Improvement based on the proof-of-work consensus algorithm

Liu Meng*, Sun Xuebo
University of Science and Technology Liaoning
949847537@qq.com

Abstract. Among the Bitcoins using Proof-of-work consensus algorithm, handling capacity is seven second each sum, which is less inefficient. But the Proof-of work depends on the computer function. It is beneficial to the higher-counting point, but unfair to the normal point. Make the computing power centralized. Aiming at the above-mentioned the waste and centralization of the Proof-of-work synchronous arithmetic, as well as the transaction-processing efficiency, we do the research and improvement.

1. Introduction
With the rapid development of digital finance, countries have accelerated their research on digital currencies. How to generate a stable digital currency that is safe and convenient, cost-controllable, fast in transactions, and highly anonymous has become the key for governments to occupy the commanding heights of the financial sector. As the core technology of digital currency, blockchain technology is widely used in data traceability, internet of things, finance, privacy protection and other fields, providing users with a trusted channel to transmit financial and other private information in unknown networks. The intricate network has a high degree of anonymity and randomness. The authenticity of the information transmitted between nodes is difficult to identify. In order to ensure the consistency of the information transmitted, blockchain technology needs an excellent consensus algorithm. Bitcoin as the origin of blockchain technology, uses a proof-of-work consensus algorithm that is more representative than other consensus algorithms. This article mainly research and improve the core algorithm of bitcoin — the proof-of-work consensus algorithm.

2. The primary-secondary- stage block consensus algorithm
Among the Bitcoins using Proof-of-work consensus algorithm[1], a round of consensus in ten second can produce one clock and the the carrying quantity is limited, which causes less inefficient handling capacity. Each point, struggling by the computer power, solve the process that verifies easily, but the sophisticated SHA256[2-3] math difficulty is its central idea, which is more beneficial to the strong point of counting capacity and causes the unfair phenomenon. This passage put forward an idea which is called the primary-secondary- stage block consensus algorithm. The choose of the main block is similar to the roof-of-work consensus algorithm, while the secondary block takes advantage of the characteristics of strong collision-resistant and the uniterminating random selection in Hash consistency algorithm to assure the fairness. Through the primary and secondary blocks, the efficiency of the consensus mechanism in processing transactions is increased, and the centralization phenomenon is reduced, making the consensus process faster, more stable, and fair.
2.1 Data structure
The block structure drawing is as shown in the figure one. The blocks include Hash values H\text{pare} in the main block (father block is the main block or the secondary block), Hash values H\text{prim} in the secondary block, the height of the block B\text{pare}.num (including the main and secondary blocks), the height of the main block B\text{prim}.num, Timestamp, Nonce, Target, Merkle root and mapping values. Values in Nonce and Target are zero. The mapping calculation formula is as follow:

\[
\text{Mapping} = \text{hash}(H\text{pare}||\text{Timestamp}||\text{the public key})
\]

A block body has a set of rally point formed by the transaction \{TS\} and the quantity T\text{number}.

![Figure 1. block structure of two level block consensus algorithm](image)

2.2 consensus process
The generation of the main block is similar to the working quantity, seeking for the proper Nonce values to solve the SHA256 math difficulty. The faster solving point can produce the new main block. The generation of the secondary block is by the random lottery form to struggle for the power of the blockchain. First, each node need calculate the value in each consensus and choose the packing block from the transaction pool, then broadcast to the whole network. The node having received the block from the whole net takes advantage of the Hash consistency algorithm to make all the mapping values gather into the rally point M\{m1, m2, m3,...,mn\} and map to the Hash circle. This circle is labeled as Ring (M).

To make Hash values of father block, the height of block, Merkle as the keywords counts the direction mapped on the Ring(M)- Site(k), then find the relative mapping values. Later broadcast this block with this mapping values to the whole net nodes. After receiving the blocks, they will verify. After verifying, this node will receive the new block and chain into the back of the father block. The process picture of the secondary block is shown as Picture two.
2.3 Arithmetic description

The process of the main block consensus is shown as algorithm one. The process of the secondary is like algorithm two.

Algorithm 1  Main block de consensus process

Procedure Main block Consensus Process

Input:  Bpre: Previous Block;{TS}:Transaction Set
Output : Update BlockChain

for SHA256(Hpare||nonce)>=Target
  nonce=nonce+1
end for
Bprim.num ← Bpre.prim.num+1
Bpare.num ← Bpre.pare.num+1
Hpare ← SHA256(Bpre)
Hprim ← Hpre.prim.
Timestamp ← Time of nonce finding
Bnew.Body ← ({TS},Tnumber)
Merkle ← Hash({Bnew.Body})
Bnew.Header ←
(Bprim.num,Bpare.num,Hpre,Hprim,Timestamp,Target,Nonce,Merkle)
Bnew ← (Bnew.Header,Bnew.Body)
if Node receives Bnew and
  Bnew.prim.num=Bnew.pre.prim.num+1 and Bpare.num=Bpre.pare.num+1
  Hnew.pare ← SHA256(Bpre) and Hnew.prim ← Hpre.prim
Algorithm 2 Sub-Block Consensus Process

Procedure Sub-Block Consensus Process

Input: Bpre: Parent Block; Bprim: Primary Parent Block; \{TS\}: Transaction Set

Output: Update BlockChain

mapping=Hash(Hpare||Timestamp||PK)

Bnew.Body \leftarrow \{\{TS\},Tnumber\}

Merkle \leftarrow \text{Hash}(\{Bnew.Body\})

Bnew.Header \leftarrow (Hpare,Hprim,Bpare.num,Bprim.num,Timestamp,nonce,Target,Merkle,mapping)

Bnew \leftarrow (Bnew.Header,Bnew.Body)

Broadcast Bnew

Blocks \leftarrow \text{Node receives the Bnew broadcasted}

M\{m1,m1,\ldots,mn\} \leftarrow \text{sort(Blocks.mapping)}

Ring(M) \leftarrow \text{Consistent-Hash}(M\{m1,m2,\ldots,mn\})

keyword=Hash(Hpare||Bpare.num||Merkle)

Site(k) \leftarrow \text{Consistent-Hash(keyword)}

mapping_k \leftarrow \text{The position of the Site(k) on the Ring(M)}

if Blocks.B.mapping=mapping_k

Broadcast B

end if

if Node receives B and

B.Hpare=Hpare and B.Hprim=Hprim and

B.header is true and B.body is true and

B.mapping=mapping_k

Update BlockChain

end if

end procedure

3. Experiment and analyses

3.1 Experimental environment

Based on the two-level blocks consensus algorithm, we do the simulation experiment and simulate the form of the blockchain. We simulate the basic working quantity confirming consensus algorithm, and compare the results to verify the steadiness and the transaction throughput. The basic environment of the experiment is below.

Language used: golang
Processor: Intel Core i5-8400u
Memory: 8GB for unit 1 and 16GB for unit 2
System: Ubuntu 16.04.5 lts64 bit.
3.2 Fairness comparison and analysis
We set separately eight or sixteen node, with one hundred consensus processed. During the eight nodes experiment, 1-6 is No.1 unit and 7-8 is No.2 unit. During the sixteen nodes experiment, 1-12 is No.1 unit and 13-16 is No.2 unit. Aimed at two algorithm experiments, the result processing the block successfully is as the picture 3 and 4.

![Figure 3 scatter diagram of the number of successful blocks of 8 nodes](image)

![Figure 4 scatter diagram of the number of successful blocks of 16 nodes](image)

When having eight nodes, the time of processing blocks is between five to twenty-five, especially No.7 and 8 having much more times than other nodes. The times are less homogeneous distribution and the inclination of the trendline is bigger. Whether the No.1 node or the No.2 node has the different times of processing blocks between ten and sixteen, with uniform distribution and steady inclination.

When having sixteen nodes, the time of processing blocks is between one to thirteen, the scatter diagram has a larger span and uneven distribution, especially No.13 and 16 having much more times than other nodes. Two-level blocks consensus algorithm has the different times of processing blocks in three or nine times. The scatter diagram distributes more average and has smaller trend than Proof-of-work consensus algorithm.
3.3 The comparative experiment of Transaction throughput

When the node number increases from two to four, eight, sixteen, the average throughput of two algorithms also add, even reaching the top value when total is eight. The average throughput of two-level consensus algorithm is 925.19 unit/second, 3.35 times than Proof-of-work consensus algorithm (275.76 unit/second). After the total eight nodes, the average throughput of two algorithms decrease gradually. Picture five shows the comparative situation of the average throughput using these two algorithms.

Figure 5 line chart of average throughput under different number of nodes

4. Conclusion

Compared two-level blocks consensus algorithm with proof-of-work consensus algorithm, each node produces the blocks on average, reducing effectively the centralization of the computers, which is more fair. Under the circumstance of different nodes, two-level blocks consensus algorithm is superior than proof-of-work consensus algorithm. The simulation experiment shows the performance of two-level blocks consensus algorithm is superior than proof-of-work consensus algorithm.

However, blockchain develops rapidly and extends greater. Not only the extension but the cross with other vocation behaves differently such as adopting different cryptology algorithms will change the structure of the consensus algorithm[4-5], then influence the integrity and structure of the blockchain. Blockchain as the digital currency technology need not only the support of cryptology, but the participation from finance as well. They also need the support from variety of network protocol in the network structure.

References

[1] Kraft D. Difficulty control for blockchain-based consensus systems[J]. Peer-to-Peer Networking and Applications,2016,9(2):397-413.
[2] Back A. Hashcash-a denial of service counter-measure[J]. 2002,2.
[3] Chen Xiaojiao. Data compression of bitcoin blockchain [D]. University of science and technology of China, 2020
[4] Yu Hui, Zhang Zongyang, Liu Jianwei. Research on bitcoin blockchain expansion technology [J]. Computer research and development, 2017,54 (10): 2390 – 2403. [doi: 10.7544/issn1000-1239.2017.20170416]
[5] He Pu, Yu Ge, Zhang Yanfeng, et al. Overview of blockchain technology and application [J]. Computer Science (4)