Anchoring the Value of Cryptocurrency

Yibin Xu
School of Computer Science
and Informatics
Cardiff University
Cardiff, UK
work@xuyibin.top

Yangyu Huang
School of Electronic Engineering
and Automation
Guilin University of Electronic Technology
Guilin, China
i@hyy0591.me

Jianhua Shao
School of Computer Science
and Informatics
Cardiff University
Cardiff, UK
shaoj@cardiff.ac.uk

***Abstract*—A decade long thrive of cryptocurrency has shown its potential as a source of alternative-finance and the security and the robustness of the underpinning blockchain technology.**

However, most cryptocurrencies fail to show inimitability and their meanings in the real world. As a result, they usually start off as favourites but quickly become the outcasts of the digital asset market.

The blockchain society attempts to anchor the value of cryptocurrency with real values by employing smart contracts and link it with computation resources and the digital-productivity that have value and demands in the real world. But their attempts have some undesirable effects due to a limited number of practical applications. This limitation is caused by the dilemma between high performance and decentralisation (universal joinability). The emerging of blockchain sharding models, however, has offered a possible solution to address this dilemma.

In this paper, we explore a financial model for blockchain sharding that will build an active link between the value of cryptocurrency and computation resources as well as the market and labour behaviours. Our model can adjust the price of resources and the compensation for maintaining a system based on those behaviours. We anchor the value of cryptocurrency by the amount of computation resources participated in and give the cryptocurrency a meaning as the exchange between computation resources globally. Finally, we present a working example which, through financial regularities, regulates the behaviour of anonymous participants, also incents/discourages participation dynamically.

***Index Terms*—Blockchain, alternative finance, financial model, Cryptocurrency**

I. Introduction

Money is used universally, representing a medium of exchange, a unit of account, a store of value, and a standard of deferred payment [1]. Currency is always associated with national identity, safeguarded by the sovereignty, required to be inimitabe and used in daily life by a population [2]. Currency is also a tool and voice in international politics [3]–[5] - a currency is usually priced by the national power and nonrenewable resources that a country holds [3], [5]. Centralised currency policies can incent and exploit labour and productivity, and have coupling effects on market behaviour [6], [7].

When considering a cryptocurrency, people usually have the impression that it has loose regulation, high privacy and fairness [8]–[10]. But people cannot associate the value of a cryptocurrency with any sovereign or nonrenewable resources like they do for a normal currency [11]–[13]. That is, cryptocurrencies do not have a value anchor, are less connected to our daily life, and are not difficult to build or replace in the current means of human economics.

In technical terms, every transaction in a cryptocurrency needs to be broadcast to avoid the double-spend problem in a decentralised environment [14]. However, ordinary people who hold home desktops, laptops or mobile devices cannot be expected to do this constantly, nor be able to be fully synchronised with the environment most of the times [15], [16]. That is, for most people, their devices will not in be in the system all the time and only a very small number of devices compared to the user population will actually act as maintainers [15]. These phenomena make cryptocurrencies unreliable and easily abandoned [17]–[20].

Ethereum [21] and other smart contract approaches [21]–[23] try to form a more decentralised sovereign where "citizens" exchange productivity through smart contracts. However, this approach tends to result in few citizens actually doing jobs on the platform, and because blockchains require all citizens to witness and conduct every job to ensure the integrity of results by consensus, productivity is reduced. Some citizens may be disadvantaged, hence eliminated, if the system does highly sophisticated jobs or an overloaded workflow. Consequently, the performance of these platforms will stay low in order to attract citizens, making smart contract platforms not even being able to perform simple tasks such as powering a game of digital pet smoothly [24].

With the idea of blockchain sharding [25]–[27] which assigns different citizens to do different tasks in parallel securely, it is promising that productivity can be improved and the usage of cryptocurrency can be extended to, for example, exchanging computation power for doing more sophisticated jobs and increasing performance of computation resources. Because transactions are divided into shards, the requirement on citizens' computation ability and network bandwidth is much reduced due to reduced workflows, more citizens can be added to this type of decentralised sovereign, and it makes it easier for data synchronisation. However, digital citizens are different from citizens in the real world. Just like characters in a video game, digital citizens can die and then start again, and can ignore regulations causing damages without punishments [28], [29]. Since computation resources are utilised to fulfill the demand in the real world, the cost of using such resources has meaning in the real world too [30]. With the smart contract approach, cryptocurrency money is earned with real effort and real labour which has a price in reality. Therefore, it should be possible to regulate digital citizens by, for example, confiscating their digital assets if they behave inappropriately, or encouraging citizens to perform jobs by exciting them with compensation. In this way, it is possible for the system to stabilise the performance of participated resources without knowing the source of them in reality or building a credit base identity.

Grid Computing [31] and Cloud Computing [32] are two computational models that utilise resources globally through the Internet. Grid computing attempts to use the idle resources globally and is a multi aid platform with credit-based identity model. Cloud computing, on the other hand, is a model where users buy resource globally owned and priced by a centralised organisation. We see the popularity of cloud computing in both industry and academia [33]–[35]. Cloud platforms have generated considerable revenues [36], and they have changed the way technology industry operates, including the way to host a website or buy computation resources. In contrast, Grid computing is less popular, due to these main difficulties: (1) needs necessary interfaces for data exchange and job execution on different devices and using different software to complete a task. (2) needs to be able to trust job results returned from an anonymous source. (3) needs to regulate participated devices worldwide, without having to know the identity of the owners, and to ensure jobs are delivered correctly and on time.

We believe that blockchains and cryptocurrencies can be exploited to address some of the problems of Grid computing we outlined here. The first challenge can be addressed by the use of smart contracts, which can unify interface of data exchange and job executions. The second issue can be tackled by blockchain, especially, blockchain sharding techniques, where the consensus job results returned from a shard is guaranteed to be correct as long as the security threshold is maintained. Compensations and punishments in a cryptocurrency associated with the blockchain can be employed to address the quality of return. By addressing these issues with blockchain sharding and smart contract techniques, the Grid computing techniques may be improved. However, to function a Grid computing platform and to make it a decentralised sovereign, economic regulation and policy must serve as a tool to adjust and control the balance between digital labour and demand in the market. The problems of inflation, deflation or even economic break down that exist in real world economy can also exist in the digital world. Therefore, an economic model in this type of decentralised sovereign is important to study.

In the remaining of the paper, we will first introduce blockchain, smart contract and blockchain sharding. Then, we discuss an autonomous finance model that use the classical money supply indicators of M0, M1, M2, and the quantity theory of money [37], [38] to price a currency by service demand and to encourage and discourage labour participation by wages. We use linear regression [39] to set the parameters to regulate the market and we provide a simulated experiment at the end of the paper.

II. Preliminaries

A. Blockchain

Proof of Work (PoW) describes a system that is difficult to be created but easy to be verified. The most widely used Proof-of-Work scheme - Hashcash [40], is based on SHA-256 and was later introduced as a part of Bitcoin (Nakamoto blockchain) as the computation strength competition method. There are different kinds of PoW alternatives proposed for blockchain [41], [42].

A block in Nakamoto blockchain embeds the information of a period; the blockchain periodically attaches new blocks to the blockchain. The difficulty is a measure of how difficult it is to generate a PoW.

$$Difficulty = \frac{Difficulty\_1\_target}{Current\_target} \tag{1}$$

where $Difficulty\_1\_target$ is a constant 256 bit number and $Current\_target$ is a 256 bit number. When calculating the difficulty for a hash, the hash itself is used as the $Current\_target$, then $Difficulty$ can be derived by (1). The Nakamoto blockchain network has a global block difficulty: valid blocks must have a hash below the current target. The hash is adjusted by changing the value of Nonce (a field in the block). The global difficulty is adjusted to limit the rate at which the network can generate one new block in an approximately fixed time interval.

Nakamoto blockchain has a pre-defined security threshold: the honest people must take more than 50% of computation power. This security threshold guarantees that the malicious people do not have enough power to create a longer fork branch of blocks when honest people are working on another branch. New participants can determine the correct records by staying with the longest chain (the mainchain) and we suppose this chain is longer than the second-longest chain for at least a given length.

No one should be able to send the same money to more than one receiver at the same time; this is the essential requirement for a decentralised-cryptocurrency. This requirement is fulfilled when participants can determine the correct record by checking whether the sender has spent the money or not in the history of the mainchain.

B. Smart Contract

A Smart Contract is a Turing Complete [43] computer protocol that is intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract [44]. Smart contracts allow the performance of credible transactions without third parties [44], [45]. These transactions are trackable and irreversible. A smart contract can be used as a unified interface for job assignment and result verification. When executing a smart contract, the user transfer an amount of funding into the smart contract address, each code execution is priced by the number of lines of codes executed.

C. Blockchain Sharding

Blockchain sharding is an improvement to the basic blockchain design which aims to achieve a secure consensus together with a sub-group of people among the whole population. Because the job workload and the nodes are divided into shards, nodes only need to process the data in a Shard and mostly communicate with nodes in a shard. This significantly reduces the requirement on computation capability and bandwidth. Also, because the shards are running in parallel, the overall processing ability is also increases tremendously. The main challenge is to inhibit the chance for adversaries who do not control the majority people globally but hold the majority people in a sub-group to temper the record. To bound the maximum chance for an adversary to gain control of a sub-group under a given security threshold, the number of sub-groups (shards) the system can have and the number of people (node) in the sub-groups are strictly restricted. Also, the model must fulfill the following requirements: (1) people cannot choose which subgroup they would be located in; (2) a transparent and random node assignment scheme must be employed, meaning no one can predict or manipulate which shard it is about to be assigned in; (3) the number of shards must be dynamically adjusted with the change of population.

We can calculate how many times of node assignments is required to guarantee an Adversary in controlling a Shard. The probability of obtaining no less than x adversary nodes when randomly picking a shard sized m (m number of nodes inside the Shard) can be calculated by the cumulative hypergeometric distribution function without replacement from a population of n nodes. Let X denote the random variable corresponding to the number of adversary nodes in the sampled Shard and t is the number of adversary. The failure probability for one committee is at most

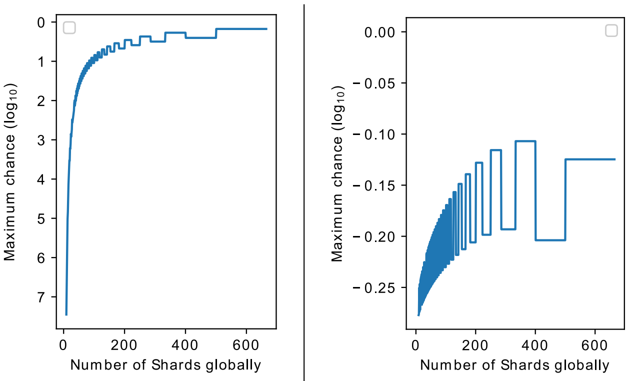
$$Pr[X > [m/2]] = \sum_{X=[m/2]}^{m} \frac{\binom{t}{X}\binom{n-t}{m-X}}{\binom{n}{m}} \tag{2}$$

Figure 1 shows the maximum probability to fail with $n = 2000$ and $m = n/s$ where s is the number of Shards. As can be seen from the result, the system has a very high failure chance when the adversary taken $n/2$ of nodes. We expect blockchain sharding approaches to maintain the same $n/2$ security level as in the original blockchain systems, but as shown in figure 1, the most blockchain sharding approaches can only withstand up to $n/3$ of nodes being bad, and only a few Shard can exist.

D. N/2 Adversary Resistant Blockchain Sharding

We previous proposed a *Jury* Hypothesis that serves as an analogy of an $n/2$ Byzantine-node tolerate blockchain sharding approach in [46], which further improves blockchain sharding.

The *Jury* hypothesis states that the member of a Jury of a court comes from the diverse background, so that when a verdict is reached, it can be seen as the decision was reached from the whole society (every class of people). If it takes m different occupations to form a jury, then when there are a s number of court hearings running in parallel, there are s number of people in each one of the m occupation. Table I shows a court schedule; each court represents a Shard, A is a person controlled by the Adversary while H is an honest person.



$t = n/3 = 666$ $t = n/2 = 1000$

Fig. 1. The chance to fail when $n = 2000$, $t = n/3$, $t = n/2$ and $m = n/s$ where s is the number of Shards

TABLE I
COURT JURY SCHEDULE

Court number Ocp	0	1	2	3
Occupation 1	A	A	A	A
Occupation 2	H	A	H	A
Occupation 3	A	H	A	H
Occupation 4	H	A	H	H
Occupation 5	H	H	H	H

It is ruled that a verdict is reached when a pre-defined T, $T > 0.5m$ number of people inside the jury reached a consensus. Assuming there exists a random assignment scheme that assigns people of the same occupation to different courtrooms where different court hearings are taken place in parallel. Then, the chance for the Adversary to gain T spots inside the target courtroom is (assuming without loss of generality that the adversary puts all its nodes into the front T occupations)

$$Pr[T] = \prod_{i=1}^{T} \frac{A_i}{s} \tag{3}$$

where A_i is the number of people inside courtroom i who are controlled by the adversary. To derive the maximised $Pr[T]$, we want $\prod_{i=1}^{T} A_i$ to be maximised because s is the same. Let the Adversary has AD number of people inside the system (Court Jury Schedule), then $AD = \sum_{i=1}^{m} A_i$. To maximise the value of $\prod_{i=1}^{T} A_i$, we consider

$$A_i = \lfloor AD/T \rfloor, i \in [1, T-1] \tag{4}$$

$$A_T = \lfloor AD/T \rfloor + AD \ mod \ T \tag{5}$$

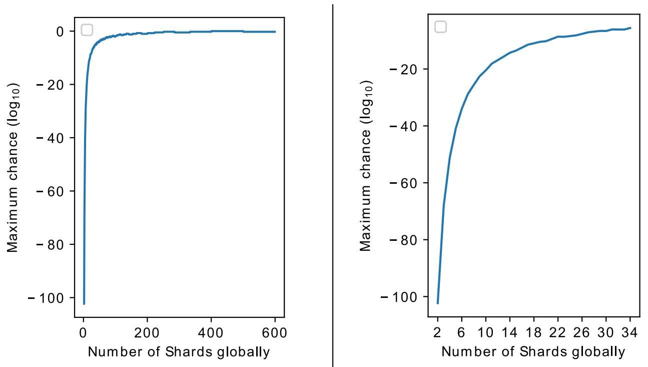
This scenario is the maximised because, given any positive integer X,

$$X * X > (X-1) * (X+1) = X * X - 1 \tag{6}$$

Thus,

$$Pr[T]_{max} \approx (\frac{AD}{T * s})^T \tag{7}$$

Though the adversary cannot manipulate a sentence when it does not have T people inside a Shard, it can halt a sentence to be reached when it has $m - T + 1$ number of the nodes in a Shard. Then this sentence cannot be made until the next court (the group of juries are re-selected). Thus, to make the system function more smoothly, we want $T \approx \lceil m/2 \rceil$ while meeting the security threshold (e.g. 10^{-6} failure chance). Figure 2 shows the maximum failure chance with different s, $n = s * m = 2000$, $T = 0.7 * m$ and $AD = 1000$ (1/2 fraction of the overall population).



$s \in [2, 600], AD = 1000 = n/2$ $s \in [2, 34], AD = 1000 = n/2$

Fig. 2. The chance to fail with different s when $n = 2000$ and $m = n/s$ where s is the number of Shards;

As can be seen from the result, when there are ten Shards and $n/2$ people being evil, the failure chance is below 10^{-20}, which significantly outperformed the traditional blockchain sharding at below 10^{-6} when it has ten Shards and only $n/3$ nodes being evil. If we set the block interval to be $30\ minutes$, then it takes over 10^{15} years to fail the system. If it is $10\ minutes$ (the same as Nakamoto blockchain), it still takes over 10^{14} years to fail. If we maintain a 10^{-6} failure chance at this circumstance with $T = 0.7 * m$, then there can be 33 Shards at the same time.

In [47], there is a method that dynamically splits and combines the occupations to recover the system from a global halting (all shards are halted at the same time).

III. Economic model

In this section, we show an economic model for a $n/2$ Blockchain sharding system with a halting recovery method. Smart contract is adopted in this blockchain sharding system. There are a funding pool and three types of accounts in our model.

1) *Transaction Account*. An address used for receiving compensation, keeping funding, and it can send and receive cryptocurrency to and from others. There is a fixed amount of transaction fee for every transaction related to the Transaction account. The fee is submitted to the funding pool automatically when the transaction is embedded into a block.
2) *Smart Contract Account*. A smart contract is associated with an account; every time the Smart Contract is executed, an amount of funding is sent to the system from this account. The funding of a Smart contract account initially comes from a Transaction account; the remaining funding can only be transferred back to that Transaction account.
3) *Margin Account*. The funding in a Margin account is frozen and cannot be used during a period.

Figure 3 shows the relationship between the accounts and the funding pool.

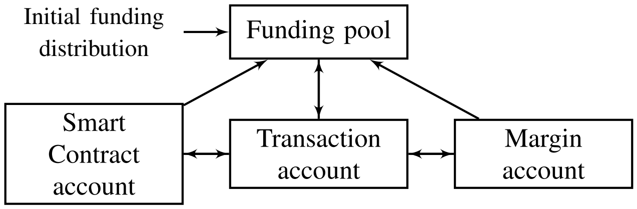


Fig. 3. Relationships between accounts and the funding pool

Grid computing models may run in mobile devices. It is common for mobile devices to go offline without prior notices due to the lousy network or low battery. This situation will not affect our model from functioning as long as there is a sufficient number of reliable nodes. We encourage reliable nodes to participate in the system by giving them compensation. In order to qualify for the compensation, the nodes need to first register as reliable nodes. They need to send an amount of funding from their *Transaction account* into the *Margin account* when registering. This funding will be unfrozen and being transferred back to the Transaction account if they stay online for a given period and carry out the duty by voting for consensus. The compensation is an additional fraction of the funding sent to the Margin account. The exact number of this additional fraction is changed by times (which is similar to the interests in time deposits in real life). If a reliable node goes offline within the period, the relevant funding in its Margin account is confiscated to the funding pool.

The price for executing smart contracts are changed time by time. This price is charged by the system, so that when a Smart Contract is executed, an amount of funding will be transferred from the smart contract account of this smart contract to the funding pool.

The same as Nakamoto blockchain, there is an initial funding distribution, where there is an amount of new funding released from the system in every block interval. This funding gradually reduces to zero block by block. After the initial funding distribution is finished, the money for executing the smart contract is the main source of funding for the funding pool to pay the compensation of the mining game as well as the compensation for reliable nodes.

A. The Pricing Model

We define the following financial indicators.

1) M_0: The amount of currency in Transaction accounts.
2) M_1: M_1 = M_0 + *funding in Smart contract accounts.*
3) M_2: M_2 = M_1 + *funding in Margin accounts.*
4) Q: The total lines of smart contract codes executed in a block interval.
5) P: The price for executing a line code in smart contracts.
6) $AVGQ$: The average of Q over a long time window.
7) U: The ideal coefficient of the M_2 used for executing smart contracts in a block interval.

P is adjusted in every block interval.

$$P_X = \frac{U \times M_{2_{X-1}}}{AVGQ + 1} \tag{8}$$

where P_X, $M_{2_{X-1}}$ are the P and M_2 at the block interval X and $X - 1$, respectively.

B. The Rewarding Model

We define the following three indexes.

1) GPL_X: The required length of time for a new reliable node at block interval X to stay online and carry out the duties. When a node transfers funding to its Margin account, it will become a reliable node after this transaction is embedded to a block.[1]
2) GN_X: The number of new reliable nodes at the beginning of the block interval X.
3) I: An indicator of the amount of funding in total to pay the compensation of nodes which start being reliable nodes at the beginning of the block interval X.

Let R_X be the amount of funding flood to the funding pool in total during the $X-1$ block interval. The funding consists of the transaction fees, the fees for executing the smart contracts and the money from the initial currency distribution.

In every block iteration, $I \times R_X$ amount of funding is remained in the funding pool and will be used to pay the nodes who start to serve as reliable nodes at block height X and finish serving at the block height $X+GPL_X$. Assumed an unreliable node $Alice$ starts to be reliable at block height X, the amount of compensation she will get at the block height $X + GPL_X$ is

$$\frac{Margin_{Alice}}{Margin_{\{X\}}} \times I \times R_X \tag{9}$$

where $Margin_{Alice}$ is the funding Alice frozen and $Margin_{\{X\}}$ is the total amount of funding that new reliable nodes at the block interval X frozen.

The "citizens" who participated in maintaining the system (propose blocks and/or vote for consensus) in block interval $X - 1$ will equally divide the remaining R_X at block height X.

C. Economic Policies

We define B as a fixed number of $\frac{M_2}{M_1}$.

We change the economic policies to make $\frac{M_2}{M_1} \approx B$ in every block interval. If $B = 2$, meaning, ideally, 50% of the overall currency should be placed in Margin accounts.

We use the L2 Regularisation [48] to analysis GPL,GN, I and $\frac{M_2}{M_1}$. Let

$$f(GPL_X, GN_X, I_X, \frac{M_{2_X}}{M_{1_X}}) = abs(B - \frac{M_{2_{X+1}}}{M_{1_{X+1}}}) \tag{10}$$

where $X \in [CH - 100, CH - 1)$, CH is the current block height.

We can then get the predicted GPL_{CH},and I_{CH} of

$$f_{min}(GPL_{CH}, GN_{CH}, I_{CH}, \frac{M_{2_{CH}}}{M_{1_{CH}}})$$

The predicted GPL_{CH} and I_{CH} then become the GPL and I at the block height CH.

Because the $GPL_X, GH_X, I_X, X \in [CH - 100, CH - 1)$ are data that every node in the system knows, every node can calculate the same GPL and I at the block height CH. So that, the economic policies are derived by the pre-defined rules instead of decided by some centralised authorities.

D. Interaction Between Currency and Resources

The maximum performance (transactions per second) of a blockchain sharding system is bound by the number of the Shards and the pre-defined maximum performance per shard. Thus, Q has upper limits. When every Shard is fully loaded, many unconfirmed transactions and smart contracts will be left in pending status. The funding associated with those pending transactions and smart contracts will also stay in pending status. Then, that affects the mobility of funding and decreases $\frac{M_2}{M_1}$.

Assuming at a block iteration, $U' \times M_2$ is the amount of money used to purchase resources, and AVQ is equal to the Q at its upper limit (fully loaded). According to equation 8, when $U' > U$, P goes up, and vice versa.

P affects the purchase demands, as well as the funding for the rewards. When P go up, the amount of reward is also increased, this can drive more devices into participating in the system. When P go down, the amount of reward is decreased, this surpasses the devices from participation. With the more devices participated, the more Shards can be formed, and that changes the upper limit of Q, vice versa. By the appropriate adjustments of the parameters, we can anchor a cryptocurrency with demands from the purchase market as well as the availability of the labour market. There is a derivation relationship between the exchange of the cryptocurrency and the resources and the exchange of the resources and real-world costs (electric bills and other fees to maintain the computation resources). Thus, the real-world value is placed to this cryptocurrency by the computation resources serving as the middle man. Figure 4 shows the relationship between the cryptocurrency, computation resources and the currency in real world.

[1] a node can be multi-reliable if it transfers funding to the Margin account when it is already a reliable node. The potential confiscation of funding and the compensation for participating are being count separately.

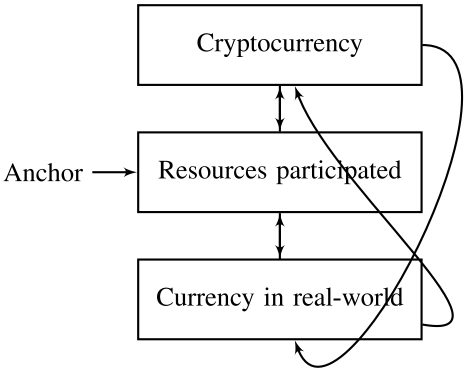


Fig. 4. The relationship between Cryptocurrency, resources and real-world currency

IV. Experiment

We simulate *20,000* nodes. They serve both as the user (the resources buyer) and the service provider (participating in generate blocks or vote for consensus). We set an amount of initial purchase demand for every node, which is presented in Figure 5. The purchase demand of every node is randomly either increased up to 5% or decreased up to 5% within every ten block intervals. We set an index called "fear line" for every node. "fear line" is an indicator for if the nodes should become reliable nodes. For example, if "citizen" Bob has 5000 currency, he uses ten currency to buy services, the time to bankruptcy is 500 block intervals, meaning after 500 block interval he will bankrupt if he continues buying and is not working. If Bob's "fear line" is 500 block intervals meaning he must start to participate in the system until the time to bankruptcy is higher than 500 again. Figure 5 shows the distribution of "fear line". When Bob submits funding to its Margin account, he ensures that he will not be bankrupted if keep buying resources during the frozen period. If the condition is fulfilled, Bob will attach a random amount of funding to the guarantee transaction.

We set $B = 2$, $U = 0.013$, and $AVGQ$ is the average Q over 50 continuous block intervals. $50,000,000,000$ amount of currency are sent in every block iteration in the initial currency distribution at first. The amount of funding sent out decreases by 2 in every 100 rounds until it reaches zero. The experiment lasted $10,000$ block iterations. In the first ten block iterations, the GPL, GN and I are set to be 0.1, 10, 0.1 and the AVQ is $5,000,000$. The upper limit is $GPL = 10,000$, $I = 0.8$ while the lower limit of I and GPL are 0.0001 and 10, respectively. In this experiment, we use linear Regression of scikit-learn of python to do the linear regression.

The purpose of the experiment is to show a working example of our model, the settings are arbitrary, as we cannot simulate the real usage patterns. The settings are beginning to be adjusted after the first ten block iterations. The adjustment of indexes may trigger some nodes to participate the system by achieving their "fear line". The indexes adjustment can be seen from Figure 6. The price is very stable, and the overall purchases are also stable while the regression algorithm makes $\frac{M_2}{M_1}$ around B.

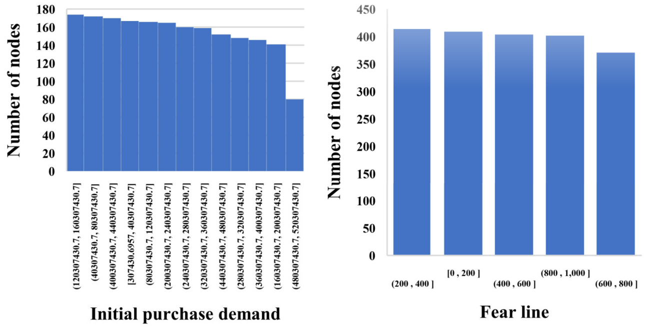


Fig. 5. Node Information

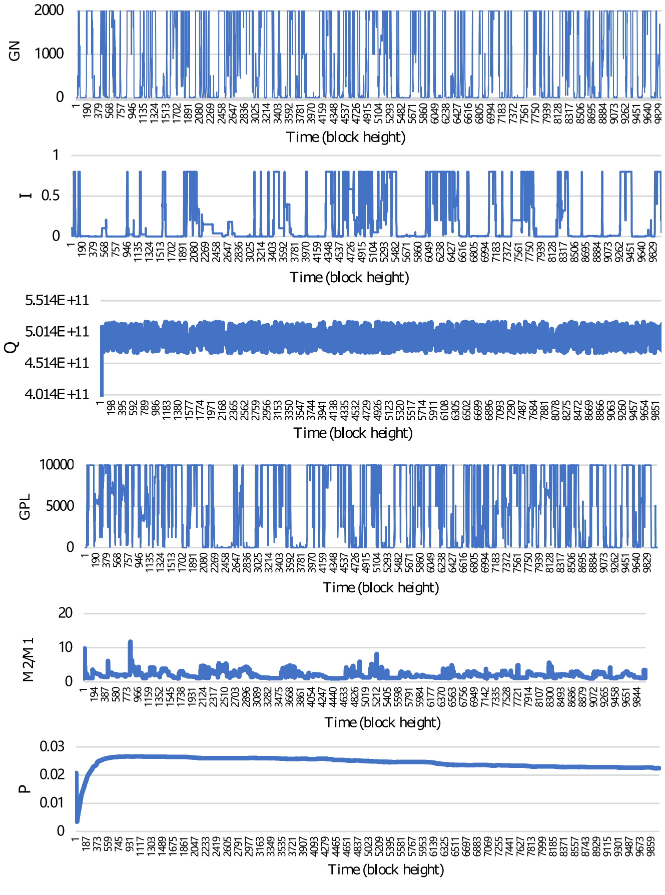


Fig. 6. Experiment Results

V. Conclusion

In this paper, we explored an economic model for blockchain sharding which is promising to power Grid computing. We attempted to link the price of resources with the digital labour market (the participation of nodes) and the resources purchase demand. We also attempt to stabilise and regulate the anonymous nodes by the financial mortgage. We bound the settings of Cryptocurrency with the amount of digital resources in the network. We anchor the value of cryptocurrency, in this way, the precise meaning of the cryptocurrency would be: It serves as the exchange among the computation resources worldwide.

References

[1] Anatol Murad. The nature of money. *Southern Economic Journal*, pages 217–233, 1943.

[2] Joseph Aschheim and George S Tavlas. Money as numeraire: doctrinal aspects and contemporary relevance. *Banca Nazionale del Lavoro Quarterly Review*, 59(239):333, 2006.

[3] Linda S Goldberg and Cédric Tille. Vehicle currency use in international trade. *Journal of international Economics*, 76(2):177–192, 2008.

[4] Helene Rey. International trade and currency exchange. *The Review of Economic Studies*, 68(2):443–464, 2001.

[5] Annette Kamps. The euro as invoicing currency in international trade. 2006.

[6] Daniel Gros. National institute economic review: a reconsideration of the optimum currency area approach: the role of external shocks and labour mobility. *National Institute Economic Review*, 158(1):108–127, 1996.

[7] Paul Maarek and Elsa Orgiazzi. Currency crises and the labour share. *Economica*, 80(319):566–588, 2013.

[8] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[9] Garrick Hileman and Michel Rauchs. Global cryptocurrency benchmarking study. *Cambridge Centre for Alternative Finance*, 33, 2017.

[10] John Fry and Eng-Tuck Cheah. Negative bubbles and shocks in cryptocurrency markets. *International Review of Financial Analysis*, 47:343–352, 2016.

[11] David Yermack. Is bitcoin a real currency? an economic appraisal. In *Handbook of digital currency*, pages 31–43. Elsevier, 2015.

[12] Robleh Ali, John Barrdear, Roger Clews, and James Southgate. The economics of digital currencies. *Bank of England Quarterly Bulletin*, page Q3, 2014.

[13] Jamal Bouoiyour, Refk Selmi, Aviral Kumar Tiwari, Olaolu Richard Olayeni, et al. What drives bitcoin price. *Economics Bulletin*, 36(2):843–850, 2016.

[14] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.

[15] Irina O Semyonova. The past and the future of blockchain. In *Recent Achievements and Prospects of Innovations and Technologies*, pages 111–116, 2017.

[16] Adem Efe Gencer, Soumya Basu, Ittay Eyal, Robbert Van Renesse, and Emin Gün Sirer. Decentralization in bitcoin and ethereum networks. *arXiv preprint arXiv:1801.03998*, 2018.

[17] Massimo Bartoletti, Salvatore Carta, Tiziana Cimoli, and Roberto Saia. Dissecting ponzi schemes on ethereum: identification, analysis, and impact. *Future Generation Computer Systems*, 102:259–277, 2020.

[18] Andrew Urquhart. The inefficiency of bitcoin. *Economics Letters*, 148:80–82, 2016.

[19] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. *Communications of the ACM*, 61(7):95–102, 2018.

[20] Wim Raymaekers. Cryptocurrency bitcoin: Disruption, challenges and opportunities. *Journal of Payments Strategy & Systems*, 9(1):30–46, 2015.

[21] Loi Luu, Duc-Hiep Chu, Hrishi Olickel, Prateek Saxena, and Aquinas Hobor. Making smart contracts smarter. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 254–269. ACM, 2016.

[22] Karthikeyan Bhargavan, Antoine Delignat-Lavaud, Cédric Fournet, Anitha Gollamudi, Georges Gonthier, Nadim Kobeissi, Natalia Kulatova, Aseem Rastogi, Thomas Sibut-Pinote, Nikhil Swamy, et al. Formal verification of smart contracts: Short paper. In *Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security*, pages 91–96. ACM, 2016.

[23] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151(2014):1–32, 2014.

[24] Carlos Pérez Jiménez. Analysis of the ethereum state.

[25] Hung Dang, Tien Tuan Anh Dinh, Dumitrel Loghin, Ee-Chien Chang, Qian Lin, and Beng Chin Ooi. Towards scaling blockchain systems via sharding. In *Proceedings of the 2019 International Conference on Management of Data*, pages 123–140. ACM, 2019.

[26] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 931–948. ACM, 2018.

[27] Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ewa Syta, and Bryan Ford. Omniledger: A secure, scale-out, decentralized ledger via sharding. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 583–598. IEEE, 2018.

[28] Rajitha Yasaweerasinghelage, Mark Staples, and Ingo Weber. Predicting latency of blockchain-based systems using architectural modelling and simulation. In *2017 IEEE International Conference on Software Architecture (ICSA)*, pages 253–256. IEEE, 2017.

[29] Axel Moinet, Benoît Darties, and Jean-Luc Baril. Blockchain based trust & authentication for decentralized sensor networks. *arXiv preprint arXiv:1706.01730*, 2017.

[30] Dr Andreas Freund. Economic incentives and blockchain security. *Journal of Securities Operations & Custody*, 10(1):67–76, 2018.

[31] Joshy Joseph. *Grid computing*. Pearson Education India, 2004.

[32] Brian Hayes. Cloud computing. *Communications of the ACM*, 51(7):9–11, 2008.

[33] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. *arXiv preprint arXiv:0901.0131*, 2008.

[34] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *2008 10th IEEE International Conference on High Performance Computing and Communications*, pages 5–13. Ieee, 2008.

[35] John W Rittinghouse and James F Ransome. *Cloud computing: implementation, management, and security*. CRC press, 2017.

[36] B Au-Yeung, D Chu, M Enfante, G Logan, and K Saelee. Industry analysis: Cloud computing, 2016.

[37] George T McCandless, Warren E Weber, et al. Some monetary facts. *Federal Reserve Bank of Minneapolis Quarterly Review*, 19(3):2–11, 1995.

[38] Milton Friedman. Quantity theory of money. In *Money*, pages 1–40. Springer, 1989.

[39] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.

[40] Adam Back et al. Hashcash-a denial of service counter-measure. 2002.

[41] Pavel Vasin. Blackcoin's proof-of-stake protocol v2. *URL: https://blackcoin. co/blackcoin-pos-protocol-v2-whitepaper. pdf*, 71, 2014.

[42] Yibin Xu and Yangyu Huang. Mwpow: Multiple winners proof of work protocol, a decentralisation strengthened fast-confirm blockchain protocol. *Security and Communication Networks*, 2019, 2019.

[43] Bogdan Aman and Gabriel Ciobanu. Turing completeness using three mobile membranes. In *International Conference on Unconventional Computation*, pages 42–55. Springer, 2009.

[44] Christopher D Clack, Vikram A Bakshi, and Lee Braine. Smart contract templates: foundations, design landscape and research directions. *arXiv preprint arXiv:1608.00771*, 2016.

[45] Patrick McCorry, Siamak F Shahandashti, and Feng Hao. A smart contract for boardroom voting with maximum voter privacy. In *International Conference on Financial Cryptography and Data Security*, pages 357–375. Springer, 2017.

[46] Yibin Xu and Yangyu Huang. An n/2 byzantine node tolerate blockchain sharding approach. *arXiv preprint arXiv:2001.05240, The 35th ACM/SIGAPP Symposium On Applied Computing*, 2020.

[47] Yibin Xu, Yangyu Huang, Jianhua Shao, and George Theodorakopoulos. A flexible n/2 adversary node resistant and halting recoverable blockchain sharding protocol. *arXiv preprint arXiv:2003.06990, Doi:10.1002/CPE.5773, Concurrency and Computation Practice and Experience*, 2020.

[48] Robert Moore and John DeNero. L1 and l2 regularization for multiclass hinge loss models. In *Symposium on Machine Learning in Speech and Language Processing*, 2011.