 != New Block Index
        return false
else if Previous Block Hash != New Block Previous Hash
        return false
else if Hash(New Block) > target
        return false
else
        return true
end if
```

4.4 Genesis Block and Initial Manager

The genesis block is hard coded during implementing the model. It does not contain any transaction data, previous hash block information, and manager information. At the beginning of the model, all miner compete with each other to solve the null genesis block except the miner who connects to the network first. S/he is considered as the manager of the first block. When a miner solves the genesis block, the epoch of the next block begins and the methodology of the proposed model begins from here. It should be mentioned that there is no transaction fee reward for the genesis block. Thus the miner who solves the genesis block only gets the mining reward.

4.5 Coding Language

The environment for the proposed method has been developed using the Go programming language. The peer-to-peer network has been developed by using the GX library under lib-P2P of Golang [24]. This is a decentralized package manager that is used to distribute the same program to different nodes. The spew package of Golang is used to write the Blockchain ledger. Some of the basic concepts of Blockchain network is adapted from [25] to create the network. Rest is edited based on the requirement of the proposed model.

4.6 Encryption Technique, Target and Nonce

In order to perform the Proof of Work, an SHA-256 cryptographic hash algorithm has been used. SHA-256 is a secured hash function which always encrypts information to a 256-bit data independent of the input. During the implementation,

this algorithm is used to encrypt the transaction and find the solution. The reason SHA-256 is chosen because the same algorithm is used to perform the PoW of Bitcoin [26]. In our prototype, there are 5 types of data has been taken as input. These are transaction record, previous hash, timestamp, block index and nonce. Here, the nonce is an arbitrary hexadecimal number which can be used only once. The number can be generated using pseudo-random technique. . In the proposed method, except the nonce, all other data are same for every miner. The miner uses a lot of nonce one by one, until s/he finds the solution which is equal or less than target. Here, the target is a 256 bit number, which is represented by 32 double bytes hexadecimal number.

```
// SHA256 hashing
func calculateHash(block Block) string {
record := strconv.Itoa(block.Index) + strconv.Itoa(block.Transaction_record)
record = record + block.PrevHash + block.Nonce + block.Timestamp
h := sha256.New()
h.Write([]byte(record))
hashed := h.Sum( b: nil)
return hex.EncodeToString(hashed)
}
```

Figure 4.5 Code snippet for making a hash

4.7 Distributed Ledger

To write the distributed ledger, mutex library of GoLang has been used. After checking off each nonce, every peer checks the distributed ledger. If the ledger is found to be longer than the ledger available to him/her, s/he updates the ledger. Again, after solving the Block, the peers maintain the same logic before broadcasting his/her solution to check if the Block is already solved and the ledger is updated.

```
mutex.Lock()
if len(chain) > len(Blockchain) {
    Blockchain = chain
    bytes, err := json.MarshalIndent(Blockchain, prefix: "", indent: "  ")
    if err != nil {
        log.Fatal(err)
    }
    fmt.Printf( format: "\x1b[32m%s\x1b[0m> ", string(bytes))
}
mutex.Unlock()
```

Figure 4.6 Code snippet to update the distributed ledger

4.8 Challenges and Solutions

The proposed model of parallel mining is different than solo mining in a different aspect. Such as communication, reward system and distribution of data. To build the prototype and evaluation of the prototype, many challenges were faced. The significant challenges with solutions are discussed following.

- **Connectivity with the Manager:** All nodes are connected in a P2P network directly or indirectly. The communication between the indirect nodes takes more time than the directly connected nodes. When all nodes need to collect data from a single peer (manager), this type of connection may create an issue. Because the directly connected node will get the data (transaction data) sooner than the node indirectly connected with the manager. Thus, it may seem that the directly connected node will get more advantage to solve the puzzle than the other nodes. If it happens repeatedly, only the nodes in a specific zone will get to be the manager. This will create centralization in the network. This issue was solved by deploying a temporary direct connection with the manager to all other peers. The connectivity period would last only for a certain Block period. As there is no indirect communication with the

manager, the delay reduced to send the data. However, the temporary communication ends when the Block is solved. Than a new temporary connection created with the new manager.

- **Malicious Peer:** The data is broadcasted in a P2P network using the Gossip protocol [38]. In the Blockchain network of cryptocurrencies, this technique is used to send the message when a Block is solved by a miner. After that, the peers update their ledgers. This technique can be used to update the ledger, but cannot be used to send data (Transaction data and nonces) in parallel Proof of Work. Because a malicious miner can modify or delete the transaction hash which can destroy the whole process. This issue was also solved when a direct connection is deployed with the manager to all other peers.

- **Resource Distribution:** The prototype was first implemented in Local PC using an internal network. The application was used by different miner using a different terminal. Each terminal acted as a peer. However, this deployment was not enough to evaluate the proposed model. First, the evaluation of the parallel Proof of Work should be done based on the number of the peer. If we consider that the resource for all peers is the same and fixed, the scalability of the process will increase with increasing the number of peers. However, it did not happen when it is deployed in this way. Because the resource is allocated to every peer was dynamic in this way. If the number of peers increased, the resource for each peer decreased. As a result, the combined resources for all peers remain the same. To solve this issue, a Docker container has been used. Using Docker container, each peer still gets the resource from local PC, but this time the resource allocated to each peer was fixed

and equal. In this case, if the number of peers increased, the combined processing power for all resources also increased. Again, through Docker network, the peer connected with each other using the internet. Thus the network latency, bandwidth, throughput also affected the evaluation result which was not possible to do using the internal network.

- **Limited Resources:** The limitation for using Docker network was about the limited resources. For local deployment, the peers got the resources from a local PC which is very small compared to any cryptocurrency network. Again, the resources were distributed to all peers. Thus, the difficulty level and the number of a peer could not be increased much. To solve this issue, the prototype has deployed in cloud platform using Google Cloud Platform. Thus every peer got the same configuration of a standard local PC. This helped to increase the number of peers as well as the level of difficulties to evaluate the prototype.

4.9 Summary

This chapter has presented the implementation of a prototype based on the proposed design from the previous section. This chapter described the implementation technique, as well as the new changes in parallel proof of mining with solo mining. The chapter also discussed the challenges faced when the prototype was deployed both in local and in the cloud. The next chapter will discuss the evaluation result of the prototype implementation.

Chapter 5

Evaluation and Results

In this chapter, we analyze the extent that this thesis is successful in achieving the objectives defined at the beginning of this paper. The result section can be divided into four major parts. These are parallel mining in local, solo mining in local, parallel mining in cloud and solo mining in cloud. Both solo mining and parallel mining has been implemented in the same network setup to understand the improvement of the proposed model.

5.1 Local Experimental Environment

It is very important to distribute the resources equally to all peers to evaluate the solution. To implement this approach, a Docker container has been used. 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 the 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 in solo. They compete with each other, as in the existing system, and a successful miner receives all the reward.

5.1.1 Resources

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. The difficulty levels are considered here are 5, 6 & 7. That means the consecutive leading zeros are 5, 6 and 7 for these difficulty levels respectively. Difficulty levels 1, 2, 3 and 4 are not considered because for those levels of difficulties, the Block solving time improvement is not significant when we increased the number of peers. Also, Block solving for such small difficulties are very short. This is not good enough to evaluate the result. The numbers of peer considered are 1, 3 and 5. Here 1 peer means only one miner is solving the Block and another miner is acting as the manager. That is, only two miners are there in the network. For 3 there is one manager and three peers are working in parallel. The same thing happens for the 5 peers.

Similar peer number and difficulty level are considered for both solo mining and parallel mining. To identify the solution, the index, timestamp, transaction hash, previous hash, and the nonce are taken as input. Here, for 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.

5.1.2 Results

Following figures represent the test result based on solo and parallel mining. Here, the Average Time(s) means the average time required to solve a block in seconds. The number of peer represents how many miners are considered in the

network (Except manager in the parallel mining). The difficulty level defines the target for which the prototype has been evaluated. The result is showing are the averages, which were conducted for several times under the same scenario.

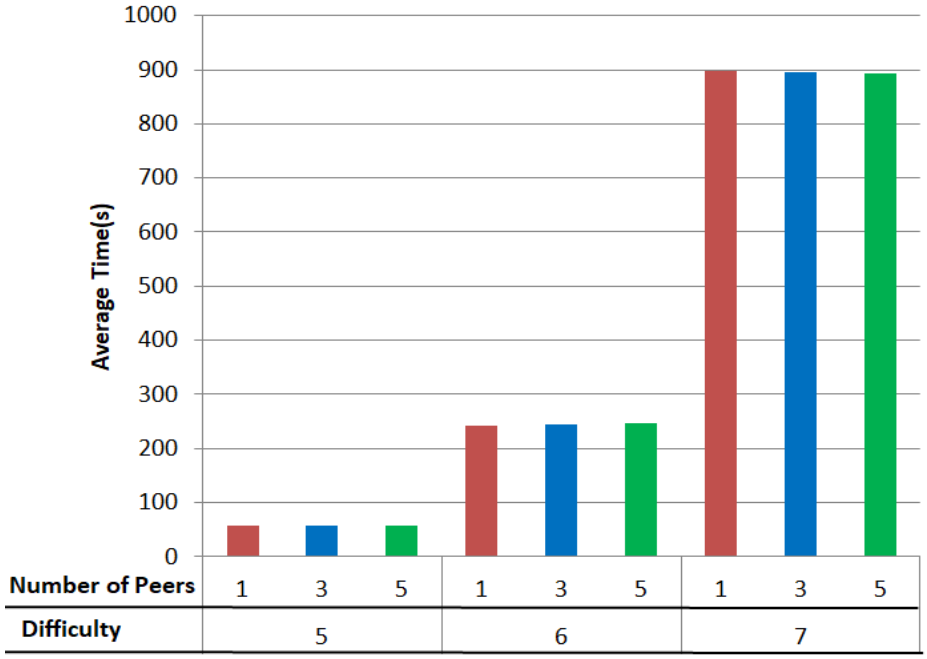

Figure 5.1 Test result for solo mining

In solo mining, the Block solving time increase exponentially with the increase of difficulty regardless of the number of peers. The average Block solving time remains the same for every number of peers.

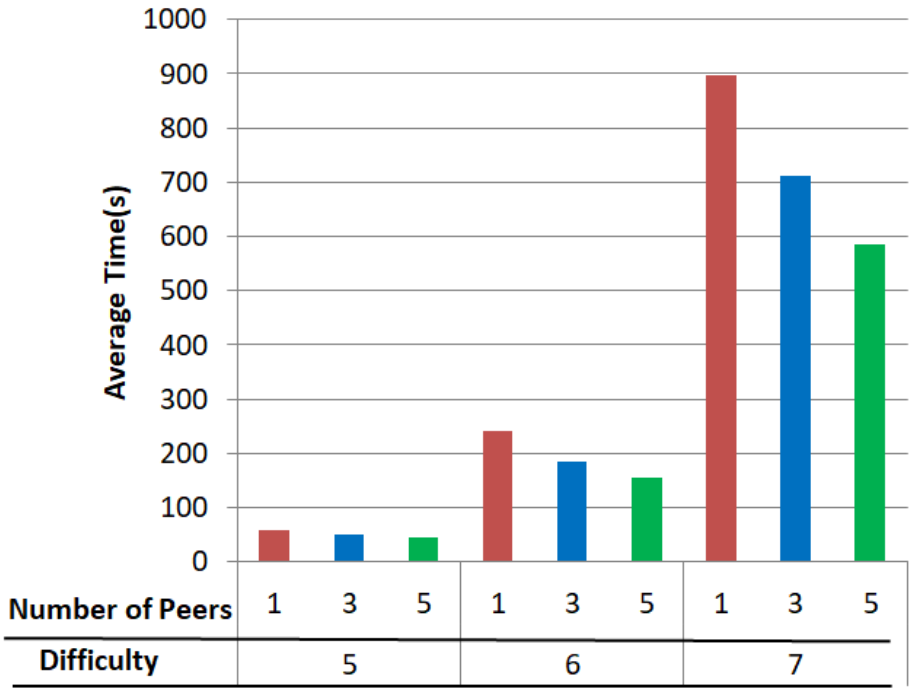

Figure 5.2 Test result for parallel mining

In parallel mining, the average Block solving time increases with the increase of difficulty. Again, if the number of peers increases, the average Block solving time decreases. The difference seems significant with the increase in difficulty. For example, for difficulty 5, the difference between average Block solving time for 3 peers and 5 peers is a few seconds. In case of difficulty 7, the difference reaches around two minutes.

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 for the same level of difficulty. This is because, when there is only one miner in parallel mining, no parallel work is taking place. The scenario is exactly the same as the solo mining. It

should be noted that the results will vary based on the processing power allocated to the miners in the same scenario for the same number of peers.

5.2 Cloud Experimental Environment

To evaluate the prototype, a total of 32 virtual machines with equal resources has been set up to do the experiment. Each machine contains 6.25 GB of memory with 1 virtual CPU. The operating system of every machine is Ubuntu 16.04 LTS with 10GB of allocated hard disc. The CPU platform is equivalent to Intel Skylake. No GPU is provided to any of the machines. The physical location of each virtual machine is in a different region with a different zone. Thus the network IP is also from a different group.

5.2.1 Resources

To do the experiment, different types of difficulty level have been chosen. The targets for different difficulties 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 next of the paper. It is similar to floating point notation. The target is represented in 32 double bytes number. For example, the difficulty 0x1d00ffff, the target number is 0x00000000ffff0000000000000000000000000000000000000000000000000000.
The decimal value of 1d is 29. Thus, the first 3 double bytes of the target should be 000000. We have to calculate the rest 29 double bytes. The next 3 double bytes should be 00ffff for this target. For the rest 26 double bytes, the value should be 0. Thus we get a full 32 double bytes number as target. The solution hash should be less or equal to that target.

The test has also been done for different number of peer in parallel mining. The numbers of the peer are 1, 7, 13, 21, 27 and 31. In every experiment for parallel mining, there is always one extra miner in every epoch who acts as the manager. To compare the result in solo mining similar environment has been created where every peer has the same resource configuration as the parallel mining. In solo mining the same difficulty level has been used.

5.2.2 Results

For parallel mining, the test has been done in different difficulties for the different peer. Following figures represents the time required to solve any 15 consecutive Block by a different number of peer in 6D and 10D difficulties respectively. For 6D difficulties, the bock solving time differences are not that significant compared to a different number of peers. For example, the highest time required to solve 1 block was around 13.5 min when the test has been done for 1 peer. Again, the highest time required to solve 1 block was around 14.25 min for 19 peers. If the lowest block solving time is considered, the time required to solve 1 block was almost the same for 1 peer, 7 peers, and 19 peers. However, when the average time is considered for a different number of a peer, a small but significant result has been found. It seems that when the number of peers increase the average block solving time decrease. The average block solving time for 1, 7, 13, 19, 25 and 31 peers were 7.92, 8.41, 8.79, 9.39 and 9.72 minutes respectively. With the increase of 6 peers, the block solving time decreased to 0.45 minutes on average.

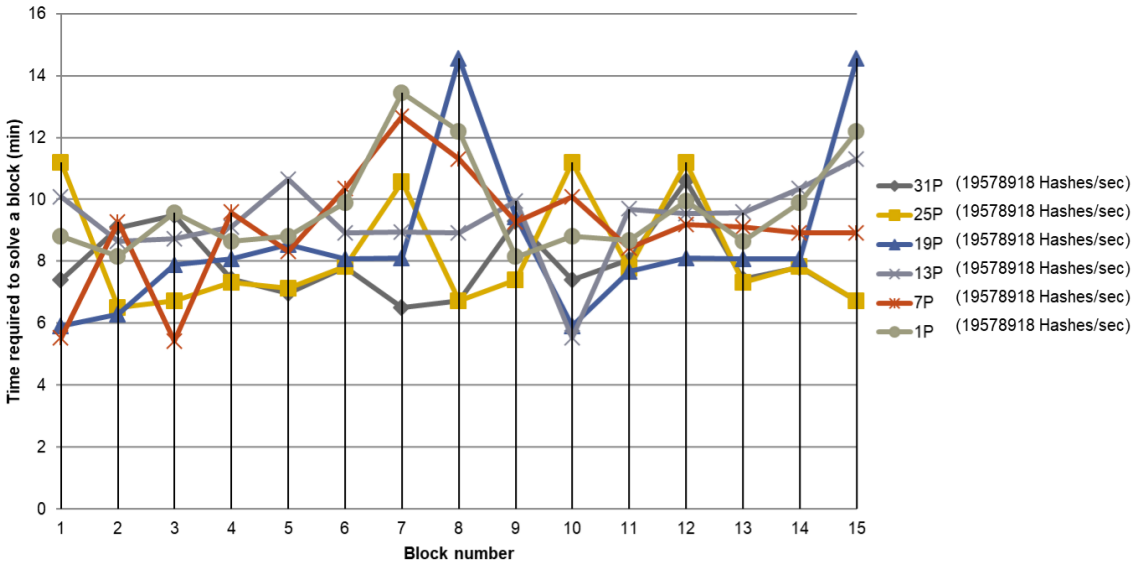

Figure 5.3 Time required to solve any 15 consecutive Block by varying number of peers in 6D difficulties

The test then conducted for 7D, 8D, 9D, and 10D. With the increase of difficulty level, the time block creation is also increased. It was obvious because the allocated resources were still the same. However, with the increase of difficulty block solving time difference increased. For example, the average block solving time difference for 7D was 0.55 minutes. It was 1.33 minutes for the 8D level of difficulty. The average time differences were 2.65 minutes and 5.51 minutes for 9D and 10D respectively with the increase of a number of the peer. In every case, this is the best case scenario while 31 peers were working in parallel having one manager as another peer. The worst case scenario was for 1 peer. Here, the average Block solving time. The following figure represents the time required to solve any 15 consecutive Block by a different number of peer in 10D difficulties. It shows more significant result compared to the 6D difficulties. The Block solving time difference between 1 peer

and 31 peers parallel working environment was significantly more compared to the result in 6D difficulties. For example, the highest and lowest time required to solve a block in case of 1 peer was around 75 minutes and 50 minutes respectively. For 31 numbers of peers, those were 40 minutes and 31 minutes respectively. If we consider the lowest Block solving time happened for a Block, it is 27 minutes. This was done in the 25P scenario. Though, the average Block solving time for 25P for 1 Block is 40.03 minutes. If the average times are considered, those are 58.84 minutes, 53.63 minutes, 48.48 minutes, 43.36 minutes, 40.03 minutes and 36.78 minutes for 1 peer, 7 peers, 13 peers, 19 peers, 25 peers, and 31 peers respectively with average 5.51 minutes time difference to create 1 block.

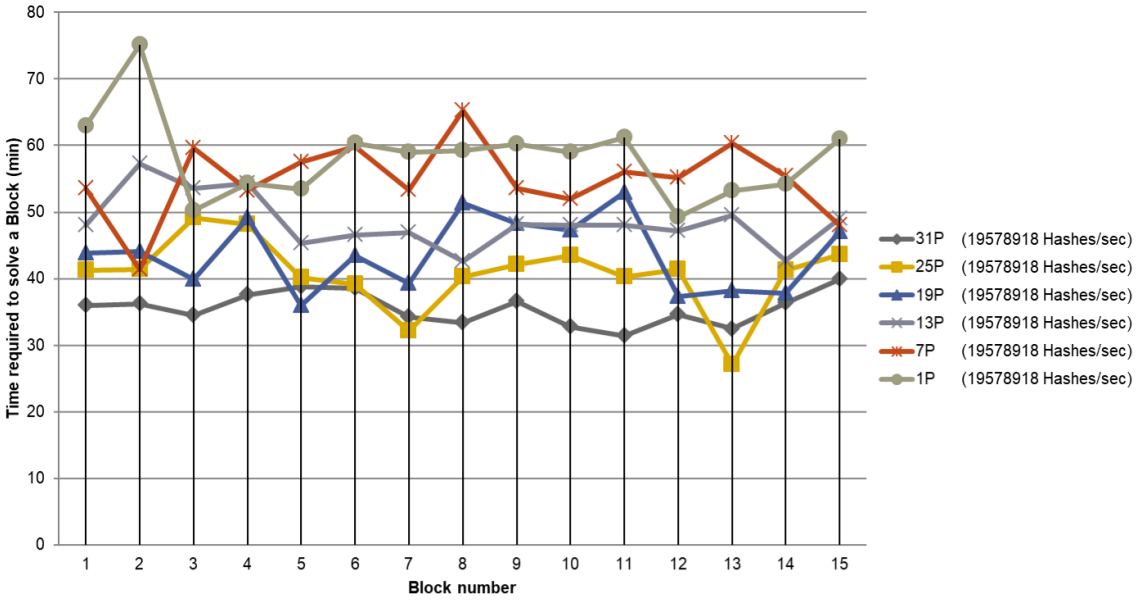

Figure 5.4 Time required to solve any 15 consecutive Block by varying number of peers in 10D difficulties

The following figure represents the average time required to solve a Block in different difficulty level with a different number of the peer. The average required

Block solving time decreased with the increase of a number of peer in the same difficulty and increased with the increase of difficulty for the same number of the peer. Though, the rate of change of Block solving time differs for a different level of difficulty. Also, the improvement differs for a different level of difficulties. The difference between 1 peer and 31 peers scenario is 1.8 minutes for 6D difficulty. When we consider 7D difficulty the difference reaches to 2.21 minutes. This value changes to 5.35 minutes, 10.58 minutes and 22.07 minutes for 8D, 9D and 10D level of difficulty scenario respectively.

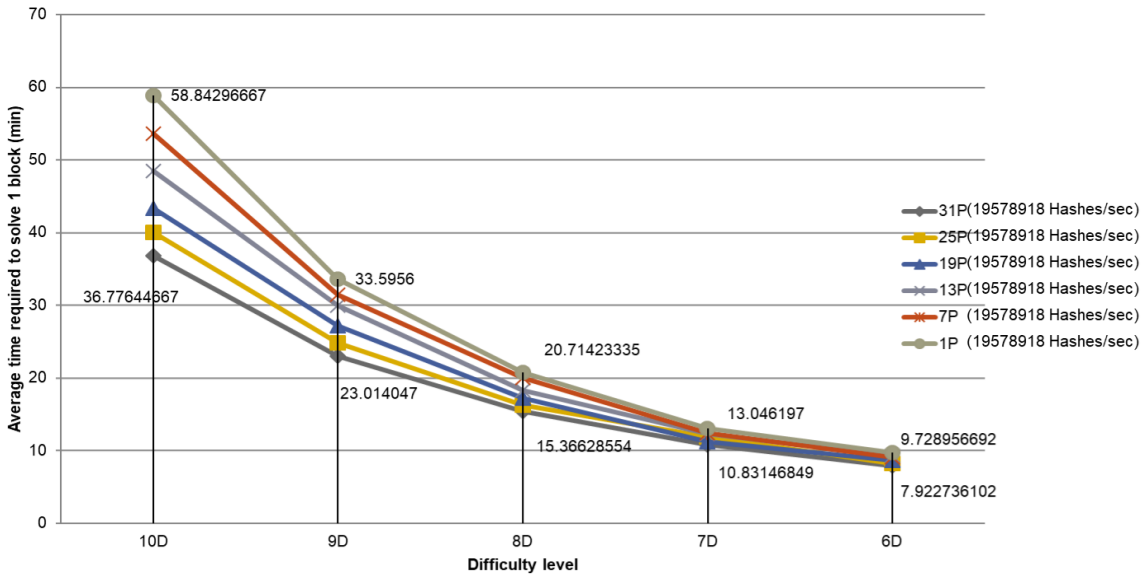

Figure 5.5 Average time required to solve a Block by varying number of peers in different difficulty levels

From the evaluation, we can observe that, the average Block solving time increase exponentially with the increase of difficulty. However, in the case of parallel mining, the improvement is rational for the same level of difficulty with an increased number of peers. In the case of 6D difficulty, the improvement is around 0.36 minutes with

an increase of every 6 peers. This improvement reaches to 2.11 minutes and 4.41 minutes for the 8D difficulty and 10D difficulties respectively. Thus, we also can say that the proposed parallel Proof of Work is more efficient with a large level of difficulties.

5.3 Solo Mining

To compare parallel mining with traditional solo mining, a similar environment has been created. The experiment has been done for different difficulty levels with a different number of peers. It seems that the block creation does not depend on the number of the peer. It depends only on the level of difficulties. Figure 5.8 the solo mining for different difficulties for any 15 consecutive Blocks. The highest and lowest time required to solve 1 Block in 10D difficulties was around 75 and 49 minutes respectively. For the 6D difficulties, it is around 12 minutes and 9 minutes respectively. The average time increase with the increase in difficulty level.

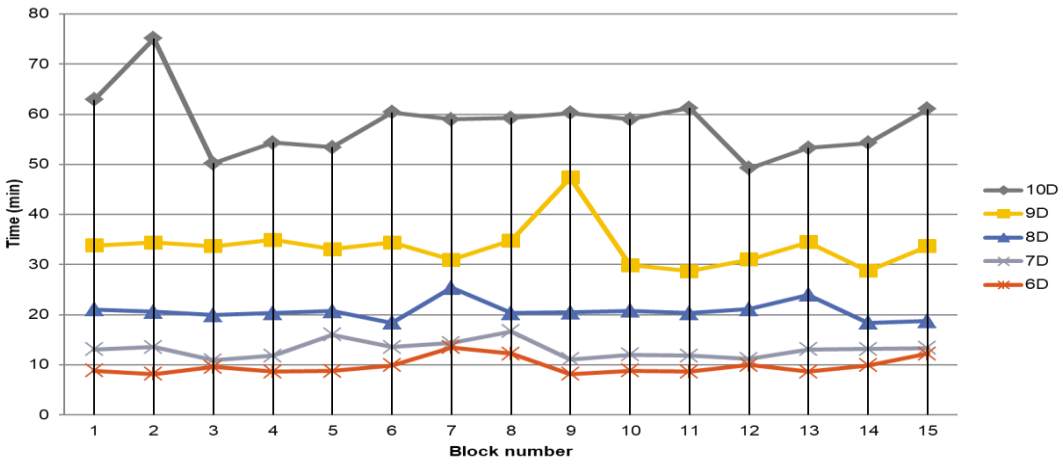

Figure 5.6 Time required to solve any 15 consecutive Block in different difficulty levels in solo mining

When we compare the solo mining average block creation time with respect to parallel mining for 31 peers, we find a similar result for the difference between 1 peer and 31 peers in parallel mining. The peer number does not affect the Block creation time in solo mining. Only the difficulty levels are considered here. Here, for 6D difficulties, the time difference is not that significant in 6D difficulties compared to the time difference in 10D difficulties. Thus, we can again conclude that the proposed algorithm is more efficient by increasing the number of difficulties.

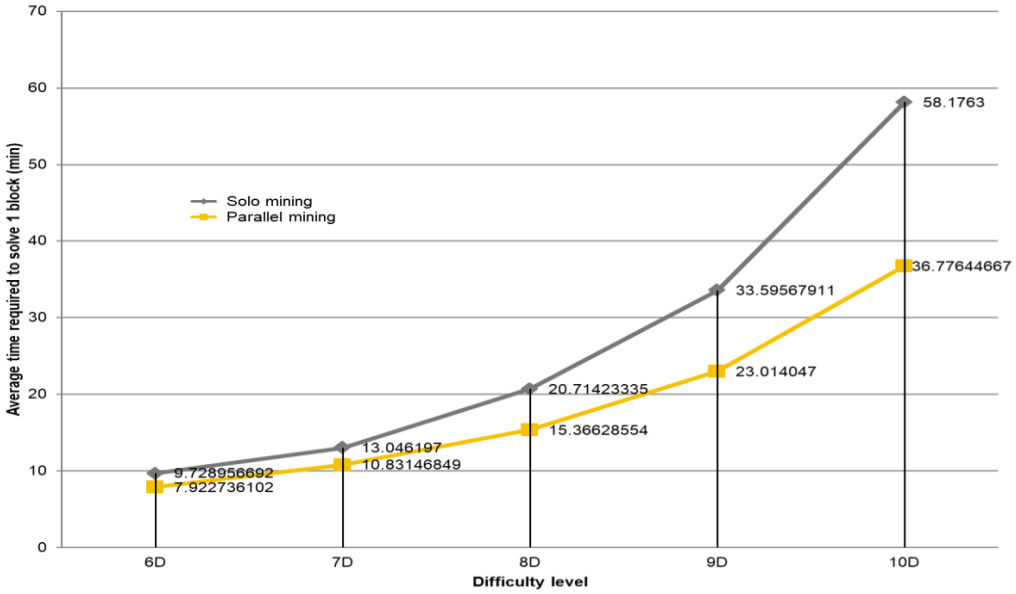

Figure 5.7 Solo mining vs. parallel mining

It is noticeable that the solo mining evaluation result and the parallel mining evaluation result for 1 peer (With a manager) is almost identical. Because parallel mining system with only one peer refers to solo mining. No parallel work has been done in this case. A similar type of result also observed in local deployment. In that case, the Block solving speed was similar for solo mining and parallel mining for 1

peer (With a manager). In other scenarios, the Block solving time differs because the Block solving speed increase with the increase of peers in parallel Proof of Work.

5.4 Transaction Speed

The transaction speed of a cryptocurrency depends on the block solving time. Currently, the transaction speed of Bitcoin is approximately 3-7 transaction per second. The size of each block is maximum 1 MB. On the other hand, the size of every transaction is 250 to 450 bytes. Thus a block can contain approximately 2275 to 4096 numbers of transactions. Again, the average block solving time is 10 minutes. Based on this data, the transaction speed of Bitcoin is assumed as 3-7 transaction per second. As our prototype is based on the attribute of Bitcoin, we also found similar type of result. The following figure shows the transaction speed for different scenario in our prototype.

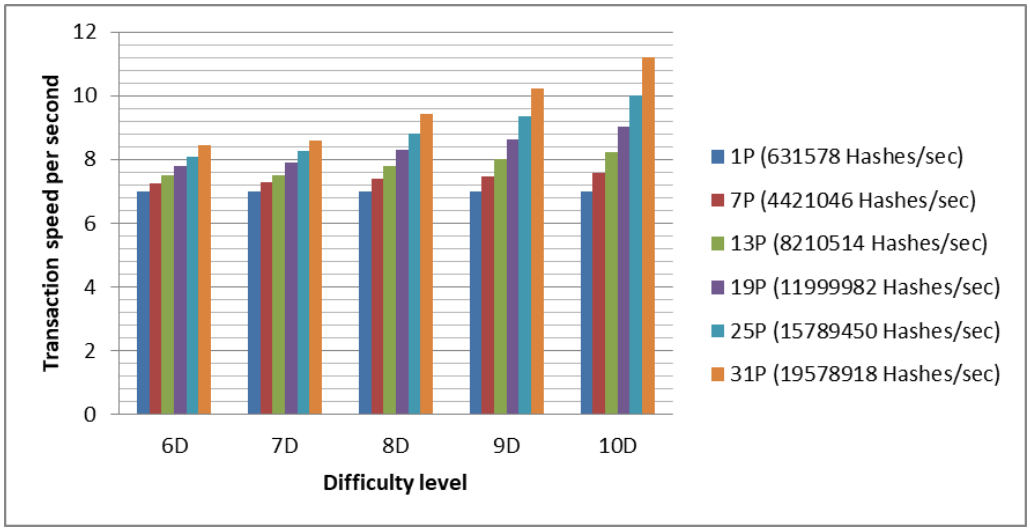

Figure 5.8 Transaction speed in different scenario based on the prototype

The comparison has been done for 5 different difficulties and for 6 combinations of peers. For solo mining, the transaction speed is assumed as 7 transactions per

second. For the 6D difficulties, the transaction speed difference is less. Here, the lower transaction speed is 7 transactions per second for 2 peers or solo mining scenario. For 32 numbers of peers scenario, the speed become almost 8.42 transactions per second. However, the speed changes with the increase of difficulty levels. For 10D level of difficulty, the lower transaction speed is 7 transactions per second. Again, for 32 number of difficulty the speed become more than 11 transactions per second. That means, the improvement is more significant in higher level of difficulty. The improvement is more than 4 transactions per second in 10D level of difficulty. It was less than 1.5 transactions per second for 6D level of difficulties. It was obvious as the block solving was also significant in higher level of difficulty (10D) than the lower level of difficulty (6D). However, the prototype results depend on the number of peers, the levels of difficulties and the combined has power of the peers. The result may change if any of the mentioned attributes changes.

5.5 Summary

The primary focus of this thesis is to develop a scalable consensus mechanism. In order to do so, the proposed algorithm has been implemented both in local and cloud. In both cases, a significant improvement has been noticed. From the evaluation, it has been found that the improvement of the proposed model depends on the number of peer and the level of difficulty. It also depends on the resource allocated to each peer. Thus, if the prototype is implemented in different environment the improvement rate may seem different.

Chapter 6

Conclusion and Future Work

Increasing scalability in a decentralized system is challenging compared to a centralized system. As the Bitcoin and other related cryptocurrency are becoming popular day by day, scalability is becoming a major drawback to increasing the involvement of these cryptocurrencies in the financial sector. Though most of the cryptocurrencies are decentralized and secure, these also need to scale to compete with the traditional currency transaction system. It is obvious that, in order to scale a permissionless Proof of Work based Blockchain system, the structure of the PoW algorithm needs to be modified. The solutions like Block size increase or network improvement will bring short term solution. In order to get a long term solution, the core attributions need to be customized. The proposed model introduced the approach of parallel Proof of Work in which all miners can together solve the puzzle by taking part in the competition. In this model, the miners compete and cooperate with each other at the same time to solve a particular Block. In a traditional Proof of Work mechanism, the effort of all peers except the winner becomes worthless. Here, there will be no worthless contribution from any miners, though only one miner will get the reward. The model has been approached after the evolution of available scalable solutions and the goal was to design a long term solution. Many researchers have settled that it is not possible to get the absolute improvement in three major factors of a peer to peer system, the scalability, security, and decentralization. The

model was designed to maintain all of these three aspects as much as possible. According to the evaluation result, the proposed model will be able to decrease the scalable challenges of these cryptocurrencies.

The model can be improved with the following future works.

- Deploying the solution in Bitcoin testnet to compare the scalability with Bitcoin in the real-life network.
- The different dynamic factor of the network such as network latency, throughput, and network bandwidth are not considered during the evaluation.
- The energy consumptions need to be evaluated.

In conclusion, the thesis includes four major portions. The evaluation and comparison of the existing scalable solution have been done to approach a new method to overcome the current gaps. This is discussed in chapter 2. Chapter 3 discussed the architecture breakdown of proposed parallel Proof of Work with its incentives and use case scenario. The implementation technique with the challenges and solutions are explained in chapter 4. Lastly, in chapter 5, the evaluation result analysis and comparative analysis with the current method has described. The proposed model brought a lot of scopes do more evaluation which has been pointed in future work. However, the evaluations done for the thesis established a significant improvement based on the prototype implementation of parallel Proof of Work.

Bibliography

[1] Flesher, Dale L. "BARTER BOOKKEEPING: A TENACIOUS SYSTEM". *Accounting Historians Journal*, vol 6, no. 1, pp. 83-86, 1979. American Accounting Association, doi:10.2308/0148-4184.6.1.83.

[2] Fraser, Ian A. M. "Triple-Entry Bookkeeping: A Critique". *Accounting And Business Research,* vol 23, no. 90, pp. 151-158, 1993. doi:10.1080/00014788.1993.9729872.

[3] "Who Invented Money?", *Wonderopolis.Org,* 2018, https://wonderopolis.org/wonder/who-invented-money. [Accessed 7-Oct-2018].

[4] Duke, David. "The Peer-To-Peer Threat". *Network Security,* vol 2002, no. 12, p. 4, 2002. Elsevier BV, doi:10.1016/s1353-4858(02)12007-1.

[5] J. Tung and V. Nambudiri, "Beyond Bitcoin: potential applications of blockchain technology in dermatology", *British Journal of Dermatology*, vol. 179, no. 4, pp. 1013-1014, 2018. Available: 10.1111/bjd.16922.

[6] Scherer, Mattias. “Performance And Scalability Of Blockchain Networks And Smart Contracts.” *Umea University,* 2017, http://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf. [Accessed: 15-May-2018].

[7] Clark, Joseph B., Andrew M., Arvind N., Joshua A., Kroll E., Felten W., "Research perspectives and challenges for bitcoin and cryptocurrencies." *Security and Privacy (SP), IEEE Symposium on IEEE,* pp. 104-121, 2015.

[8] "Understanding Cryptocurrency Transaction Speeds – Coinmonks – Medium", *Medium*, 2018. [Online]. Available: https://medium.com/coinmonks/understanding-cryptocurrency-transaction-speeds-f9731fd93cb3. [Accessed: 12- Jul- 2018].

[9] C. Malmo, “Bitcoin Is Unsustainable”, *Motherboard*, 2015. [Online] Available: https://motherboard.vice.com/en_us/article/ae3p7e/Bitcoin-is-unsustainable. [Accessed: 21-May-2018].

[10] Nakamoto, S. "Bitcoin: A peer-to-peer electronic cash system". *Bitcoin.org*, 2009, Available: https://bitcoin.org/bitcoin.pdf.

[11] "Cryptocurrency Market Capitalizations, Coinmarketcap". *Coinmarketcap*, 2018, https://coinmarketcap.com. [Accessed: 7-Dec-2018].

[12] Zheng Z., C., Xie, D. "An overview of blockchain technology: Architecture, consensus, and future trends." *Big Data (BigData Congress), 2017 IEEE International Congress on IEEE*, pp. 557-564, 2017.

[13] Lamport L., Robert S., Marshall P. "The Byzantine generals problem." *ACM Transactions on Programming Languages and Systems (TOPLAS)* vol. 4.3, pp. 382-401, 1982.

[14] Nguyen, Giang-Truong, Kyungbaek K.. "A Survey about Consensus Algorithms Used in Blockchain." *Journal of Information Processing Systems*, vol. 14, No.1, pp.101-128, February 2018.

[15] P. De, V. Jacob and R. Pakath, "A Formal Approach for Designing Distributed Expert Problem-Solving Systems", *Information Systems Research*, vol. 4, no. 2, pp. 141-165, 1993. Available: 10.1287/isre.4.2.141.

[16] Dwork C., Nancy L., and Larry S. "*Consensus in the presence of partial synchrony." Journal of the ACM (JACM)*, vol. 35.2, pp. 288-323, 1988.

[17] I. Eyal, A. Gencer, E. Sirer, Renesse R.. "Bitcoin-NG: A Scalable Blockchain Protocol.", *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16*), pp. 45-59, March 16–18, 2016.

[18] Boyen, Xavier, Christopher C., Thomas H. "Blockchain-Free Cryptocurrencies. A Rational Framework for Truly Decentralised Fast Transactions." *IACR Cryptology ePrint Archive, p. 871*, 2016.

[19] Florea, Bogdan C. "Blockchain and Internet of Things data provider for smart applications." *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1-4, IEEE, 2018.

[20] King, S, Nadal, S., "PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake", *White Paper*. Amsterdam, The Netherlands: Stichting Peercoin Foundation; https://decred.org/research/king2012.pdf. 2012. Accessed October 29, 2018.

[21] "How Do Mining Pools Work? Is It Better Than Solo Mining? | Captainaltcoin". *Captainaltcoin*, 2018, [Online], Available: https://captainaltcoin.com/what-is-pool-mining. [Accessed: 20-Oct-2018].

[22] Jordan T., "10 Best and Biggest Bitcoin Mining Pools 2019 (Comparison)", *Buybitcoinworldwide.com*, 2019. [Online]. Available: https://www.buybitcoinworldwide.com/mining/pools/. [Accessed: 27- Jan- 2018].

[23] Schollmeier R. "A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications." *Proceedings First International Conference on Peer-to-Peer Computing.* IEEE, pp. 101-102, 2001.

[24] "The pioneer's guide to GX — decentralized dependency management on IPFS", *Hacker Noon*, 2019. [Online]. Available: https://hackernoon.com/the-pioneers-guide-to-gx-decentralized-dependency-management-on-ipfs-90064858f4c2. [Accessed: 06- Aug- 2018].

[25] "Code a simple P2P blockchain in Go! – Coral Health – Medium", *Medium*, 2018. [Online]. Available: https://medium.com/@mycoralhealth/code-a-simple-p2p-blockchain-in-go-46662601f417. [Accessed: 15- Feb- 2018].

[26] Courtois, Nicolas T., Marek G., Rahul N. "Optimizing sha256 in bitcoin mining." *International Conference on Cryptography and Security Systems.* Springer, Berlin, Heidelberg, pp. 131-141, 2014.

[27] Chaum D. "Blind signatures for untraceable payments." *Advances in cryptology.* pp. 199-203, Springer, Boston, MA, 1983.

[28] Lunt P. "E-cash becomes reality, via Mark Twain and Digicash." *American Bankers Association,* p. 62, ABA Banking Journal 88.1,1996.

[29] Dai, Wei. "B-money proposal." *White Paper*, 1998.

[30] Zhang H., "Simultaneous multispectral coded excitation using Gold codes for photoacoustic imaging." *Japanese Journal of Applied Physics 51.7S (2012): 07GF03*.

[31] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, V. Kalyanaraman, "Blockchain Technology", *Sutardja Center for Renessereneurship & Technology Technical Report,* pp. 6-10, 2015.

[32] Castro M., Barbara L. "Practical Byzantine Fault Tolerance And Proactive Recovery". *ACM Transactions On Computer Systems*, vol 20, no. 4, pp. 398-461, 2002.

[33] P4Titan. “Slimcoin A Peer-to-Peer Crypto-Currency with Proof-of-Burn”, Available:https://www.chainwhy.com/upload/default/20180703/4ae7cee40462e7951f508b28dd1d9936.pdf, May 17, 2014.

[34] Larsson T., Thorsén. “Cryptocurrency Performance Analysis of Burstcoin Mining.”, *University West, Department of Engineering Science*, pp. 21, 2018.

[35] Bentov I., Lee C., Mizrahi A., Rosenfeld M., "Proof of activity: Extending bitcoin's Proof of Work via Proof of Stake." *ACM SIGMETRICS Performance Evaluation Review,* vol. 42.3, pp. 34-37, 2014.

[36] Sankar, Lakshmi S., Sindhu M., Sethumadhavan M., "Survey of consensus protocols on blockchain applications." *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS),* pp. 1-5, IEEE, 2017.

[37] "Digiconomist - Cryptocurrency Fraud and Risk Mitigation", *Digiconomist*, 2018. [Online]. Available: https://digiconomist.net/. [Accessed: 28- Nov- 2018].

[38] Leitao J., José P., Luis R. "HyParView: A membership protocol for reliable gossip-based broadcast." *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN'07),* pp. 419-437, IEEE, 2007.

[39] Gencer, Adem E. Robbert R., Emin S. "Short paper: Service-oriented sharding for blockchains." *International Conference on Financial Cryptography and Data Security*, pp. 393-401, Springer, Cham, 2017.

[40] Gao Y., Hajime N. "A Proof of Stake Sharding Protocol for Scalable Blockchains." *Proceedings of the Asia-Pacific Advanced Network,* pp. 13-16, 2017.

[41] Lewenberg, Yoad, Yonatan S., Aviv Z. "Inclusive block chain protocols." *International Conference on Financial Cryptography and Data Security,* pp. 528-547, Springer, Berlin, Heidelberg, 2015.

[42] De R., Gerard, Ikram U., Paul H., "How to break IOTA heart by replaying" *2018 IEEE global Communications conference, GLOBECOM 2018: Gateway to a Connected World,* 2018.

[43] Herrera J., Jordi, and Cristina P., "Privacy in bitcoin transactions: new challenges from blockchain scalability solutions." *International Conference on Modeling Decisions for Artificial Intelligence.* pp. 26-44, Springer, Cham, 2016.

[44] McCorry P., "Towards bitcoin payment networks." *Australasian Conference on Information Security and Privacy*, pp. 57-76, Springer, Cham, 2016.

[45] Nguyen, Giang T., Kyungbaek K. "A Survey about Consensus Algorithms Used in Blockchain." *Journal of Information Processing Systems,* Vol.14, No.12018, pp.101-128, 2018.

[46] Bach, L. M., Mihaljevic B., Zagar M., "Comparative analysis of blockchain consensus algorithms." *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO).* pp. 1545-1550, IEEE, 2018.

[47] Chalaemwongwan N., Werasak K., "State of the art and challenges facing consensus protocols on blockchain." *International Conference On Information Networking (ICOIN),* pp. 957-962, IEEE, 2018.

[48] Lewenberg, Y., Bachrach, Y., Sompolinsky, Y., Zohar, A., Rosenschein, J. S. "Bitcoin mining pools: A cooperative game theoretic analysis". *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems* . International Foundation for Autonomous Agents and Multiagent Systems. pp. 919-927, May 2015.

[49] Hazari, S. S., Mahmoud, Q. H. "A Parallel Proof of Work to Improve Transaction Speed and Scalability in Blockchain Systems". *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC,)* pp. 916-921. IEEE. January, 2019.

[50] Hazari, S. S., Mahmoud, Q. H., "Comparative Evaluation of Consensus Mechanisms in Cryptocurrencies". *Internet Technology Letters*, doi: 10.1002/itl2.100, 2019.

Appendices

Appendix A: Selected Source Code

Two code snippets have been provided here. A.1 shows the technique to establish a P2P network, maintain the consensus and update the distributed ledger. A.2 shows the manager and other peers relation.

A.1 P2P Network

```
func main(){
   t:=null
   gBlock:=Block{}
   gBlock=Block{0,t.String(),0,Hash(gBlock),"", difficulty,"",
"",null}

   Bchain=append(Bchain, gBlock)
   golog.SetAllLoggers(gologging.INFO)
   waiting:=flag.Int("l",0,"new peer can connect")
   dest:=flag.String("p","","next node")
   section:=flag.Bool("section",true,"secio check")
   peer:=flag.Int("peer",0,"introduce identity")
   flag.Parse()
   ja, error:=makeHost(*waiting,*section,*peer)

   if *dest==""{
      log.Println("waiting for new peer")
      ja.SetStreamHandler("/p2p/1.0.0", handleStream)

      select {}
   } else{
      ja.SetStreamHandler("/p2p/1.0.0", handleStream)
      address:=ka.NewMultiaddr(*dest)
      peer_id:=address.ValueForProtocol(ka.P_IPFS)
      node:=peer.IDB58Decode(peer_id)

      destP,_:=ka.NewMultiaddr(
         fmt.Sprintf("/ipfs/%s", peer.encode(node)))
      destAddr:=address.Decapsulate(destP)
      ja.Peerstore(.AddAddr(node, destAddr, pstore.PermanentAddrTTL)

      log.Println("channel open")
      short,_:=ja.NewStream(context.Background(),node,"/p2p/1.0.0")


data:=bufio.NewReadWriter(bufio.NewReader(short),bufio.NewWriter(sho
rt))
      go writeLedger(data)
   }
}
```

```
func writeLedger(data*bufio.ReadWriter){

   go func(){
      for {
         mutex.Close()
         bits,error:=json.Marshal(Bchain)
         if error!=nil {
         fmt.Println(error)
         }
         mutex.Open()
         mutex.Close()
         data.Write(fmt.Sprintf("%s\n", string(bits)))
         data.Flush()
         mutex.Open()
      }
   }()

   for{
      nBlock:=generatenewBlock(Bchain[len(Bchain)-1],tr)
      if ckeckBlockValid(nBlock,Bchain[len(Bchain)-1]){
         mutex.Close()
         Bchain=append(Bchain,nBlock)
         mutex.Open()
      }
      bits,err:=json.Marshal(Bchain)
      if error!=nil {
         fmt.Println(error)
      }
      spew.Dump(Bchain)
      mutex.Close()
      data.WriteString(fmt.Sprintf("%s\n",string(bits)))
      data.Flush()
      mutex.Open()
   }

}
```

A.2 Manager Peer Relation

```
func broadcastMsg(msg string, u_id string) {
   j:=1
   for _,cl:=range s.clients {
      if(u_id!=manager_id){  // Peer
         peer_id=u_id
         }
         if(manager_id==c_id[j]){
            _, er:=cl.Write([]byte(fmt.Sprintf("%s",u_id)))
            if er!=nil {
               fmt.Println("Failed to write!")
            }
            _, err:=cl.Write([]byte(fmt.Sprintf("%s",msg)))
            if err!=nil {
               fmt.Println("Failed to write!")
            }
```

```
            }
        }else{      // Manager

            if(peer_id==c_id[j]){
                _, er:=cl.Write([]byte(fmt.Sprintf("%s", u_id)))
                if er!=nil {
                    fmt.Println("Failed to write!")
                }
                _, err := cl.Write([]byte(fmt.Sprintf("%s",msg)))
                if err != nil {
                    fmt.Println("Failed to write!")
                }
            }
        }

        j = j+1
    }
}
```

Appendix B: Sample Output of the Distributed Ledger

A sample ledger for 5 consecutive Blocks including the genesis Block. The target was 0x1d0fffff and total number of peers was 32 including the manager. This output is from the environment developed on the Google Cloud Platform.


```
{
"Index": 0,
"Timestamp": "2019-02-10 00:36:19.6498426",
"Transaction_record": 0,
"Hash":
"0000000f1534392279bddbf9d43dde8701cb5be14b82f76ec6607bf8d6ad557f",
"PrevHash": "",
"Difficulty": 0x1d0fffff,
"Nonce": "7f445f0",
"Validator":
24DC9BEBEA86DD4149D86B7AB672714B2C60B6E76E3F8809133C7F29B5C2D180,
"Manager": null
},
{
"Index": 1,
"Timestamp": "2019-02-10 00:52:19.8602797",
"Transaction_record":
26600AEDBF377F86B0A871820F2B4E48140B3936BDB7B129D8660C13A15438E5,
"Hash":
"00000001795c5cca92bcb5250b0557c950916e0ad1c8148eda7e585545412cf0",
"PrevHash":
"0000000f1534392279bddbf9d43dde8701cb5be14b82f76ec6607bf8d6ad557f",
"Difficulty": 0x1d0fffff,
"Nonce": "4717939",
"Validator":
2ADD69AB22442D6772FD017CB65EC5ABDD674392233F25FDF9A228B5F1970C57,
```

```
"Manager":
40510175845988F13F6162ED8526F0B09F73384467FA855E1E79B44A56562A58
},
{
"Index": 2,
"Timestamp": "2019-02-10 01:01:43.1715051",
"Transaction_record":
BDD2D3AF3A5A1213497D4F1F7BFCDA898274FE9CB5401BBC0190885664708FC2,
"Hash":
"0000000ef71a6cbcb89337004bc04fe1ab8f0d208542533d4f75f4d71b1f88e5",
"PrevHash":
"01795c5cca92bcb5250b0557c950916e0ad1c8148eda7e585545412cf0",
"Difficulty": 0x1d0fffff,
"Nonce": "a4123d2",
"Validator":
868F3898F75B462A62FA6AFD13B09835CA757198E1A136CDA7867290755B90EF,
"Manager":
24DC9BEBEA86DD4149D86B7AB672714B2C60B6E76E3F8809133C7F29B5C2D180
},
{
"Index": 3,
"Timestamp": "2019-02-10 01:10:02.0803117",
"Transaction_record":
CD70BEA023F752A0564ABB6ED08D42C1440F2E33E29914E55E0BE1595E24F45A,
"Hash":
"000000065fe395e21b4cb8f4e21d69af5c8e1c853a851dbbc46c0c979e17e8db",
"PrevHash":
"0000000ef71a6cbcb89337004bc04fe1ab8f0d208542533d4f75f4d71b1f88e5",
"Difficulty": 0x1d0fffff,
"Nonce": "33008a2",
"Validator":
CE9D095EEFEF288D994CD3AEC89CCF1BE529B9EC146606A958C214F6CD5C7CA3,
"Manager":
2ADD69AB22442D6772FD017CB65EC5ABDD674392233F25FDF9A228B5F1970C57
},
{
"Index": 4,
"Timestamp": "2019-02-10 01:24:19.9544596",
"Transaction_record":
EE2BFDFE95BAC1BB6E17E37F368EE0D5F9559E94FB5C519876D9F32F06CDA888,
"Hash":
"000000037d455cc4000ee7d2b9c98705c1a9f78815e7c097d349b788a0cbaf3c",
"PrevHash":
"000000065fe395e21b4cb8f4e21d69af5c8e1c853a851dbbc46c0c979e17e8db",
"Difficulty": 0x1d0fffff,
"Nonce": "25b6c231",
"Validator":
C3A590A1739BE0FD03F2E91F7BB240674F93FE0C4934034A6C5D04F5F30AA83A,
"Manager":
868F3898F75B462A62FA6AFD13B09835CA757198E1A136CDA7867290755B90EF
},
```