IMPLEMENTATION OF PARALLEL PROGRAMMING TO IMPROVE TRANSACTION SPEED AND SCALABILITY IN BLOCKCHAIN SYSTEMS

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Joshua Aaron DeNio

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

July 2021

Fargo, North Dakota

North Dakota State University
Graduate School

Title

IMPLEMENTATION OF PARALLEL PROGRAMMING TO IMPROVE TRANSACTION SPEED AND SCALABILITY IN BLOCKCHAIN SYSTEMS

By

Joshua Aaron DeNio

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Simone Ludwig
Chair

Saeed Salem

Ying Huang

Approved:

July 21, 2021
Date

Simone Ludwig
Department Chair

ABSTRACT

This thesis presents a parallel mining architecture model intended to be used in blockchain systems to improve transaction speed and network scalability while maintaining decentralization. Typical blockchain validation times are significantly slower than traditional digital transaction systems. The model presented is intended to allow devices with limited computational power to make meaningful contributions to the blockchain system by introducing parallel proof of work, managed by automated manager nodes. This will allow blockchain systems to be integrated into cloud environments and the internet of things. The presented model is also intended to address and reduce power consumption problems current blockchain systems face, by allowing the network to validate transactions without the need of high-powered specialty mining machines. Automation and virtualization of network nodes is intended to utilize hardware already online to preform parallel proof of work together in contrast to nodes all competing against each other and ultimately wasting electrical power.

ACKNOWLEDGMENTS

I would like to thank Dr. Saeed Salem for his support and guidance throughout my academic journey. He has always been the person to turn to with any questions and has always provided the best advice. I would also like to thank Dr. Simone Ludwig for being there for everyone. She is an inspiration to us all.

DEDICATION

I dedicate this work to everyone who has helped guide me to where I am today.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

1. INTRODUCTION

Blockchain technology has gained popularity in recent years causing a large influx of new interests in the potential application of blockchain technology into real world systems. Blockchain systems serve as the foundational technology behind cryptocurrencies and smart contract systems. In recent years blockchain systems have gained international attention particularly in the field of cryptocurrencies. This renewed interest has caused stresses on the current mining pools and in the cases where mining pools cannot keep up with the demand has even temporarily caused outages on some block chains [1, 2]. This thesis intends to introduce a change in the underlying architecture of blockchain mining that will be able to support the vast expansion and scalability while not wasting valuable resources. Current mining approaches are very resource intensive and compete with one another over mining the same blocks. In the current model only one miner or mining pool can actually solve a proof of work on a block at a given time, and as a result the remaining miners and mining pools have wasted their resources trying to mine the block [3]. This waste of resources causes the cost of contributing to mining to go up significantly as to remain competitive, the miners must either join a large mining pool or devote intensive resources comparable to a mining pool in order to be capable of completing a proof of work before a mining pool can complete the same block [4]. Therefore, research is needed to find a more efficient way to process blockchain transactions that will not result in the waste of electricity.

1.1. Background

In the case of traditional transaction systems, transactions are verified by a third party. This verification is required to ensure that both parties involved in the transaction can be assured that the transaction will be processed correctly and neither party can cheat the other. Historically this third-party roll has been filled by banks, institutions, governing bodies, and other such entities capable of settling disputes and ensuring the satisfactory resolution of transactions. Bitcoin was introduced in 2008 by a developer or group of anonymous developers who publicized Bitcoin under the pseudonym Satoshi Nakamoto [5]. Bitcoin introduced the idea of implementing a system of distributed legers over a peer-to-peer network. This system is now referred to as a blockchain. Transactions conducted on the blockchain are encrypted and stored on the network after being validated by nodes called miners. Miners work to validate transactions and integrate them into the blockchain from a transaction pool called a mempool [6]. In exchange for their services miners are paid a fee in Bitcoin upon completion of a block. Bitcoin's presented advantage is a decentralized transaction system. Centralized transaction systems potentially have a single point of failure where if the central authority is tampered with transaction data may be manipulated or changed. Hacking attempts show the vulnerabilities these central authorities face and the potential risks to the integrity of transactions on traditional systems. Blockchain technology reduces the likelihood that transactions can be tampered with dramatically as each node has access to the distributed ledger and transactions not validated by the network are rejected. Once a transaction is validated and added to the blockchain it cannot be removed and is stored in a safe encrypted form [7]. Another use that has presented itself for blockchain technology is the use of the blockchain to store smart contracts. These contracts are stored into the blockchain just like recorded transactions and once validated the contract is permanently recorded in the blockchain. Blockchain technology generally comes in two different approaches, permissionless and permissioned blockchains. In permissionless blockchains any node can serve to validate the blocks where in contrast permissioned blockchains designate what nodes are authorized to validate blocks. Both offer users access to a disseminated record of the complete transaction history of all affirmed transactions. Clients are free to check the exchanges between different clients without the need to gain permission from any external authority. The transactions themselves and the user’s information remain anonymous, and the entire ledger is recorded and continuously added to. This growing ledger is always available to all nodes on the network.

1.2. Decentralization

Blockchain technology has gained notoriety due to its lack of a central authority role in the case of permissionless blockchains. By providing decentralized transactions users have a very significant amount of freedom to conduct transactions that are free from oversight. Users of a decentralized system have complete authority over their assets and can freely exchange them with anyone at any time [8]. Although decentralization is one of the key points that attract people to cryptocurrencies, it is also one of the main reasons that governments have tried to regulate and ban them. Decentralization allows for the sale of goods without the ability of governments to step in and enforce tax laws or monitor transactions. The creation of Silkroad, an online black market launched in February of 2011, displayed just how attractive this decentralized transaction system can be to those who want to buy and sell contraband items [9]. Drug sales on the platform produced well over 1 billion dollars’ worth of revenue in Bitcoin, which was later confiscated by the United States government [4]. This use of decentralized transaction systems for illegal purposes has put a target on cryptocurrencies in the eyes of government agencies who worry about its possible use in money laundering and other illegal activities [4]. This work does not aim to address the issues of legality, nor does it intend to propose solutions to illegal use of this technology as these are complex issues and deserving of their own dedicated research.

The focal point of this work is to expand upon the scalability and efficiency of blockchain technology by integration of true parallel processing in a cloud environment. This should dramatically improve the transaction throughput of the network and reduce wait times between the initial transaction and confirmation. The implementation discussed and presented in this work is based on the research of Shihab Shahriar Hazari and Qusay H. Mahmoud which was presented in July of 2020 [10, 11]. This thesis aims to extend upon their previous work by adding to the manager roles to better apply redundancy and ensure that there is never a single point of failure. In this implementation, miners will work on the same transaction data with different nonce values. This will ensure that multiple miners do not compete against each other to solve the same work. Nonce values will be generated, managed, and distributed by the managers. A nonce is a seed value used to produce a hash through the hashing algorithm similar to a key. The goal is to find the perfect nonce that will enable the miner to produce a suitable hash value that can be linked to the previous block in the block chain. Mining pools use a similar approach with the difference being that mining pools introduce a centralized authority and enable the implementation of the pools authorities to implement the 51 percent attack on the blockchain if they control at least 51 percent of the computational power in the network [12, 13]. Our implementation will use true parallel computation in a decentralized manner making it significantly different than the implementation of a traditional mining pool [13]. The reward system will also be adjusted to ensure that contributors are all compensated for their contribution to the network fairly based on the amount of work performed.

1.3. Scalability concerns

A notable concern in transaction processing is the scalability of the system. What this means is how quickly can the system be expanded to handle additional load and usage and what is the maximum number of transactions that can be completed per unit of time. The system that supports Visa card transactions has already reached a peak of 10,547 transactions per second [14]. Scalability concerns effect all transactional systems including those that use blockchain technology [15]. The theoretical limit for Bitcoin transaction speed is defined by the following formula: (Block size limit) / ((Lowest possible text size) * (Block time in seconds)). Other cryptocurrencies use other protocols thus, have differing transaction speeds. Our aim is to improve both the scalability and reduce the overall transaction times required for block chain transactions by making better use of available network resources.

Table 1. Transaction Speed of Some Common Cryptocurrency Blockchains.

Cryptocurrency	**Number of transactions per second**	**Average block time**	**Number of confirmations required**	**Average confirmation time**
Bitcoin	3 - 7	10 Minutes	4	60 Minutes
Bitcoin Cash	61	10 Minutes	15	60 - 150 Minutes
Dash	48	2.6 Minutes	2	2 – 10 Minutes
Dogecoin	16 - 33	1 Minute	40	40 Minutes
Ethereum	15 - 25	14 – 17 Seconds	20	2 Minutes
Lightcoin	26 - 56	2.5 Minutes	12	30 Minutes
Monero	4+	2 Minutes	15	30 Minutes
Nano	100 - 185	5 – 10 Seconds	n/a	0.67 Seconds
Neo	1000+	15 Seconds	n/a	15 – 20 Seconds
Ripple	1500 – 50,000	n/a	n/a	4 Seconds

The data in Table 1 shows the capabilities of various crypto currencies as of the date of this printing. The data has been collected from the cited sources [16, 17, 18, 19]. As seen in Table 1, the number of transactions per second is quite low in contrast to traditional transaction systems [20], for example, the Visa network is capable of handling up to 65,000 transaction messages a second [21]. Ripple is much faster than the other crypto currencies in the table, this is due to the use of centralization used to validate transactions and it does not use proof of work nor a blockchain. While Ripple also known as XRP proved that cryptocurrencies can be fast they also showed some reasons for concern with the centralization of cryptocurrencies. While the original vision for crypto currencies was that they facilitate a decentralized transactional system, there is a growing push for centralization from governments and large financial entities as well as mining pools. Ripple has shown us some of the possible problems' centralization can introduce. The main problem Ripple exhibited is the unrestrained production and sale of tokens. Ripple executives are documented to have produced and sold over 14.6 billion XRP coins for a profit of over 1.38 billion US dollars [22]. These coins were produced and sold to create wealth for the Ripple executives and have caused users of XRP to challenge Ripple legally. The goal of this research is to improve the transaction speed while also maintaining decentralization through the implementation of parallel mining using network managers that are not controllable by any party on the network.

1.4. Power consumption

Energy consumption is another issue that effects the scalability of blockchain systems. Bitcoin, a leading cryptocurrency, is said to consume as much power as a small nation [2]. In 2018, the Bitcoin network was estimated to consume approximately 3.57 Gigawatts of electricity [23], and by 2019 the energy consumption was estimated to have risen to 7 Gigawatts [2]. An estimate provided by the University of Cambridge stated that in 2021 the energy consumption of the Bitcoin network will consume more than 178 Terawatt-hours. Making the Bitcoin networks' power consumption comparable to that of Switzerland. This tremendous power consumption has led to the embrace of renewable power supplies such as Geothermal in Iceland, Hydro-electric in Quebec and Austria [24]. Washington state has also attracted the attention of companies interested in mining Bitcoin due to its use of Hydro-electric power and surplus of electricity [25]. Thus, improving the efficiency of the mining process will not only increase the usability of a cryptocurrency but will also decrease environmental strain due to the carbon footprint required to run the network. Today climate change is a major concern and by increasing the efficiency of blockchain technology we can help reduce the effects its use has on our environment.

To improve the efficiency of blockchain systems we must first look at the underlying processes as they currently are employed.

1.5. Mining

Mining is the term used to refer to the process of validating transactions and adding them to the blockchain. Nodes on the network use their computational resources to fulfil the role of a miner and if they contribute to the mining operation, they are awarded a processing fee in some way depending upon the framework being used [7].

Mining requires these three main functions to be performed by the miner:

- Verify transactions: Transactions are verified using peer consensus to ensure that only valid transactions are saved to the blockchain from the mempool.
- Create a block: A block is created, and miners must find a suitable hash using cryptography to interface the new block with its predecessor. Once the hash is found the new block can be added to the blockchain at which point other miners can verify its validity.
- Verify a new block: Miners verify the validity of hashes in the blockchain and reject any transactions that are not backed up by the rest of the network. This makes tampering with the blockchain virtually impossible without direct control of every node on the system.

It is important that a miner has the computational capabilities to solve the cryptographic hash in a timely manner so as not to cause time discrepancies in the blockchain. Miners contribute to the blockchain mining process in hopes of gaining financial compensation for their contribution. In the upcoming sections, we will discuss some of the methods employed to track an individual miner's contribution to the mining operation.

1.6. Proof of work

Proof of work is one of two commonly used Sybil deterrence mechanisms used in cryptocurrency mining [5, 26]. Proof of work was originally developed to deter against Denial-of-service attacks and reduce spam emails by requiring senders to perform a set amount of work per request. The first version of proof of work was introduced by Cynthia Dwork and Moni Naor in 1993 as a means of safeguarding against denial-of-service attacks on their network as well as an attempt to reduce spam emails [27]. Their methods were later formalized by Markus Jakobsson and Ari Juels who introduced the term “Proof of Work” in their paper written in 1999 [28]. Jakobsson and Juels formalized the concept of proof of work as we know it today, and it later became popularized by its adoption into the Bitcoin mining process.

Proof of work is the method used to add new blocks to the blockchain by confirming the transactions and adding the transaction to the block as well as finding the required hash value to tie the new block to its predecessor. Miners compete against each other to execute the proof of work before the other miners as only those who solve the proof of work get paid for the operation. Proof of work requires that the miner finds a suitable hash value which is done using algorithms to solve a complex mathematical puzzle to obtain a specific desired output by finding the perfect key. This process contains several elements namely a puzzle protocol and a hash function. The larger the network grows the more complex the puzzle becomes. The cryptographic algorithm used in Bitcoin mining is SHA-256, Ethereum uses the Ethash algorithm, and Lightcoin uses scrypt hashing [29]. The efficiency of the algorithms used has an impact on the overall transaction time. Proof of work is used to both prove that miners are contributing to the execution of transactions as well as enabling the formation of the consensus strategy used in permissionless blockchains. Depending upon the number of nonce values tested probabilities can be used to determine whether the miner solved the puzzle correctly or if some shortcut was used to circumnavigate the protocol. Depending on the miners’ computational capabilities the probabilities of the node finding a suitable nonce value can be easily calculated and used in the validation process. Any miners who do not meet the normal probabilities can be considered invalid contributors and blocks they attempt to add can be checked against the others to see if they are attempting to tamper with the blockchain. Any miners found in violation can then be blacklisted and excluded from future work on the blockchain and the blockchain will fall back to the last validated state. To validate a transaction, it is critical that the validation steps required to ensure validity are asymmetric. Meaning that the work of performing the work should be substantially more difficult than the work required to verify that the work was done correctly. Validation should be possible with a minimal consumption of resources so as not to bog down the entire process. Many cryptocurrencies have improved their performance by reducing the number of steps taken to ensure validation [15], where this does result in improved system performance it also introduces possible security problems.

1.6.1. Tragedy of commons

The Tragedy of Commons is a potential problem when using Proof of work. This occurs when there are few miners available due to little or no block reward. This results from users opting to pay lower fees if possible and as a result fewer miners contribute their resources because of the diminishing returns. This occurs when the only fees to be earned are transaction fees, which also can diminish if not managed correctly. The tragedy of commons increases the systems vulnerability to the 51 percent attack as the lower the number of contributors the easier it becomes for a malicious party to gain an upper hand in the network. More details about the 51 percent attack will be addressed later in this thesis. As for the Tragedy of Commons our presented proposal will counteract the Tragedy of Commons by allowing even small computational devices to contribute to the processing power of the network. Thus, so long as the network is being utilized then every device that is a part of the network can contribute to the overall function of the network. There will be less need for large expensive mining rigs and reduced power consumption. Devices that are left on normally will be able to contribute to mining without dramatically effecting their power consumption. Our approach is related to the internet of things concept where even small computational devices can contribute to the overall system. As a result, the network itself will provide mining services and there will be reduced reliance on dedicated mining rigs. Rewards will still be beneficial and will encourage users to leave their devices connected to the network.

1.7. Proof of stake

Proof of Stake is the second Sybil deterrence mechanism we will discuss in this work. Proof of stake differs from proof of work as it requires miners to prove that they are invested in the network to a required degree. This consensus mechanism relies on the miners showing that they are invested in the blockchain before they can contribute to adding blocks to the blockchain. Proof of stake requires that malicious users obtain a large influence in the network before being able to manipulate the blockchain. Unfortunately, mining pools allow the entities controlling the mining pool to easily amass such influence over the blockchain [30, 31]. Proof of stake enables a miner to mine blocks based on the number of coins they own. This can cause problems when a wealthy party buys a majority share in the network granting them control over the mining operation. Proof of stake was introduced as a means of reducing the overall power consumption of the network by reducing the computational cost of completing blocks. Where it has reduced the power needs of blockchain networks that employ proof of stake it has introduced its own set of problems and reduced the level of decentralization by basically granting network control to the wealthiest parties. On the positive side an entity with 51 percent of the cryptocurrency on a network is less likely to see advantages to attacking the network considering that they could potentially loose the invested capital used to gain the 51 percent share.

1.8. Proof of activity

Proof of activity is another method that is currently being implemented and attempts to combine the benefits of both proof of work with that of proof of stake. The proof of stake system starts with a mining process like the proof of work system but differs after a new block is mined where the system transitions to resemble a proof of stake system. Decred is currently the most well-known implementation of this approach [32, 33]. This consensus mechanism is also compatible with parallel mining and has a great deal of potential as the consensus mechanism of choice for our application of parallel mining into cryptocurrencies.

1.9. Mining pools

Mining pools were introduced to solve the problems associated with the increasing computational power required to solve the puzzles used in cryptocurrency mining as the blockchain grows in length. The general approach is to divide the work among miners in the pool and split the reward with everyone that contributes to the mining operation. This approach to mining has many positive contributions to the cryptocurrency sphere but it also introduces a few problems.

1.9.1. Problems with mining pools

The 51% attack occurs when a malicious party gains control over 15% or more of the computational power of a mining network [12]. This enables them to validate fraudulent blocks and invalidate valid transactions. They also gain control over the mem pool and can discard transactions at will [12].

1.9.2. Attacks against competing mining pools

Another issue with the use of mining pools is the inherent competitive nature of conflicting interests, where it is beneficial for one party to make money be reducing the amount of money another entity can make. As with any competing entities mining pools find themselves at odds with their competitors and this can lead them to do things to undermine their competitors. Examples of this include, but are not limited, to cyber-attacks, physical attacks on hardware, attacks on power grids, attacks on reputation, and so on [12, 31]. There are even instances where large mining pools have taken down the value of a crypto currency because they did not get their way in regard to decisions made by the developer teams or national regulatory agencies [31]. The amount of power wielded by these mining pools on the Ethereum blockchain goes against the fundamental idea of decentralized currency.

1.9.3. Gas

The Ethereum network and other related blockchains utilize a concept called “Gas”. Gas is a form of payment to run transactions on the blockchain and has many positive uses but also serves to increase the cost of very small transactions.

1.9.4. Scalability concerns

Mining pools can also suffer from scalability constraints as there are often regulations concerning joining a mining pool and as a result some users may not join a mining pool as they could be concerned with all the fine print and rules associated with membership of a given mining pool. There may also be issues stemming from mining pool directors taking time to integrate or accept new nodes into the mining pool. This time taken could potentially affect the scalability of the network. Another scalability concern is dated hardware, where the operators of the mining pool may not consistently update the hardware and as a result the systems running the control of communication between nodes on the pools network may not be communicating as effectively as those that are out on the internet. As the pool is restricted to communicate through the servers run by the pools owners the overall network is dependent upon the pool owners to ensure the network can handle the traffic and load applied to the mining pool.

2. RELATED WORK

The work presented in this thesis is based on the research presented by Shihab Shahriar Hazari and Qusay H. Mahmoud in their paper published in July of 2020 [3]. This approach is also related to the mining pool approach but differs in its more decentralized application, manager roles and responsibilities, as well as the contributions of active miners and the reward system also has some significant differences.

The managers used in the presented solution are based on the idea of coordinator selection, which was first implemented by Gerard Lelann in 1977 [20]. The presented solution greatly improves their role and increases redundancy thus, reducing any potential for a point of failure. This role of process coordinator is a crucial part of improving the quality and performance of a distributed system. Dework et al. introduced a consensus protocol using coordinator election for a partially synchronous processor in 1988 [34]. In their presented work the coordinator distributes work to peers in proportion to the number of peers within the network. When the work is completed, a final decision is made using the consensus protocol to validate the work.

2.1. Bitcoin-NG

Bitcoin-NG, introduced in 2016, presented the concept of decoupling Bitcoins blockchain task into two planes: a leader selection and exchange serialization plane [35]. Bitcoin-NG also partitions time into epochs where each epoch has a solitary leader [35]. In our application of managers, the leaders will form a team where one is active, and the remaining leaders are available to take over if something happens to the active manager. The solitary leader used in Bitcoin-NG introduced a single point of failure, which our solution addresses by adding redundancy to the role with the implementation of a team of backup managers in proportion to the network size. These back up managers will all be in synch with the active manager, and when the active manager fails the group will promote a new manager to active status randomly, then update the miners of the new active manager. Bitcoin-NG uses two types of blocks one being a Key Block and the other being a Microblock [35].

The Bitcoin-NG key block contains information relating to the leader as well as the previous block. In contrast, the Microblock contains the transaction information. Proof of work is used to produce the Key block. Once a leader is selected the leader is charged with issuing Microblocks using the leaders private key, which also contains the transaction information. Microblocks contain no proof of work so have little effect on the overall weight of the blockchain [35].

2.2. The Boyen model

Another related approach is that presented by Xavier Boyen et al. also in 2016 [36]. In their approach each transaction is connected to two or more verified transactions and miners verify new transactions in a parallel network. The network used consists of a graph structure like the network structure used in Bitcoin, Tangle, and IOTA [36, 37]. The Boyen model also utilized proof of stake rather than proof of work to validate blocks and append them to the blockchain [36]. Boyen's approach has done away with the traditional blockchain and implements a lean graph of transactions making verification times much faster [37].

2.3. The Hazari-Mahmoud model

Shihab Shahriar Hazari and Qusay H. Mahmoud presented a model in 2020 which the presented approach is based on [3]. Their presented model makes use of a single manager node and a network of miner nodes [3]. The manager is selected based on which node completed the last block making it easy to predict and target the manager. The network is a peer-to-peer network [3], which also poses potential vulnerabilities to attack as a single node loss could cause the network to temporarily loose communication capabilities as nodes reconnect. The aim of this thesis is to address the possible weaknesses of the Hazari-Mahmoud model and make a derived model that is more robust and resistant to node failures while maintaining the benefits of the Hazarti-Mahmoud model of distributed parallel proof of work. The presented approach differs from the Hazarti-Mahmoud model in that it introduces multiple manager roles to account for redundancy and introduce more recoverability. Also, the presented approach introduces a star like hybridization of the ring peer-to-peer network topology used in the Hazarti-Mahmoud model [3]. This interconnects each node more tightly with other nodes in the network and allows for better data retention and less local forking of transactions entering the mempool as more nodes will be present locally to verify the incoming transactions.

2.4. Mining pools

Mining pools were developed when the complexity of single computer mining became unfeasible due to the growing complexity of generating a valid proof of work for a block. Miners pool their resources to process the hashes and validate blocks. Mining pools work in parallel but are generally controlled by a central entity that may or may not always act with everyone’s best interest in mind. Mining pools generate considerable computational power and solve hashes quite effectively. The rewards are distributed depending upon the rules set up by the mining pools controlling entity. There are several differing reward systems implemented such as: Shared Maximum Pay Per Share, Capped Pay Per Share with Recent Backpay, and Equalized Shared Maximum Pay Per Share, to name a few [38].

A problem arises when multiple mining pools or individual miners mine on a network. In the case of Bitcoin miners and pools work to process transactions and generate the next block in the series. Unfortunately, only the miner or mining pool that successfully adds the block gets payment and the combined work of all the other parties on the network are wasted. This wasted effort not only consumes large amounts of electrical power but also consumes valuable resources as expensive computer components burn out and fail over time. This creates a high demand to create new hardware and by extension mine more raw materials from the earth. The only way mining pools could alleviate this waste of resources is if a single mining pool had complete control of a blockchain's mining operation. Which would in turn introduce centralization to the blockchain network. The proposed solution presented in this thesis aims to solve this situation by creating a completely decentralized mining pool and promoting miners to solve the proof of work by also distributing the reward based upon their individual worked contribution to processing a block. This will maintain decentralization while also reducing the amount of power and resources consumed by the network.

2.5. Problems with existing parallel approaches

The existing parallel implementations utilize a single manager system which could present a single point of failure to be targeted by malicious parties. While the system will not completely fail in the existing strategies if the manager node is removed from the network or compromised; the system will suffer a speed decrease as the system will revert to solo mining speeds making the blockchain itself more vulnerable to attacks on the blockchain. The manager nodes in the above listed approaches are comparatively easy to track and predict making targeting them for attack more convenient and increasing the likelihood of a successful attack on the network. Since blockchain transactions are transparent and the node that solves the hash gets the manager role it is easy to track the nodes that are in line to become manager in the event of a failure of the current manager. Another vulnerability is the peer-to-peer communication model used as any disruption of single nodes can have an impact on the overall network's functionality. The presented approach intends to address these problems by introducing a group of managers that will introduce redundancy to the system and in the event of a failure of the active manager will replace it immediately without dropping the network back to solo mining. The network architecture as well will be changed to include more redundancy than a normal peer to peer network by utilizing broadcast like protocols such as those used in MPI (Message Passing Interface) to ensure that when nodes disconnect from the network there is little impact on those remaining on the network.

3. PROPOSED APPROACH

In this thesis, we present a method to improve the transaction speed and scalability by implementing parallel processing and validation of blocks across a decentralized network of peers. In the method proposed most of the nodes will perform work to validate the same block and the remaining nodes will be the active manager and backup managers, respectively. Miners will receive data from the active manager and the backup managers will receive updates from the active manager and take over if the active manager becomes unavailable. We will develop a consensus mechanism to ensure that managers are randomly selected, and nothing can be done to force a specific node to become a manager. This will be done using the update feature of the managers as well as a detection mechanism to determine if anything suspicious has occurred to cause the transfer of the manager role in which case the management team or group of managers will randomly select a new manager and purge any suspicious managers from the network. Worker nodes will perform all the required steps to validate transactions and append to the blockchain requesting new nonce values, as necessary.

3.1. Node roles and communication

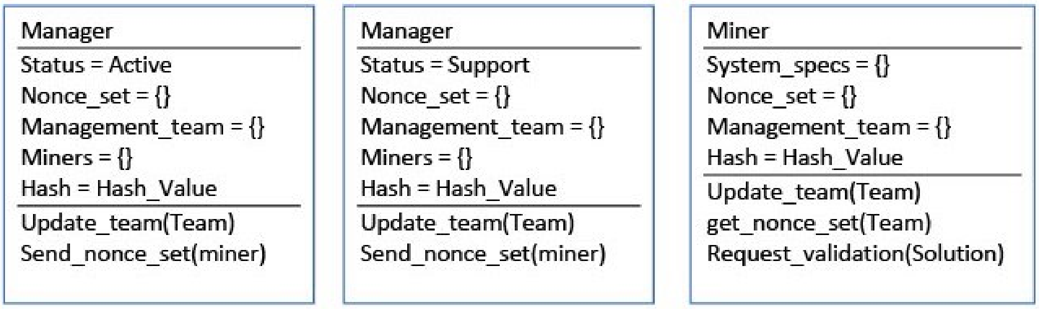


Figure 1. Node roles.

Figure 1 shows a visual representation of the 2 types of nodes used in our implementation and the two functional states that the manager nodes can be in at any given time.

3.1.1. Managers

Managers will be implemented in two forms: an active manager and a team of support managers we will call the management team. The active manager will distribute transaction data from the mempool to the miners and will issue each miner a set of nonce values. The active manager will ensure that no two miners receive the same nonce values for a given block. The manager is also responsible for creating the transaction hash that miners are to solve, along with the nonce value set they will apply to attempt to solve the hash.

When the nonce values are depleted, the active Manager will generate more and disperse them as needed. Another roll of the active Manager is to synch with the management team and keep them up to date with its activity as well as nonce ranges and data dispersed.

The role of the Support Managers is to form a team and if needed replace the active Manager in the event of a failure. The replacement should be randomly chosen to prevent a malicious user from gaining control over a manager and tampering with the network. The Support team will periodically elect new managers at random so long as they meet the system specifications required to fill the role. The management team will be a parallel network nested within the main network and should be continuously in contact with each other to ensure they are all in synch and contain the same information. If a single manager has differing information the others will expel the corrupted manager from the team and elect a replacement.

The management team will replace the active manager at given time intervals we will call epochs like those used in Bitcoin-NG [35]. This will be done to give rest to hardware components and ensure that the management role does not stay active on a specific hardware device for an extended amount of time. This is directed at reducing the ability of users from tampering with the active manager as they should have no way of knowing what node will become the active manager or for how long it will be active. The management team will monitor its members and elect new Support managers as needed and rotate them in and out of service depending upon the needs of the network. The level of redundancy required will need to dynamically scale with the size of the network with a percentage of the network being support managers to ensure that there will never be a failure that will remove them all at once. If possible, they will be geographically chosen to ensure that they are dispersed enough to evade failure even in the case of a continental power outage. As shown in Figure 1, the managers status will define its role and the support managers will be tasked with monitoring what manager is currently active and in the case that an incursion occurs, and the active manger is changed without the consensus of the group, the offending node will be removed from the network.

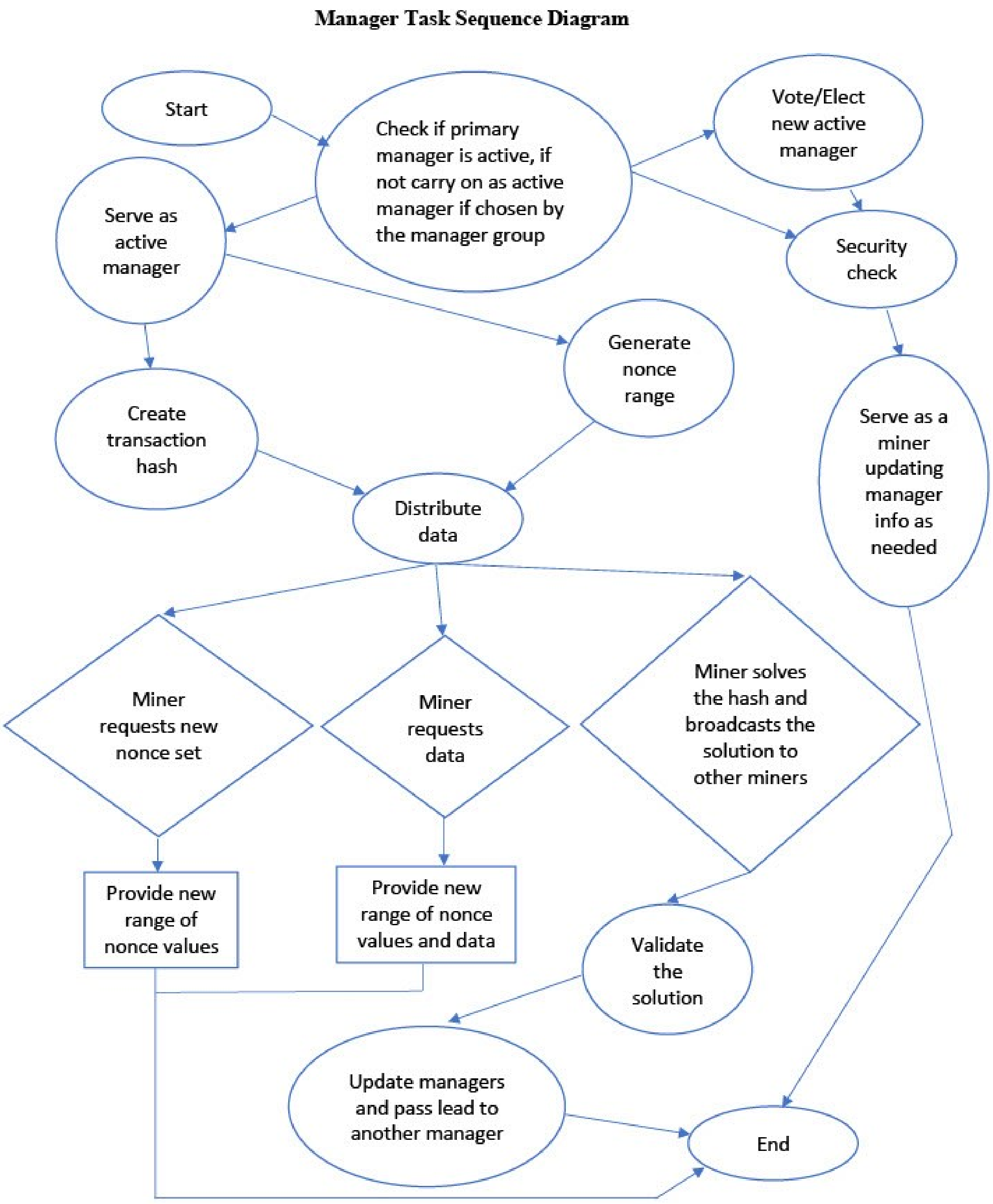

Manager Task Sequence Diagram
Start
Check if primary manager is active, if not carry on as active manager if chosen by the manager group
Vote/Elect new active manager
Serve as active manager
Security check
Generate nonce range
Create transaction hash
Serve as a miner updating manager info as needed
Distribute data
Miner requests new nonce set
Miner requests data
Miner solves the hash and broadcasts the solution to other miners
Provide new range of nonce values
Provide new range of nonce values and data
Validate the solution
Update managers and pass lead to another manager
End

Figure 2. The workflow of a manger node in a parallel network.

Figure 2 shows the general workflow of the manager nodes. In the case that the manager is not an active manager it will be tasked with monitoring the activity status of the active manager and verifying security, optionally it could temporarily act as a worker if everything is up to date. If the active manager is not active the group will elect a new active manager based on the capabilities of the available managers in a semi random selection putting preference to those with the highest computational capability. It is important that it be difficult to predict what manager will step up to fill the role to reduce the likelihood an attacker is able to target the next active manager in advance. In the case of the node being the Active Manager, it will be tasked with distributing data and nonce values and reporting to the management team to verify that it is active and inform the team of the current data and nonce values in operation.

3.1.2. Miners

Miners in parallel mining will initially send a request to the management team, which will be accepted by the active manager. The Active Manager will then send a block of data and a set of nonce values to the miner. No two miners should be doing the same work at the same time. The miner will attempt to solve the puzzle and generate a suitable hash value using the set of nonce values received. Once a suitable solution is found the successful miner will broadcast the completed block and the other miners will check to verify whether the solution is valid. If the solution is acceptable, they will update the manager and they will receive the next set of data to begin working on the next block. All miners will receive the same data set but each one will work on a separate set of nonce values. Once the set is depleted and if no solution was found a new request will be sent to the Managers and the miners will await a new set of nonce values.

Miner Task Sequence Diagram

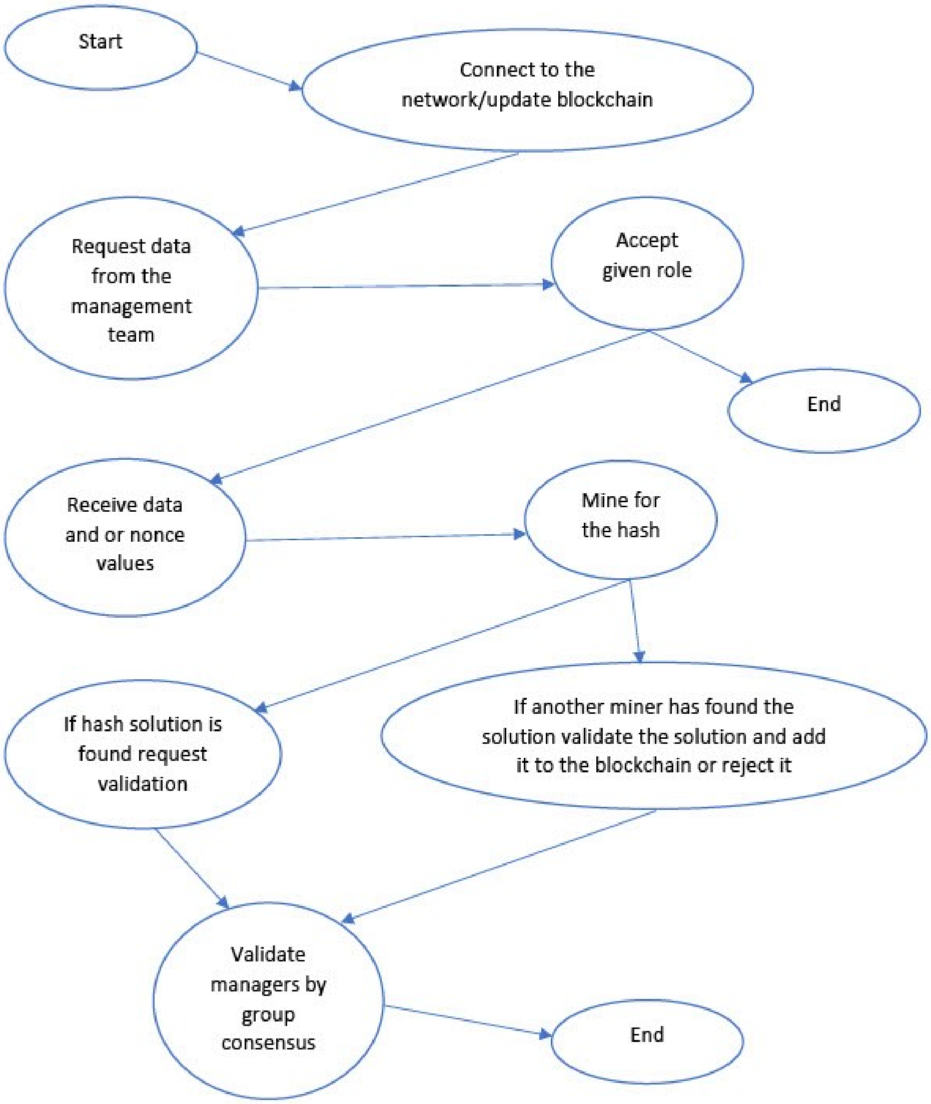

Start
Connect to the network/update blockchain
Request data from the management team
Accept given role
End
Receive data and or nonce values
Mine for the hash
If hash solution is found request validation
If another miner has found the solution validate the solution and add it to the blockchain or reject it
Validate managers by group consensus
End

Figure 3. Miner task sequence diagram.

Figure 3 shows details concerning the tasks a miner performs starting with its connection to the network. At this point it will contact the management team and receive an update to its blockchain data. The miner will then request data to work as well as data relating to management team rolls used to validate managers as group consensus will be used to ensure that false managers cannot be injected into the management pool and only those elected are accepted as managers. The miner will then be given a roll by the management team. This role may be as a miner, or a supporting manager based on the needs of the network. In the case that the node is elected as a manager it will accept the role. Otherwise, the node will then receive data and or nonce values and will move on to mine the hash. If it finds a solution it will broadcast the solution to be validated by the other nodes. If another node broadcasts a solution the solution will be checked and if valid added to the blockchain and if invalid measures will be taken to address the issue. Next, a security and validation phase will be executed to ensure that only authorized nodes are serving as managers and in the case that a node is not abiding by its designated role or otherwise posing a threat to the network actions will be taken to remove the threat from the network.

3.2. Network communication and security

In traditional blockchain systems nodes are connected to one another via intermediate nodes. In our presented approach to parallel mining, the nodes will still be connected via a peer-to-peer network, but they will also be connected to a team of managers that are also interconnected and in sync with each other.

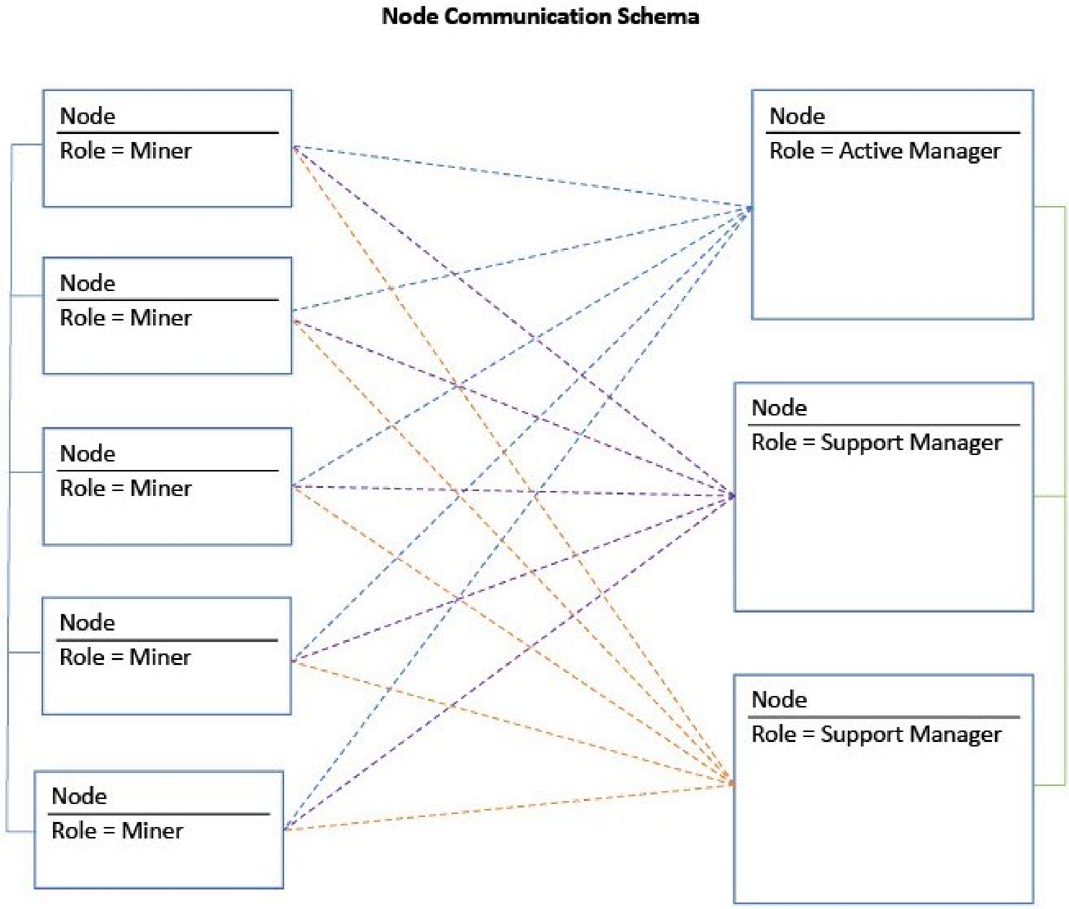


Figure 4. Node network communication.

Figure 4 shows the communication of nodes within the network. Note that each miner communicates to all managers and all managers communicate with all miners and other managers. Thus, each node is directly communicating with the managers and its peers, and the managers are communicating with each other collectively. Miners communicate with other miners using standard peer-to-peer ring topology communication but communicate with managers using a modified star topology resulting in a hybrid topology. This is to ensure redundancy and prevent a single point of failure. All communication should be verified from the other sources and rejected if the data is invalid. Consensus is required to validate any block and managers must agree upon the election of new managers. This consensus will ensure decentralization and prevent tampering with the network. Figure 1 shows more details of the node roles and how they differ. This modified star topology differs from the ring topology used in existing approaches and introduces more resiliency while maintaining concurrency in communication.

Table 2. Network Topology Comparisons

Ring Topology	**Modified Star-ring topology**
Within a ring topology each node is connected to nodes on its left and right sides.	In this variant of the star topology nodes are connected to a group of managers that function as the central hub.
Within a ring topology any node can be a point of failure.	There is no single point of failure as in place of a single node in the hub there exists redundant replacement nodes that serve together to replace the single point of failure in a star topology.
Ring topologies are cost efficient for single transfer, but costs increase as the communication hops from node to node.	The communication cost increases with the number of additional managers added to the network.
A ring topology passes data from node to node requiring the data to be exposed to potential vulnerabilities and threats before it reaches its destination.	Data is transferred directly to the management team reducing data vulnerability and increasing the resolution speed of executions.
Less connections are required to transmit data.	Star topologies require more active connections to function properly.
There are n links in the ring topology where n is the number of nodes.	There are n*k + ((k-1) *k) + n links in the modified stare topology; where n is the number of worker nodes and k is the number of manager nodes.
The connection must be broken when adding a new node to the network.	The connection is not broken when new nodes are added to the network.

Table 2 shows some contrasting differences between a typical peer to peer topology and the modified star topology used in this implementation. Communications between nodes have been tested using the Go language using the libp2p library found at [39]. An implementation was also tested using MPI broadcast, scatter, gather, send, and receive to accomplish the same communication topology type. Further research can be done to address the most effective communication methods and determine if MPI can be used to replace peer to peer communication. While this communication schema introduces more network communication it ensures that the network is more robust and will be able to handle multiple failures while remaining in operational condition. This topology also allows for rapid updates between nodes and faster group consensus as all nodes will have access to the records of the management group and the log history. This will allow the network to monitor itself for violations and anomalies such as users trying to change the role of a node. These anomalies once detected will be handled by removing the offending node from the network or disregarding its input. Other related security threats should be identified by the management group and updates to the groups security protocols should be regularly implemented to ensure that the system remains robust and able to resist attacks.

3.2.1. Points of failure

When compared to existing techniques such as those presented by [35] and [36] the additional backup managers remove the points of failure present in both implementations as the before mentioned approaches both utilize a single manager that when targeted or removed have a noticeable impact on the network's functionality and operational capability. In the presented approach the vulnerabilities presented by only having a single manager is addressed by providing backup managers that can step in and fill the role when needed. This will result in very little impact of the loss of a single manager. Also, the network itself is more robust than that used by [35] and [36] as a normal peer-to-peer network will suffer when nodes are removed, and interruptions can occur. In contrast the presented approach will maintain more communication links and even with the removal of nodes no interruption can occur as the remaining links will serve to maintain communication across the network. Managers will also be more difficult to target as the roles will be changed for each block and there will not be an obvious way to predict what node will become the next Active Manager. In contrast to the approach presented by [36] where the manager is assigned based on the node that has solved the last block making targeting of the active manager more straightforward. Managers in the presented approach will be elected semi randomly with a preference to those nodes with greater computational capabilities and even when a node becomes the active manager the specific address of the active manager is not directly logged in the blockchain for users to view. The active manager will not be directly communicated to instead the miners will send their requests to the manager pool and the communications will be accepted by the manager acting as the Active Manager.

3.2.2. Denial of service attacks (DDOS)

Research into the security vulnerabilities of blockchain technology indicates that DDOS attacks have greater effect on blockchains than on other more traditional transaction networks [12]. In the event of a Denial of service attack the Active Manager will be overwhelmed and the management team will need to elect another active manager. At this point the management group should trigger a defensive feature to detect the reason that the active manager has failed. An effective detection mechanism should be developed and deployed on the manager nodes to detect the initiation and execution of a DDOS attack and impose countermeasures to mitigate the effectiveness of the attack. Once detected the management team or active manager can blacklist or expel an offending node from the network if requests from the node seem to indicate a possible DDOS attack. Each node serving as a supporting manager will be periodically monitoring the active manager and ensuring that it is responding to requests effectively. This monitoring process can be used to detect a DDOS attack as the number of requests will be significantly higher than normal and should have other characteristics that separate these requests from valid requests that can be used to identify an attack.

3.2.3. Malicious managers and detection methods

Malicious managers are a possible threat that will be addressed by the management team by periodically verifying that only managers that have been elected are serving as a manager. If a node is detected that is acting as a manager, but its election is not traceable by the Support Manager pool it will be expelled from the network and if needed any invalid data added to the blockchain by the expelled manager will be rolled back to prevent corruption of the system by malicious managers. Only nodes that are dynamically elected may serve their respective roles. Logs will be maintained in encrypted form to allow the managers to trace all the roles served and elected for each block.

3.2.4. Validation

Validation will follow existing methods of peer consensus used in blockchains such as Bitcoin. Forks that deviate from the accepted branch of the blockchain will be discarded in favor of the longer accepted valid blockchain preventing malicious nodes from attempting to insert transactions into the blockchain. This does not differ from the methods used currently in proof of work blockchains. Double spending verification will be handled like that of Bitcoin with no real difference in the block validation process, and the number of blocks used to validate transactions will not differ from validation methods employed by the Bitcoin network.

3.3. Genesis block

The initial block will be created by the miners if the blockchain is empty. This first block will contain no transactions. Initially all managers will be elected at random using a test protocol to determine that they possess the required computational capabilities to fulfill the role. The node that completes the first block will become the active manager for the next block. Each new block will start a new period that we will call an epoch, with a new active manager managing each new epoch.

3.4. Manager election

Support managers will be elected in an unpredictable fashion with consideration to their computational contribution to the network. Nodes that provide a greater computational contribution will have increased odds of being elected to the management team. This will ensure that each manager elected has the capability of fulfilling its role. Manager candidates can be selected based on their computational capabilities and added to an array. Managers can then be selected from the group of possibilities using random number generation from the management team to select the index of the newly elected manager. This should result in managers being capable of handling the loads required to fulfill their roles and ensure that the incoming managers cannot be predicted by outside observers. If for some reason the active manager is unresponsive or slows to an unsatisfactory response time the management team will elect a replacement to fill the active manager role. In cases where the system has scaled to the point that a single node can no longer support the tasks of the managers, nodes can be combined to divide the tasks and fill the role as if they were a single node. This can be done using virtualization to increase the computational capabilities of nodes within the network [40]. As managers require more computational capability than miners the management team will need to be capable of measuring and dynamically scaling to meet the networks requirements.

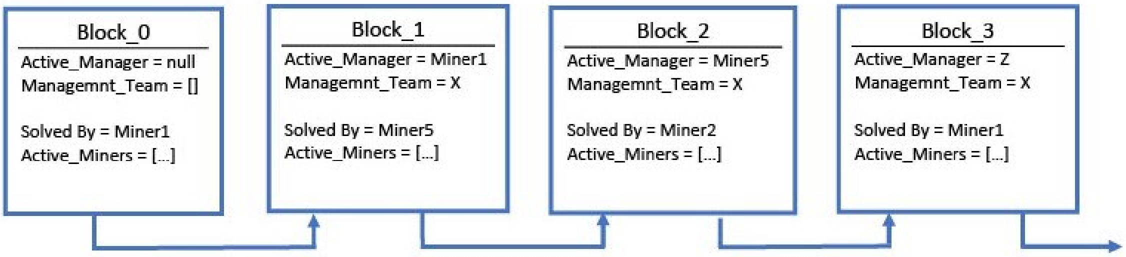


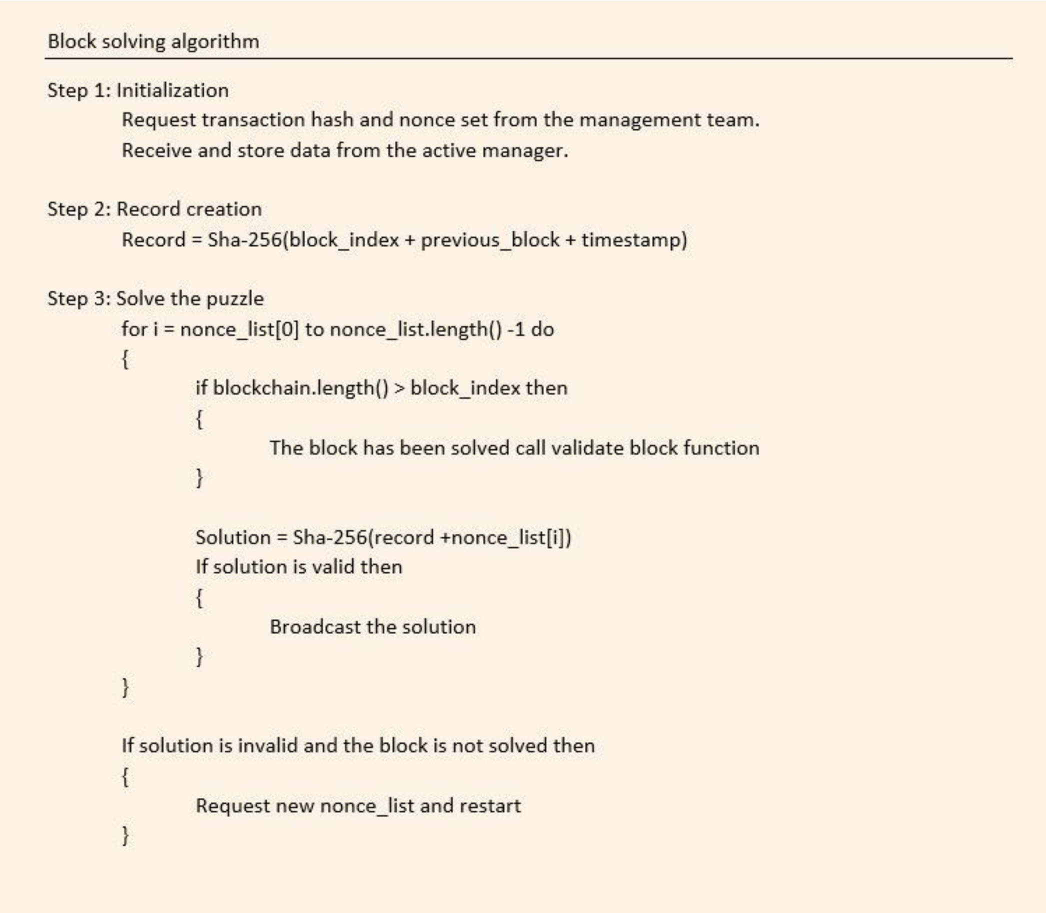
Figure 5. Block flow.

The Active_Miners in Figure 5 are sorted by top performing miners. X = the team of support managers sorted by their capabilities. Active managers are chosen from the list X and promoted randomly with a preference for those that have the highest capabilities. X is a list populated with its number of elements as a percentage of the network, from the most capable active miners.

This application of rotating elected managers ensures that no single node can gain excessive influence over the network thereby maintaining the decentralized nature of the system. Figure 5 illustrates the block flow and the main elements acting within a block that allow for the transition of workers and managers from one block to the next.

3.5. Algorithms

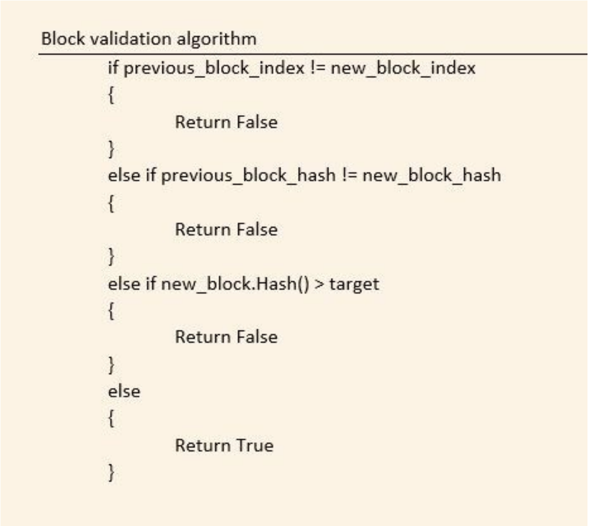
In the presented parallel mining system, each miner will have an equal opportunity to become a manager. Each miner will be compensated for their contribution to the network at the end of each epoch. This role of compensation will be monitored by the group of managers and only by full consensus will they distribute pay to the workers. Obviously, it will be beneficial to have the most powerful hardware involved in the management role. The miners who invest more resources into producing the greatest mining power will have higher likelihood of being elected to the management team. And as a reward will gain more revenue based on their network contribution. As the more powerful machines in the network solve more nonce values, they will in turn receive larger compensation than those who work less. Figures 6 and 7 show the basic algorithm chosen for block solving and block validation.



```
Block solving algorithm

Step 1: Initialization
    Request transaction hash and nonce set from the management team.
    Receive and store data from the active manager.

Step 2: Record creation
    Record = Sha-256(block_index + previous_block + timestamp)

Step 3: Solve the puzzle
    for i = nonce_list[0] to nonce_list.length() -1 do
    {
        if blockchain.length() > block_index then
        {
            The block has been solved call validate block function
        }

        Solution = Sha-256(record +nonce_list[i])
        If solution is valid then
        {
            Broadcast the solution
        }
    }

    If solution is invalid and the block is not solved then
    {
        Request new nonce_list and restart
    }
```

Figure 6. Block solving algorithm.

Figure 6 shows the details of the algorithm used to solve blocks. This algorithm is written using pseudocode and is intended to convey the basic logic involved in the block solving process. Nonce values and data will be provided by the Active Manager and backed up to the supporting managers to preserve redundancy and recoverability in the case of a node failure.



```
Block validation algorithm
        if previous_block_index != new_block_index
        {
                Return False
        }
        else if previous_block_hash != new_block_hash
        {
                Return False
        }
        else if new_block.Hash() > target
        {
                Return False
        }
        else
        {
                Return True
        }
```

Figure 7. Block validation algorithm.

3.6. Workload size and data distribution

Workload size will be determined by the capabilities of the miner as well as its availability. The Active manager will distribute work in accordance with a given node's computational ability thus maximizing the effectiveness of the network and reducing the instances where a node receives more work than it can process in a reasonable period of time.

The active manager is responsible for transmitting data such as the transaction hash, as well as nonce values to miners as shown in Figure 2. If there are n miners active on the network the manager will distribute n distinct nonce sets. It is important that no two distributed nonce sets share any values. When any node has depleted its set of nonce values it will send a request for a new set to the management team and the active manager will distribute a set dependent upon the node's capacity. High throughput nodes will receive larger sets vs smaller nodes will receive smaller workloads. New miners joining the network will receive data (the hash) and a set of nonce values to work with.

3.7. Transaction speed

The goal of implementing parallel mining is to dramatically improve the transaction and verification times as well as increase the overall scalability of the network. Using a parallel mining approach, miners can quickly reach consensus and verify transactions efficiently. This increase in efficiency and reduction in transaction times will improve the user experience and create a transactional environment that users can come to rely upon. In contrast to solo mining parallel mining provides a significant improvement in transaction speed and throughput. Examples of this improvement is seen in the implementation of mining pools as well as previous work on parallel mining [3, 35]. Later in this thesis we will discuss some data comparing solo mining to parallel mining for more details.

3.8. Fees

Transaction fees are used on many blockchains, the most popular of which is the Ethereum blockchain and its use of gas to pay transaction fees. Fees can easily be incorporated into parallel mining, but it is important to note that fees should be scaled in relation to the transaction size. Ethereum has some drawbacks when processing small transactions as the fees can cost more than the transaction itself in some cases [41]. This is counterproductive to encouraging users to utilize a network so ensuring that fees remain affordable and proportional to the transaction size is crucial to maintain usability. A reasonable service fee would be somewhere between 1% and 2% of the overall transaction but there must also be an upper and lower bound to ensure that no customer is charged an unreasonable transaction fee. Say for example if a transaction is an exchange of $100,000,000 or 1 cent a percentage service fee will not make the transaction viable on the network and customers will go elsewhere to process their transactions. Thus, keeping transactions affordable is a key element to the overall success of the network.

3.9. Scalability

Parallel mining of proof of work is quite scalable as the more users using the network the more miners become a part of the network. Users can also opt to connect small computational devices to a single account making use of the internet of things to contribute to the blockchain and be rewarded based on the contribution utilized. If the network grows in size to the point that a single hardware node can no longer support service as the Active Manager, the Support manager pool may elect to elect multiple nodes as the Active manager and in this case the nodes will work in parallel to perform the work required of the Active Manager role. This creation of a virtual or composite node should extend the scalability of the system indefinitely as the more nodes are added to the network the more powerful and capable the management team will become. The algorithms required for this kind of dynamic scaling can be developed in future work.

3.10. Parallel mining compared to pool mining

Table 3. Mining Pools Compared to Parallel Mining.

Attribute	**Parallel Mining**	**Mining Pools**
Centralization	Parallel mining is decentralized with mangers rotating with each epoch keeping decentralization intact.	Pool mining by nature has a central authority making decentralization impossible. The central authority is responsible for distribution of work and payment to all workers contributing to the pool.
Rewards	The rewards are split among participants based on their contributions to the network.	The rewards are split among participants based on their contribution and policies of the pool.
Pool fee	There is no fee to be a working member of the network.	Depending on the pool there may be a membership or participation fee as well as other charges.
Difficulty target	The difficulty target in parallel mining will be the same as the target in the mainstream.	The difficulty target assigned within in a mining pool is normally lower than the target of the main Blockchain stream.
Responsibilities of manager	The managers responsibilities include distribution of transaction hashes and nonce ranges, as well as selection of new managers within the network.	Mining pool managers monitor and control everything within a mining pool and act as a central authority.
Contribution to the network	A node's contribution to the network is independent of all peers and differs based on its resources.	A node's contribution is assigned based on the resources it can contribute to the pool.

Table 3 shows how parallel mining compares to pool mining. While pool mining is slightly more resource efficient as only one node serves as a manager it is vulnerable to attacks and introduces centralization as the pool manager is not a free part of the network like those used in parallel mining but controlled by the mining pool administrators. As such mining pools do away with the concept of decentralization. In contrast parallel mining maintains decentralization while making use of a distributed workforce to faster solve proof of work.

3.10.1. How the presented solution will address the tragedy of commons problem

The tragedy of commons will be addressed by making even small computational devices capable of meaningful contribution to the network. This will ensure that there will always be a surplus of miners and so long as all miners that are being utilized receive a fair share of the reward based on the amount of work performed there will be a reason to keep miners available to the network to provide services and collect the rewards of their contribution.

3.10.2. Integration with the cloud

Cloud resources work well with the presented parallel approach as nodes hosted on the cloud can contribute their resources when the resources are not being utilized and by setting a small workload size, they can request packets of nonce values that can be small enough for them to contribute while still being able to quickly transition to other work as required.

3.10.3. Integration with the internet of things

One of the most novel characteristics of this parallel decentralized blockchain network is its ability to integrate with the internet of things; allowing small devices to contribute their computational power to processing transactions on the network. This could dramatically change the way cryptocurrencies are used as transaction validation times could be reduced and transactions would be processed quite quickly.

3.10.4. Node virtualization

Smaller nodes can be combined dynamically by the management team to produce virtual nodes that fill the requirements of the system by allotting resources from smaller nodes to work together as a single node. This will in theory allow computational devices with limited abilities to contribute to the network by joining together as a single virtual node [40]. Most likely this will provide the most utility when the network has reached a large size and single management nodes can no longer manage the large number of nodes in the network. By increasing the capability of the managers by merging node resources we can essentially create super computers using the collective capabilities of smaller nodes serving as a single node. This will allow the network to infinity scale as new nodes are added to the system.

3.11. Rewards

Block rewards will be distributed by the management team's consensus based on the miner's contribution to the block with payments being distributed each epoch. The collective of manager nodes will distribute the block reward based on the work performed by each miner so long as they made a meaningful contribution to the processing and validation of the block. Transaction fees will also be distributed based on the level of involvement a particular miner contributed to possessing the block. The most basic approach would be to sum up the transaction fees and add them to the block reward for distribution. Both the fees and block reward will benefit from being dynamically adjustable to ensure stability of the network and prevent inflation or scarcity issues that may arise in the future to unforeseen events.

3.12. System events

3.12.1. Multiple nodes solving the hash simultaneously

If multiple nodes solve the hash at the same time the first solution received by the active manager will be considered the first to be completed and be moved on to the block validation process. There is also the option of following traditional blockchain approaches and let forks occur and prune them after a set number of blocks choosing the longest chain as the valid path.

This approach has been deemed effective and used in most mainstream blockchains in use at the time of this printing.

3.12.2. Nodes entering the network

New nodes entering the network will make a request to the management team and will receive an updated dataset to ensure that the new node is concurrent with the current state of the blockchain. If there are no managers active the new node will receive its data from the other nodes on the network and the process of electing managers will be initialized by the collective.

3.12.3. Nodes leaving the network

When a node leaves the network, it will have minimal impact on the network as workers can freely leave and if the nonce solution was in its set a new solution will be found using another nonce value. If the leaving node is the active manager, the management team will elect a replacement and the parallel mining will continue without any noticeable impact. A new manager will be added to the manager support pool to replace the manager that was elected to become the new active manager. The number of managers in the pool will be dynamically scaled in proportion to the total size of the network and optimal redundancy requirements.

3.12.4. A miner requests a new nonce range before completing a range

This is a highly unlikely situation and indicates a flaw in the operation of the miner. This will be counterproductive as the solution may be in one of the skipped values. While this will not cause damage to the networks functionality due to there being multiple possible solutions it will reduce the nodes chances of solving the puzzle thus, it is very unlikely that nodes would be altered to cause such a behavior.

3.12.5. The active manager goes offline and a manager from the backup manager pool has found the solution to the hash

Manager candidates will not be permitted to be elected if the manager in question has submitted a hash solution to the management pool in the absence of an active manager. The new active manager will be elected and the manager that found the solution will be treated as a normal miner until the block has been finalized.

3.13. Possible concerns

Coin value will undoubtably become a topic of concern as the cryptocurrency becomes popular and things like smart contracts become a common practice on the network. We must ensure that we have a built-in method to burn surplus coins if too many are generated as well as a method to enforce a soft cap as to the number of coins that can be produced. This can be done by dynamically adjusting the transaction fees and burning a percentage of the coins when needed to ensure that inflation does not become a problem and the market is never flooded with coins. Also, in the case that there are too few coins in circulation for whatever reason there should be a method to increase the block reward to encourage the miner's contribution to the network. This dynamic system will need the ability to adjust the number of coins in circulation without presenting a security vulnerability or method of exploitation. The management team will be responsible for enforcing these constrains and ensuring a stable coin value relative to global currency transactions taking place to ensure that fees are constantly affordable, and mining is feasible. Bitcoin and most other current cryptocurrencies experience extreme value fluctuations which can deter customers from wanting to use them as the price of an item may change dramatically in a short period of time. For example, at the time of this writing Dogecoin is worth about $0.57 USD and if a customer purchased a candy for two Dogecoin then the next day the

Dogecoin was worth $0.30 the seller would not be satisfied with the transaction. Therefore, it is crucial that there be built in methods to ensure the relative stability of the price.

4. EXPERIMENT AND RESULTS

In this section experiments have been conducted on several test environments including both physical and cloud-based systems. The intended goals of these benchmarks are to illustrate the advantages of parallel proof of work using multiple manager nodes. To achieve this, we will start with the environment setup and network communication using SSH to establish a peer-to-peer network. The proposed modified star network was modeled using MPI communication via SSH. To establish the benefit of adding nodes to the network a benchmark program was made using the SHA-256 hashing algorithm to hash simple messages and the timer function was used to track the number of hashes that can be completed per second; by executing a set number of hashes and timing how long it takes to complete them with various numbers of nodes active as miners in the network. Several benchmarks will be implemented using the Go programming language and compared to that used in the work of Shihab Shahriar Hazari and Qusay H. Mahmoud. Failures will be introduced, and block time will be measured using a timer to determine the changes in block time when manager nodes fail. These results will show the benefit of manager redundancy and the impact of a failed manager node on the parallel system. In the case of the proposed approach the backup managers will be set up to take over the active managers role in the event that a miner requests a nonce set, and the active manager fails to notify the management team that it has filled the request. In a real-world implementation, there will be a much more complicated manager election process that will fill this role which was not implemented during the testing process. This election process can be simulated by requiring the incoming manager to perform a small amount of work before taking over to better represent the time taken to elect a new manager, and to ensure that the Active manager actually needs to be replaced.

4.1. Environment

This thesis research made use of several test environments including a physical Linux cluster, a cloud cluster, and Raspberry Pi to collect benchmarks on parallel proof of work. The environments will be discussed in the following subsections in detail.

4.1.1. Physical environment

Local benchmarks were taken by setting up a parallel computing environment consisting of 8 Linux machines running Ubuntu version 16.4 LTS. These machines were connected via ethernet cables and a switch with static IP addresses assigned to each machine to create a Linux cluster. Table 4 displays the environment specifications used to produce the benchmarks computed on the network. The network itself utilized password-less communication via ssh using stored key value pairs to connect 8 machines to form a Linux cluster on a LAN. The machines were connected via ethernet cables routed through a switch. The host files were edited to facilitate communication and NTFS was added to allow the machines to share programs across the LAN. Most of the benchmarks were executed using a Sha-256 hashing algorithm (Secure Hashing Algorithm 256). The reason that this hashing algorithm was chosen over others is that it is used in Bitcoin mining and the hash value is restricted in size. What this means is that for any given input message the output hash value will be 256 bits in length. This feature will greatly improve data storage capacity when the messages become large. The tests conducted on the Raspberry Pi platform were made utilizing the Blake2 hash within the random-x hashing algorithm used in mining Monero. This thesis is focused on the increase in hashing capabilities provided with parallel proof of work so less focus will be given to the particularities of the hash functions themselves as the concept of parallel proof of work with multiple manager redundancy can be implemented with any blockchain that utilizes proof of work regardless of the hash function used. Sha-256 is a good starting point due to its use in the Bitcoin network and efficient data storage capabilities.

Table 4. Environment Specifications

Architecture:	x86_64
CPU op-mode(s):	32-bit, 64-bit
Byte Order:	Little Endian
CPU(s) Per node:	4
Thread(s) per core: 1	1
Core(s) per socket:	4
Socket(s):	1
NUMA node(s):	1
Vendor ID:	GenuineIntel
CPU family:	6
Model:	60
Model name:	Intel(R) Core(TM) i5-4570S CPU @ 2.90GHz
Stepping:	3
CPU MHz:	3303.323
CPU max MHz:	3600
CPU min MHz:	800
BogoMIPS:	5786.89
Virtualization:	VT-x
L1d cache:	32K
L1i cache:	32K
L2 cache:	256K
L3 cache:	6144K

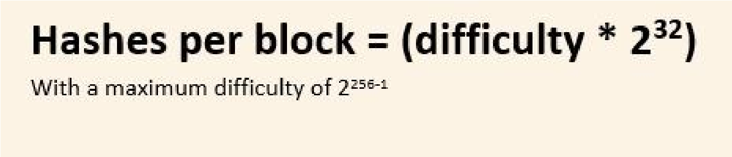
4.1.2. Cloud environment

A cloud environment was also deployed using both Microsoft azure as well as Google Cloud Platform. Both deployments contained 8 nodes with 2 cores each. A virtual network was set up on Azure and communication between nodes was conducted via SSH. The performance of the two providers did not show any substantial differences between the two providers.

Benchmarks were taken on the cloud environments to obtain block times as well as the possible hashes per second on the network with differing node counts.

4.2. Benchmarks

Hash difficulty refers to the number of leading consecutive zeros of an acceptable hash. The greater the number the more difficult the hash is to solve and by extension the more work is required to solve it. Running a benchmark with a difficulty of 1 will be solved significantly faster than the same input with a hash difficulty of 10. The average time taken to solve a hash in seconds is used to measure the performance of the network. The average is calculated by measuring the time taken to solve a block a set number of times than dividing the sum of all times by the number of blocks solved. Hashes per second are calculated by timing the number of seconds taken to solve a block then dividing that time by the number of nonce values used to find the solution to the block. Optionally hashes per second can be measured by starting a timer, preforming a set number of hash attempts then stopping the timer and dividing the number of hashes executed by the number of seconds the timer has run. When measuring hashes per second we will see that different hash algorithms produce differing results and the hash difficulty IE the number of leading zeros will also have a significant effect on the hash speed of the benchmarks. Many CPU and GPU manufactures will opt to post hashes per second for their hardware that is excessively high in comparison to the hashes that will be seen when mining a crypto currency. For example, the Raspberry Pi is said to be able to produce an average of 108 hashes per second but at operational difficulty levels it only manages an average of 2.3 hashes per second as seen in Table 5. The formula to determine the average number of hashes required to solve a block in the Bitcoin network can be seen in Figure 8.



Hashes per block = (difficulty * 2^{32})

With a maximum difficulty of 2^{256-1}

Figure 8. Average hashes required to solve a block in the Bitcoin network.

As of the time of this writing Bitcoins hash difficulty ranges significantly higher than that capable of being supported using CPU mining. The difficulty adjustment is directly related to the total mining power estimated by the Total Hash Rate (TH/s) chart [42]. This means that the hash difficulty is dynamically scaling to become more difficult over time. With our proposed approach the difficulty will also scale based on the networks capabilities to ensure against forking attacks, but care will be taken to ensure that the difficulty never exceeds the networks' ability to efficiently handle transactions. The difficulty used in our calculations refers to the number of leading zeroes the resulting hashed value must have to be considered a valid solution.

Benchmarks for hashes per second were implemented using a difficulty level of 1 and the chrono library in C++ using mpich. Network communications used were broadcast, send and receive. Between nodes serving the designated roles, all roles were hard coded and dynamic node scaling was not implemented at the time of benchmarking. Other benchmarks were collected using the time library of the Go language with network communications provided by the go-libp2p library. Code relating to the benchmarking process can be found at [43].

Solo mining results in the speed of the fastest node being the average as the fastest node will always solve the hash before the others and the work of the others is wasted except during the validation step. During solo mining all the nodes are competing against one another to solve the hash before the others and only the fastest node will receive the reward for solving the proof of work.

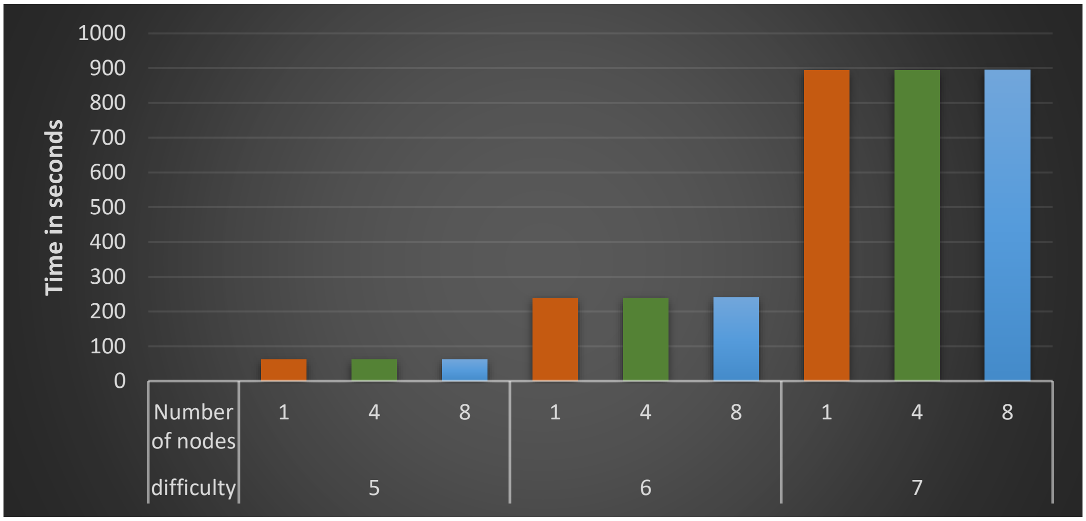


Figure 9. Solo mining time in seconds to complete a hash.

Figure 9 shows hashing times of a set of nodes solo mining. Note that the solution times are relatively the same regardless of the number of nodes as any advantage gained by introducing new nodes is only going to be visible if the new node has more computational ability than the others on the network. When the computational ability of the nodes is identical the hash times will be very close with minor deviations as other tasks run in the background. After averaging the runs, we get consistent results with no added benefit from the addition of more nodes.

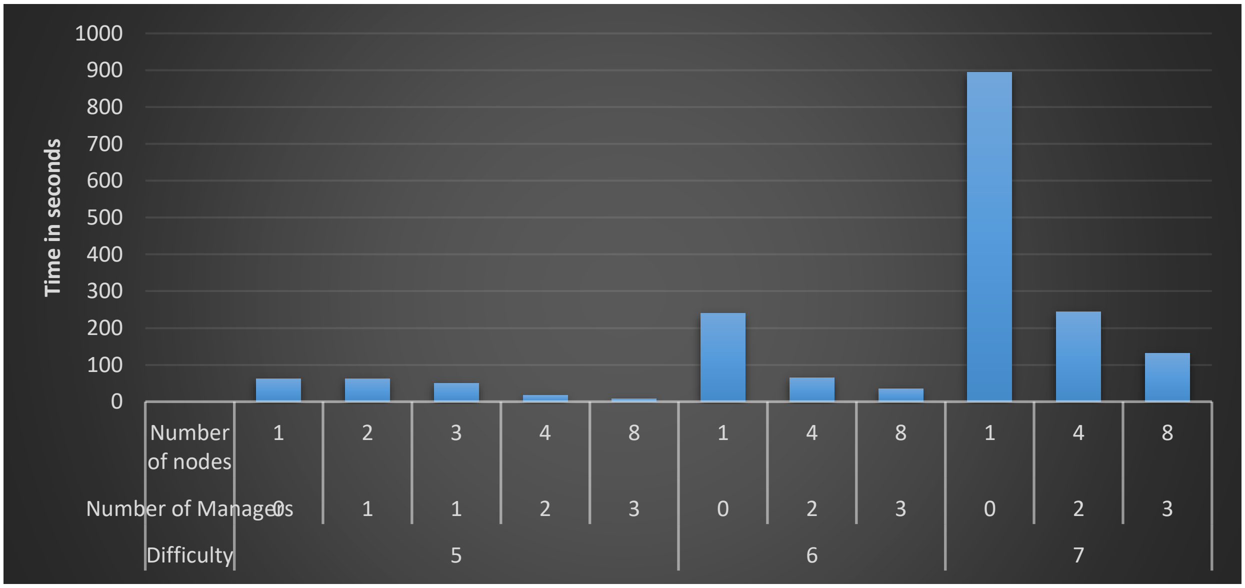


Figure 10. Parallel mining with managers in seconds to completion.

Figure 10 displays the benefit of additional nodes when using parallel proof of work. As seen in the figure the addition of new nodes has a visible effect on the hashing power of the network and drastically reduces the time taken to solve a block especially in the case of higher difficulty hashes. With a difficulty of 4 and below there is a relatively low amount of variance in the time taken to solve the hash due to its simplicity but as the difficulty increases the advantage of having more nodes begins to become greater. Once we reach a difficulty of 7 the benefit of parallel mining becomes obvious. There is little difference in hashing speeds with the addition of additional backup managers as the backup managers can still contribute to the mining process so long as they are not the designated active manager. If a manager finds the solution and the active manager is offline the management team will not permit the manager that found the solution to be elected as the active manager.

Figure 10 shows some interesting data relating to the hash difficulty and the number of nodes. Where the hash difficulty is low the time reduction the system sees is much lower than when the hash difficulty is increased. What this indicates is that in the presence of a large workforce the miners may become underutilized and if the nonce values are over spread the communication times may rise higher than the performance gained by dividing the work. Thus, Managers will need to monitor the work to worker ratio and divide the work accordingly leaving some workers idle if necessary. Idle workers will not be consuming the same power levels as working nodes thus this will result in energy savings across the network. As shown in Figure 10 networks with less than 3 nodes see no benefit from the addition of managers but any node count above 2 will benefit from additional nodes as even the manager nodes can dynamically scale their role back and serve as a miner when up to date creating added service to the network.

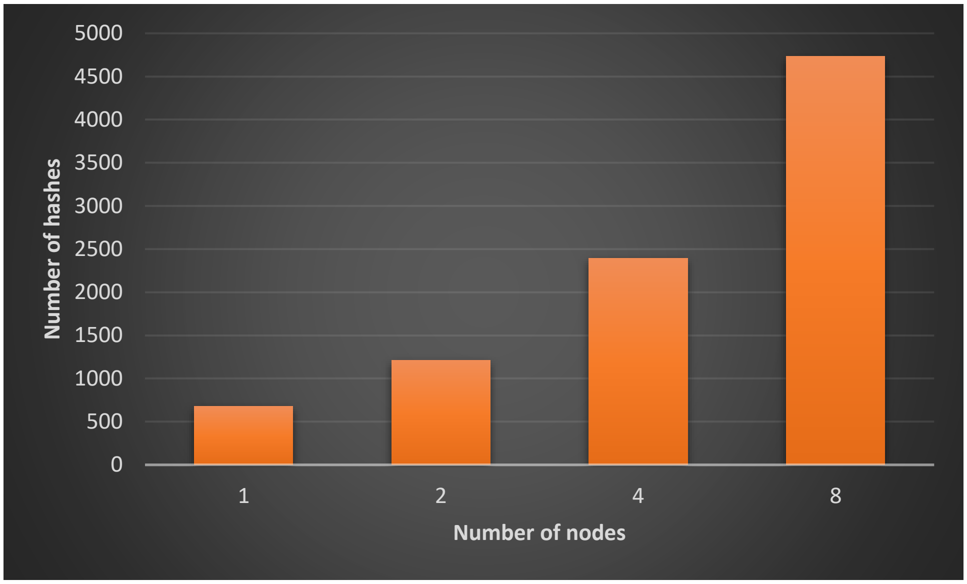


Figure 11. Hashes per second with introduction of additional nodes.

Figure 11 displays hashes per second that can be computed with the addition of more nodes to the network. As additional nodes are added the computational capability of the network increases allowing the network to compute higher numbers of hashes as the node count increases. Note that the increase in hashes per second does not double when the node count is doubled; this is due to communication overhead required for communication between nodes across the network. Next, we will see how the data in Figure 11 contrasts with that in Figure 12 where the blocks difficulty is also considered.

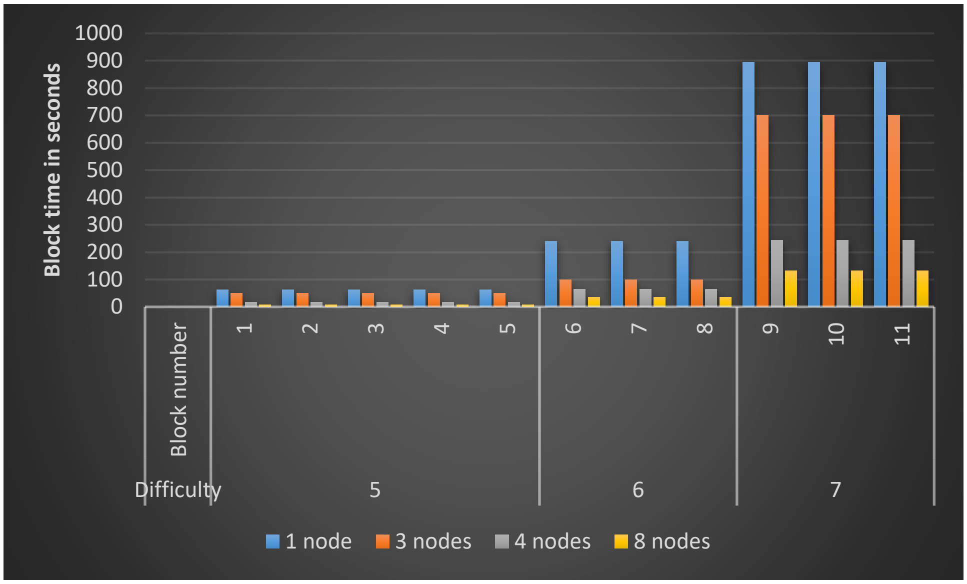


Figure 12. Block time with increasing hash difficulty.

Figure 12 shows block times in seconds as the difficulty increases with the lines indicating networks with differing node counts. Longer time periods to process a block are not ideal and we are aiming to achieve the lowest time possible to complete the block as with the use of higher difficulty levels the time period will increase exponentially. As we saw in Figure 11 the network with 8 nodes has the highest rate therefore produces the block in a fraction of the time required for the single node solo mining which is represented as the blue bars where the 8-node network parallel mining is the yellow bars in Figure 12.

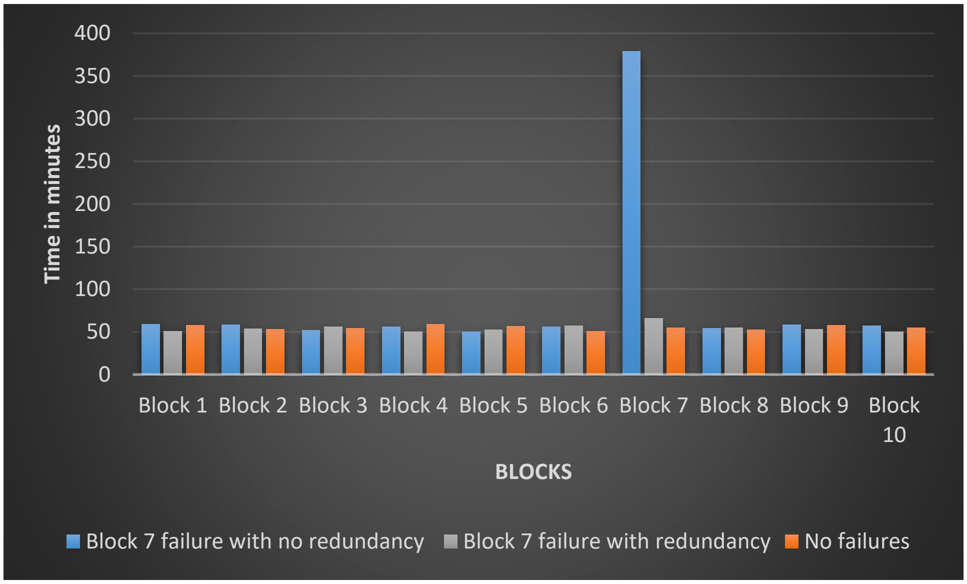


Figure 13. Block time comparison with node failures.

Figure 13 illustrates the impact of manager redundancy during the event of a node failure and network recovery. Note that the failure is only affecting block 7 and all other blocks are consistent. The difficulty level for this test was set at 10 leading zeros for an acceptable hash value. When the model running the Hazari- Mahmoud model with a single manager node represented by the blue line in our graph experiences a failure of the manager node parallel hashing stops and for the remainder of the block the network will perform at solo mining speeds. While in the case of the presented approach the failed manager is replaced by a supporting manager and while the supporting manager is no longer able to contribute to mining hashes the parallel work continues through the remainder of the block causing only a small increase in time taken to process the block. Thus, in the rare event of a network attack or node failure the presented approach provides a more robust and effective solution while introducing marginally higher communication costs to ensure network reliability.

4.2.1. Raspberry Pi tests

The Raspberry Pi is a good testing unit to consider when talking about the internet of things. If our network is to be connected to small devices, we should gather a baseline of what these devices are capable of in terms of hashes per second. Tests on the Raspberry Pi platform were done using the Raspberry Pi 4 with a Sandisk 32GB microSD card using the Raspberry Pi OS with Desktop. Heatsinks were added to aid in cooling the chips. The Raspberry pi was chosen because it has a low computational capability, and our aim is to develop a system where such devices being part of the internet of things may contribute in a meaningful way to the overall network. As we see in Table 7 the Raspberry Pi was not very capable when mining by itself resulting in an average hash rate of 2.3. With proper dynamic management and enough contributors, a smart network will be capable of sustaining a decentralized parallel mining system. One node by itself may not be very useful but together many of them could provide a scalable decentralized smart network to facilitate transactions true to Satoshi's vision of cryptocurrency [7].

Table 5. Mining Results for a Single Raspberry Pi 4

CPU type:	Raspberry Pi 4 – Arm Cortex-A72
Coin type:	Monero
Time:	8 Hours
Difficulty:	177,307,724,796
Hashes per Second:	1 – 7
Average hashes per second	2.3
Blocks:	o
Bad shares:	1
Invalid shares:	31
Good Shares:	357
Total mined:	0.000001410642

Table 5 shows the Raspberry Pi 4's mining capability when mining Monero as a member of a mining pool. Note that when mining in a pool the hash difficulty was at 177,307,724,796 which is an extremely com difficulty thus resulting in exceptionally low hashes per second. These kinds of difficulty levels are common in the normal operation of a cryptocurrency but rarely used when benchmarking as most CPU benchmark tests aim to achieve the highest results possible without regard to real world load. Monero is one of the most used CPU mined cryptocurrencies and uses the RandomX hashing algorithm.

4.2.2. Scalability of distributed work

As shown by increasing the number of worker nodes the work becomes easier. There will be a time when the number of available miners is higher than that optimally required to compute calculations most efficiently. Thus, managers must ensure that the nonce values being distributed are not below a set size in comparison to the number and capabilities of the miners. If the nonce sets are too small the communication overhead could be higher than any gain achieved by splitting the work resulting in a loss of efficiency. The managers should dynamically monitor the networks condition to ensure that miners are not used unless needed to maintain the best energy and network efficiency. With new nodes joining the network available for mining if needed, the system should be dynamically scalable and can easily support the needs of the users with performance increasing as the number of users increases.

Table 6. Hashes Per Second in a Cloud Environment

Number of nodes	Hashes per second
1	339.00
2	670.54
3	1005.47
4	1326.30
5	1659.87
6	1994.06
7	2329.66
8	2653.34

Table 6 shows the scaling of hashing capability as new nodes are added to a cloud environment. With the addition of nodes, the collective hashing capability of the network will increase as will the difficulty required to solve the hashes. This will reduce the likelihood that a 51% attack could occur as there will be little chance of a single entity gaining such a dominant foothold in a large network. This approach supports both decentralization and dynamic network scaling.

4.2.3. Concerns and limitations

The small scale of the experiments conducted leave room for further testing and a viable real-world application based on this architecture will require integration with additional node types such as wallet nodes. Some concerns may arise because of decreased revenue on the part of the miners as single nodes will not receive a large income as in traditional mining operations but the reduction in system requirements needed to contribute to mining a block will allow for average users to take the place of dedicated mining rigs. This should be offset by a reduction of transaction fees as less specialized resources will be needed to handle transactions. This is intended to produce an internet-of-things based transaction system in place of a dedicated infrastructure as needed with current crypto currency mining.

5. CONCLUSION

In conclusion cryptocurrencies are a growing technology and have great potential to leave a lasting mark on civilization. Integrating a decentralized cryptocurrency into both the cloud and internet of things will provide great scalability and accessibility to the blockchain. This is achievable by integrating parallel computing into a decentralized blockchain protocol to create a smart network. Parallel computing has many advantages to offer and its integration into blockchain technology will increase the benefits of the distributed transactional system. A truly decentralized transaction system will benefit from parallel application over the cloud and across the internet of things. The computational resources required to run blockchain technology can be dramatically reduced and confirmation times will improve with the addition of new nodes into the network. The scalability of this system is substantially superior compared to traditional blockchains and the presented approach solves several security vulnerabilities present in traditional blockchain applications. Most notably the 51% attack as it will be extremely difficult for a single party to gain a majority share in the network, especially if the network is globalized and integrated into the internet of things. Overall power consumption of transactional systems can be reduced as there will no longer be a need for specialized mining operations using large amounts of electrical power. By harnessing the internet of things transactions can be exchanged freely over a smart network in a decentralized manner just as Satoshi envisioned all those years ago [7].

Further research should be conducted to develop effective algorithms to dynamically expand the management team using both node virtualization and ratio protocols that can ensure that the managers can optimally serve their roles as well as detect various attacks. Procedures should also be studied and developed to address anomalies within the network as undoubtably they may arise and will need to be handled appropriately to ensure that the system is both effective and user friendly. Node virtualization techniques should be analyzed and implemented to allow for system scalability and integration to the cloud and the Internet of Things. Another area for improvement and expansion is the manager roles moderation of the distributed work to ensure optimality. The management team should rely on protocols to ensure that both the optimal number of working miners, as well as the optimal number of managers the system requires are utilized to prevent over distribution of data and possible wasting of resources. Implementation of such protocols will address any concerns with the network becoming overly redundant in terms of managers which could introduce scalability limitations. Future research will be required to ensure price stability of coins minted by the network. A system should be developed using machine learning to monitor the coins relative value as well as the total market cap and coins in circulation; and when necessary, burn or increase the number of coins minted to stabilize the value. This will require a great deal of research and is beyond the scope of this thesis but is a topic needing future research. Without stability cryptocurrencies can never hope to replace fiat currencies. A recommendation for future research would be to develop a protocol that would ensure that the coins value be tied to that of a precious metal or an average of global fiat currencies. Although if the currency truly aims to replace fiat money it should not be tied to any fiat currency. Therefore, precious metals seem to be the most logical baseline to tie the currency to if it is intended to become a global currency. The presented approach makes parallel proof of work models far more resistant to attacks both improving the network performance and recoverability in the rare event of network attacks. Machine learning can also benefit blockchain technologies and integration of machine learning into the manager pool could be a key method to ensuring that the model remain truly decentralized and capable of effectively detecting and reacting to new developing threats the networks operation.

REFERANCES

[1] N. Marinoff. “The Bitcoin Price Drop May Have Been Caused By a Power Outage,” Live Bitcoin News, 20-Apr-2021. [Online]. Available: https://www.livebitcoinnews.com/the-bitcoin-price-drop-may-have-been-caused-by-a-power-outage/

[2] C. Baraniuk. "Bitcoin's global energy use 'equals Switzerland'". BBC News, 03-Jul-2019. [Online]. Available: https://www.bbc.com/news/technology-48853230

[3] S. Hazari and Q. Mahmoud. “Improving Transaction Speed and Scalability of Blockchain Systems via Parallel Proof of Work.” Future internet 12.8 (2020): 125–. Web.

[4] A. Hern. "US seizes $1bn in bitcoin linked to Silk Road site". The Guardian. 6 November 2020. [Online]. Available: https://www.theguardian.com/technology/2020/nov/06/us-seizes-1bn-in-bitcoin-linked-to-silk-road-site

[5] J. Douceur. "The Sybil Attack". Peer-to-Peer Systems. Lecture Notes in Computer Science. 2429. pp. 251–60. doi:10.1007/3-540-45748-8_24. ISBN 978-3-540-44179-3. 2002.

[6] O. Beigel. “What is the Bitcoin Mempool? A Beginner’s Explanation” 99 Bitcoins. 2021, 15-January-2021. [Online] Available: https://99bitcoins.com/bitcoin/mempool/

[7] S. Nakamoto. “Bitcoin: A Peer-To-Peer Electronic Cash System”. Bitcoin.org. Bitcoin Project 2009-2021. [Online]. Available online: https://bitcoin.org/bitcoin.pdf

[8] M. Crosby, P. Nachiappan, S. Pattanayak, Verma, V. Kalyanaraman. “Blockchain Technology”. In Sutardja Center for Renessereneurship & Technology Technical Report. Sutardja Center for Entrepreneurship & Technology: Berkeley, UC, USA, 16 October 2015.

[9] J. Bartlett. "Dark net markets: the eBay of drug dealing" The Guardian. 5 October 2014. [Online] Available: https://www.theguardian.com/society/2014/oct/05/dark-net-markets-drugs-dealing-ebay

[10] S. Hazari. H. Qusay. “A parallel proof of work to improve transaction speed and scalability in blockchain systems”. In Proceedings of the IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 916–921.

[11] S. Hazari. “Design and Development of a Parallel Proof of Work for Permissionless Blockchain Systems”. Master’s Thesis, Ontario Tech University, Oshawa, ON, Canada, 2019.

[12] H. Hasanova, U. Baek, M. Shin, K. Cho, M. Kim. “A survey on blockchain cybersecurity vulnerabilities and possible countermeasures”. International Journal of Network Management, 29(2), e2060–n/a. [Online]. Available: https://doi.org/10.1002/nem.2060

[13] L. Cong. “Decentralized Mining in Centralized Pools”. National Bureau of Economic Research, 2019.

[14] Visa, INC. “Visa Fact Sheet”, [Onling]. Available: https://usa.visa.com/dam/VCOM/global/about-visa/documents/visa-fact-sheet-april-2019.pdf

[15] M. Scherer. “Performance and Scalability of Blockchain Networks and Smart Contracts”. Umea University. 2017 [Online]. Available: https://www.diva-portal.org/smash/get/diva2:1111497/FULLTEXT01.pdf.High-level

[16] “Cryptocurrency Transaction Speeds: The Complete Review”. The Daily Hodl. 2018. [Online]. Available: https://dailyhodl.com/2018/04/27/cryptocurrency-transaction-speeds-the-complete-review

[17] H. Agrawal. “Top 10 Cryptocurrencies With Fast Transaction Speeds”. CoinSutra - Bitcoin Community. 5-September-2019. [Online]. Available: https://coinsutra.com/transaction-speeds/

[18] Kracken. “Cryptocurrency deposit processing times”. Kraken.Com. 12-May-2021. [Online]. Available: https://support.kraken.com/hc/en-us/articles/203325283-Cryptocurrency-deposit-processing-times

[19] Qwahzi. “Nano Stress Tests - Measuring BPS, CPS, & TPS in the real world”. Nano Forum. 2-April-2021. [Online]. Available: https://forum.nano.org/t/nano-stress-tests-measuring-bps-cps-tps-in-the-real-world/436

[20] L. Lann. “Distributed Systems-Towards a Formal Approach”. IFIP Congress. 1977, 7, 155–160.

[21] L. Kenny. “The Blockchain Scalability Problem & the Race for Visa-Like Transaction Speed”. Towards data science. 23-July-2019. [Online]. Available: https://towardsdatascience.com/the-blockchain-scalability-problem-the-race-for-visa-like-transaction-speed-5cce48f9d44

[22] P. Crossman. "How will SEC complaint affect banks' relationships with Ripple?". American Banker. 29-December-2020. [Online]. Available: https://www.americanbanker.com/news/how-will-sec-complaint-affect-banks-relationships-with-ripple

[23] A. Vries. "Bitcoin's Growing Energy Problem". Joule, Volume 2, Issue 5. 16-April-2018. ISSN 2542-4351. [Online]. Available: https://doi.org/10.1016/j.joule.2018.04.016

[24] K. Miles. “The Little Coin That Ate Quebec: A Canadian Hydropower Operation Put Out the Welcome Mat for Bitcoin Miners. Shortly Thereafter, It Was Overrun.” Technology review. 1-May-2018. Vol.121 (3), p.33.

[25] P. Roberts. "This Is What Happens When Bitcoin Miners Take Over Your Town - Eastern Washington had cheap power and tons of space. Then the suitcases of cash started arriving". Politico. 9 March 2018.

[26] "Cryptocurrencies and blockchain" (PDF). European Parliament. July 2018. Retrieved 29 October 2020. [Online]. Available: https://blog.elitex.ir/wp-content/uploads/2020/06/Cryptocurrencies-and-Blockchain.pdf

[27] C. Dwork, M. Naor. "Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology". CRYPTO'92: Lecture Notes in Computer Science No. 740. Springer: 139–147. doi:10.1007/3-540-48071-4_10. 1993.

[28] M. Jakobsson, A. Juels. "Proofs of Work and Bread Pudding Protocols". Secure Information Networks: Communications and Multimedia Security. Kluwer Academic Publishers: 258–272. doi:10.1007/978-0-387-35568-9_18. 1999.

[29] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu and R. Brooks, "A brief survey of Cryptocurrency systems," 2016 14th Annual Conference on Privacy, Security and Trust (PST), 2016, pp. 745-752, doi: 10.1109/PST.2016.7906988. [Online]. Available online:
https://ieeexplore.ieee.org/abstract/document/7906988?casa_token=uFCnD-

4qj4AAAAAA:ahlEijVlsythhFCzyClu7yebJfXCsBj8LWR76zxVgKo9tjQsdXojVP1eSP4Eov4Sgno4gg2x6A

[30] P. Tasca, C. Tessone. "A Taxonomy of Blockchain Technologies: Principles of Identification and Classification". Ledger. 4. doi:10.5195/ledger.2019.140. ISSN 2379-5980. 15-February-2019.

[31] S. Haig. "Ethereum miners plot hash-power 'show of force' against EIP-1559". Cointelegraph. 11-March-2021. [Online]. Available: https://cointelegraph.com/news/ethereum-miners-plot-hash-power-show-of-force-against-eip-1559

[32] C. Jepson. "DTB001: Decred Technical Brief." [Online]. Available at https://www.cryptoground.com/storage/files/1527488958_dtb001.pdf Additional information available at https://www.decred.org (2015).

[33] Decred. "Decred Documentation: Block Production Times." Decred.org. 15-March-2021. [Online]. Available: https://docs.decred.org/research/block-production-times/

[34] C. Dwork, N. Lynch, L. Stockmeyer. "Consensus in the presence of partial synchrony". Journal of the ACM, 1988-04-01, Vol.35 (2), p.288–323.

[35] I. Eyal, A. Gencer, E. Sirer, R. Renesse. "Bitcoin-NG: A Scalable Blockchain Protocol". In Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI '16), Santa Clara, CA, USA, 16–18 March 2016; pp. 45–59.

[36] X. Boyen, C. Carr, T. Haines. "Blockchain-Free Cryptocurrencies. A Rational Framework for Truly Decentralized Fast Transactions". IACR Cryptol. ePrint Arch. 2016, 2016, 871.

[37] S. Popov. “The Tangle”. White paper 1, no. 3 .2018. [Online]. Available: http://www.descryptions.com/Iota.pdf

[38] A. Tulic. “How Do Mining Pools Work? Is It Better Than Solo Mining?”. Captainaltcoin. 2018. [Online]. Available: https://captainaltcoin.com/what-is-pool-mining

[39] Golang libp2p. Available online: https://github.com/libp2p/go-libp2p

[40] T. Kawakami. “A Node Virtualization Scheme for Structured Overlay Networks Based on Multiple Different Time Intervals”. Applied Sciences, 10(8596), 8596. 2020. [Online] Available: https://doi.org/10.3390/app10238596

[41] D. Carl, C. Ewerhart, “Ethereum Gas Price Statistics”. University of Zurich, Department of Economics, Working Paper No. 373, 22-December-2020, [Online]. Available: https://ssrn.com/abstract=3754217 or http://dx.doi.org/10.2139/ssrn.3754217

[42] Blockchain.com. “Network Difficulty,” Blockchain.com. [Online]. Available: https://www.blockchain.com/charts/difficulty

[43] J. DeNio. “Benchmarking Code for Parallel Mining”. 2021. Available: https://github.com/SliverOverlord/Masters_Paper