Parallel Computing based Bitcoin Currency System Analysis Approach

Zhaokai Luo, Wenfeng Shen
School of Computer Engineering and Science
Shanghai University
Shanghai, P. R. China
whitelok@gmail.com, wfshen@mail.shu.edu.cn

Luokai Hu
Lenovo Mobile Communication Technology Ltd.
Xiamen, P. R. China
luokaihu@gmail.cn

***Abstract*—With the rise and extensive usage of Bitcoin, a peer-to-peer electronic cash system beginning at 2008, the number of transactions is growing. In order to analyze the activity in this currency system, we present a parallel analysis approach for meeting the need of building the transaction graph of this financial system. In order to test the performance and the realistic possibility of our approach, we implemented our approach and conducted some comparing to test the performance of our system. Through the experiment, we confirmed that this method is highly efficient and reliable compared with the traditional method.**

Keywords-Bitcoin; Parallel Computing; Transaction Graph

I. INTRODUCTION

Bitcoin [1] is a peer-to-peer network based digital currency which means that there is not central authority to control money remittance channels. To avoid the potential safety hazard of the central reserve bank bad policy and instability caused by, the whole currency system has to be managed by the whole network nodes. Therefore, In order to carry out the financial activities in the decentralized system, Bitcoin must make a detailed record of the financial system in each transaction. However, without the center of this currency, it is hard to manage the transaction activity for example: tracing a special account's transaction records. In the financial field, it is meaningful to analyze the flow of funds through the each transaction records. However, with the usage of Bitcoins transaction activities growing, this will produce huge amounts of data daily that make every financial analysis becomes difficult. Therefore, to provide a fast and real-time transaction analysis graph, this paper proposes a parallel data processing approach which is suitable for cluster computing.

In section 2, we will discuss the related work of the Bitcoin data analysis. Then, we present a new approach to gain the raw data fast and store the data in distributed cluster in section 3. In section 4, we build up transaction graph with the parallel computing. In section 5, we implemented a system with our approach and analyzed its performance. Finally, we draw our conclusion and plan our future work in section 6.

Project supported by the National High-tech R&D Program of China (Grant NO. 2009AA012201), the Shanghai Leading Academic Discipline Project (Project No.J50103), and the Innovation Project of Shanghai University.

II. RELATED WORK

As a decentralized currency system, Bitcoin, raised by Satoshi Nakamoto[1], has attracted mass attention since 2008. Because Bitcoin is a combination of financial, economic and monetary system of the new computer technology, there are many researches on it.

The most research content is the system of analysis for example: Bitter to Better—How to Make Bitcoin a Better Currency [2], Bitcoin: an innovative alternative digital currency [3], Bitcoin: censorship-resistant currency and domain system for the people [4]. All these paper focus on the system introduction.

The other people focus on privacy of the transaction for example: Evaluating User Privacy in Bitcoin [5], Bitcoin: Tempering the Digital Ring of Gyges or Implausible Pecuniary Privacy [6]. To a certain extent, these papers analyze the security part of the Bitcoin currency system instead of the researching environment.

Comparing with the existing articles, this paper raises a parallel research-environment for Bitcoin transaction. With the method we proposed, people can get huge Bitcoin data fast. What's more, with the distributed storage approach in our paper, we can get all the information of Bitcoin accounts with a low time-consumption.

III. DISTRIBUTED STORAGE

A. Parallel Block Download

As the description of the paper [1], all the transactions are stored in the block chain which is shared by all nodes participating in a system based on the Bitcoin protocol. Bitcoin's block chain contains every transaction ever executed in the currency from the beginning of the Bitcoin currency system. According to the research [7], the usage of Bitcoin growth rate reaches double digits and its block chain's size increases rapidly. Therefore, to download all the transaction records in the block chain in serial is the time-consuming work. Hence, we propose a parallel approach to download the block fast in the cluster and build up a cluster computing environment for further parallel computing.

978-0-7695-5125-8/13 $26.00 © 2013 IEEE
DOI 10.1109/CSA.2013.118

As the blocks are shared by all nodes participating in a system, every node in our cluster can access the different blocks randomly. To identify which block has been downloaded, each node in our cluster maintains blocks' hash value list and data list. The initial lists are empty at the beginning. Before downloading a block, the node will extract the hash value from this block. To define whether to download this block, we define a formula as below:

$$V(block_i)=\begin{cases}1 & \text{if not exist in other nodes' lists or timeout}\\ 0 & \text{if exist in other nodes' lists}\end{cases} \quad (1)$$

A node can decide whether to download the node with this hash value according to formula (1).

$$T(V(block_i),block_i)=j \quad (2)$$

As shown in formula (2), we can define a function that chooses the node to download the block. When we give a block this function, we can get node with "j" tag to download this block. The original block contains hash value as the block ID.

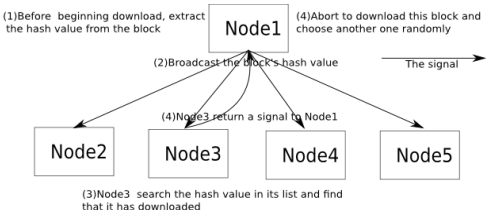

Figure 1. Nodes communication before downloading.

As shown in figure 1, when a node beginning to download a block, the node will broadcast the hash value extracted from the block to all nodes in the cluster. And the node which receives the broadcast will find the hash value in its own list. If the hash value exists on the list, the node will send a signal to the broadcast sender. When received the signal, the node which broadcasts the block hash value will abort the downloading procedure. In the meanwhile, this node will choose another block and broadcast the block's hash value again.

After the node broadcasting the block's hash value, the other nodes response timeout. The broadcast sender will consider that the block has not been downloaded and begin to download this block. After downloading the block, the node will begin to download the previous block according to the current block chain. We can find the previous block through the previous field of block structure.

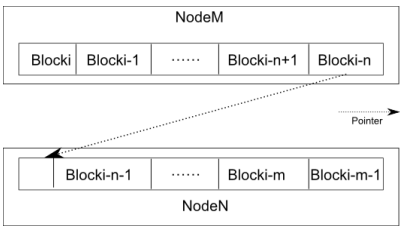

Figure 2. The situation of non-empty list.

There is a situation when the list on the node is not empty that the broadcast sender receives a signal. It means that the list on the broadcast sender and the signal sender can be merged. Because we conduct the next downloading task with the reference of prior block, we add a pointer to the tail of broadcast sender's list. It means that the prior block of the last element of broadcast sender's list is the first element of the signal sender's list as shown in figure 2. And then, the broadcast sender will create a new empty list and choose another block to download before broadcasting.

When finishing all downloading tasks in the cluster, we can get a segment list with the whole block chain stored distributed on the nodes.

Parallel downloading in this way is based on the fact that there is only one chain without branched in the Bitcoin currency. It is ensured by the Bitcoin protocol [8]. In this way, we can gain all blocks in the chain fast. According to our experiment data, the efficiency of downloading the block chain is increased by 70% compared with the official Bitcoin client: Bitcoin-qt.

B. Graph Update

For the Bitcoin generating block in ten minutes, when we finished downloading the old block chain, there are some new blocks at the last of the block chain which is shared by the whole network. Therefore, when there are some blocks, we can conduct the same step discussed above to update the local data.

However, there is a situation that when we finished the procedure of downloading, the currency system just has created a new block. Hence, our block downloading procedure will degenerate to be the procedure downloading the block on single machine. In order to improve the situation, we add a constraint condition for our approach.

At the moment of there is just a new block shared in the network, every node in the cluster will read the transaction record in the input list and output list parallel. Ignoring the invalid or unconfirmed transactions, our system will trace the corresponding output and input record from the previous records following the approach discussed above.

Downloading the block with this method, all the nodes in the cluster will save this block's hash value (block's ID) in their own list with a tag.

C. Block Data Backup

In order to ensure that data is not lost, we have to backup our block data in the situation of adding or removing nodes.

Considering every block can be identified by it hash value and be downloaded any time, we can just backup the list of block downloading. In our experiment, we backup the node's block downloading list in two other nodes. With this backup method, we can keep our system running at the moment of 50% nodes in the cluster have broken down.

When restore the block data, node can get the block list from other nodes and re-download the block by the list.

IV. PARALLEL DATA ANALYSIS

A. Raw Data Structure

After saving the whole block chain, we will extract the transaction data from the block. In order to get useful data, we must know about the structure of the transaction.

There are two kinds of the transaction existing in Bitcoin currency: Mining and Paying.

At the situation of paying the bill, address A broadcasts a output; address B broadcasts a input to the whole currency system. At this time, the two sides will broadcast the transaction information to the whole Bitcoin currency network that A pays E to B. In order to prevent the third party forging the transaction information, the transaction information will be encrypted by A's private key. Network nodes which receive the transaction information can be used to address A's public key to verify the transaction information. As the wiki of Bitcoin, user account's ID is as same as its public key. To finish the transaction, user has to keep their private keys.

When the network node receives the transaction information, it will verify the effectiveness of the transaction. After rejecting the invalid transaction, the node will combine the hash value of the latest block with the transaction without verified and create a new block. Hence, we can ignore the unverified transaction without redeeming.

Therefore, to trace every account's activity in the situation of paying, we have read all the records input-by-input as well as output by output.

At the situation of mining, an account will gain reward from creating a new block. This is the source of the original Bitcoin.

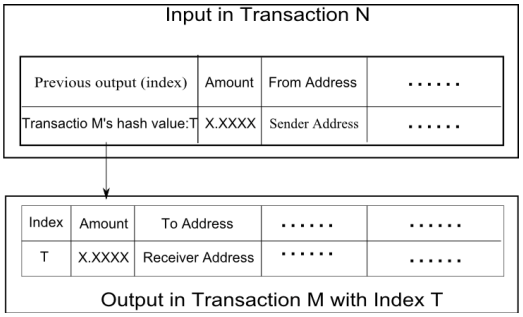

Figure 3. Two parts of transaction in Bitcoin protocol.

According to the protocol of Bitcoin [9, 10] which shown above, when someone wants to trigger a transaction, he must state the amount and the target account (Bitcoin address) in an output block. When verifying the validity of the B global, it generates an input in a new block.

TABLE I. INPUT SAMPLE[A]

Previous output (index)	Amount	From address	Type	ScriptSig
343ddc…:124	0.10985983	1JPFCUtH2gc…	Address	30440…

A.Sample from http://blockexplorer.com/.

TABLE II. OUTPUT SAMPLE[A]

Index	Amount	To address	Type	ScriptSig
2	0.10985983	1DRd1waDtV9BIZ…	Address	OP_DUP…

A.Sample from http://blockexplorer.com/.

In the table Ⅰ, the column "Previous output" means that the truncated hash of a previous transaction and the index of the output that this input is redeeming (after the colon). The first output in a transaction has an index of 0. Therefore, we can trace the output by this field. As shown above, we can trace the input shown in table Ⅰ and its corresponding output at table Ⅱ. As shown in the two tables, we can get the sender address (sender account) and receiver address (receiver account). What's more, they can get the amount of the transaction from different block.

B. Graph Creation

To get the graph of the Bitcoin, we propose a structure for building a fast computer environment of analysis. In order to create such graph fast and real-time, we process the data of block in cluster with the distributed storage.

After finishing downloading all blocks in the block chain, nodes begin to scan the block chain to extract the transaction information.

$$F(transaction_i) = j \tag{3}$$

As the formula (3), when the nodes gain transactions from block with parallel computing, we can define the node which should save this transaction by this function provided a transaction.

At the beginning of the procedure, each node in the cluster has to create an empty database to store the graph. And this database contains two tables: Sender Table and Transaction table. In the Sender Table, it contains three columns: sender address (sender account), sender transaction hash value and the sender's transaction output index. In the Transaction Table, it contains three columns too: sender address, transaction amount and receiver's account.

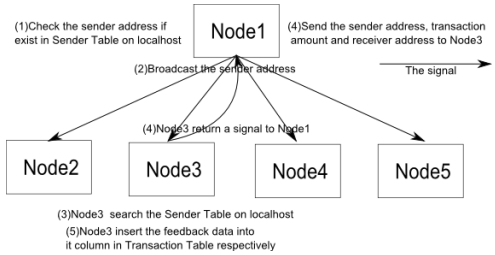

Figure 4. The communication of creating the graph.

When scanning the blocks, we can get sender's accounts and the receiver's accounts from transaction records. However, these two addresses are stored in different transaction. Therefore, when the node searching the transaction's output list, it will search its Sender Table with this sender account. If it exists, the node will insert the sender account, transaction amount and receiver account into Transaction Table. And the

receiver account will be inserted later. When scanning the input list, we meet two situations: the previous output field is null in the record and the previous input field is not null.

When previous output field is null, we can consider that the receiver account gain a fee from currency system. In this situation, we can insert a transaction into the Transaction Table on the node contains a sender account of "N/A".

When previous output field is not null, we can broadcast its "From address" field's value to all nodes in the cluster. When nodes receive this message, they will search its own Sender Table. At the moment of the sender does not exist in Sender Table, the node will broadcast the sender's account to all nodes in the cluster. When the other nodes receive the sender's account, they will search their Sender Table with this sender account. If the sender account exists in the Sender Table, the node will send back a signal to the original account sender. When the original account sender receives the signal, it will send the sender account, transaction amount and receiver account to the node which sends the signal. And the signal sender will insert the sender account, transaction amount and receiver account into its own Sender Table. There is a situation that after broadcast the sender account to all nodes in the cluster, the response is timeout. In this situation, the broadcast sender will create a new sender in its own Sender Table.

We build up a graph in the cluster as distributed storage after above steps. Requiring the information of the account is just a parallel search in the cluster. We can delete the original blocks data and just save the block hash value on nodes that we have already saved our data in the database on the nodes.

C. *Backup Graph*

For the reason of data reliability, we have to create the rule of data redundancy. According to our definition, we save transaction identified by the sender account. It means that we save all transactions with the same sender account in one node. At the meanwhile, the sender account on this node is unique in the cluster. We can just backup the Transaction Table to two other nodes at the leisure time of the inner network.

V. EXPERIMENT

Firstly, we compare the raw data downloading speed with Bitcoin-qt which is the official client of the Bitcoin currency system.

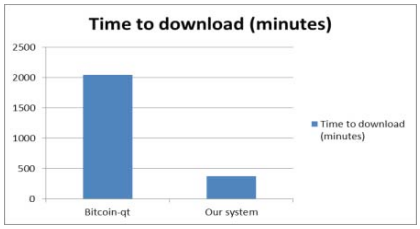

Figure 5. The downloading speed of to system.

The figure 5 shows our system is 81.55% faster than Bitcoin-qt.

Secondly, we compare the account balance retrieving speed with Bitcoin-qt.

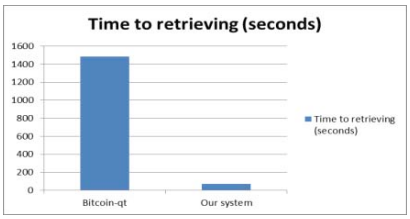

Figure 6. Two parts of transaction in Bitcoin protocol.

The figure 6 shows our system is 95.14% faster than Bitcoin-qt.

VI. CONCLUSION

Comparing with other tools of gaining the block chain data, the downloading speed of our approach is faster. In the view of the block data search website, users can customize their application based on our approach. As the character of real-time, our approach has the practical value in the financial field. Through creating the graph, we can trace every transaction and search balance of account fast and easily in our approach. What's more, while our approach is based on the distributed storage, we can implement the analysis tools of big data easily. We also built up an efficient platform for the huge amount data analysis.

We will use our method in different situation not only the Bitcoin currency system. In order to consider globally, we will take Bitcoin and other currency exchange rate into account to make the graph be closer to the actual.

ACKNOWLEDGMENT

The work is partially supported by the Postdoctoral Workstation of Lenovo Mobile Communication Technology Ltd.

REFERENCES

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. Consulted, 2008, 1: 2012.

[2] Barber S, Boyen X, Shi E, et al. Bitter to Better—How to Make Bitcoin a Better Currency, Financial Cryptography and Data Security. Springer Berlin Heidelberg, 2012: 399-414.

[3] Grinberg R. Bitcoin: an innovative alternative digital currency. Hastings Sci. & Tech. LJ, 2012, 4: 159.

[4] Bitcoin: censorship-resistant currency and domain system for the people, Forum American Bar Association. 2011.

[5] Androulaki E, Karame G, Roeschlin M, et al. Evaluating User Privacy in Bitcoin. IACR Cryptology ePrint Archive, 2012, 2012: 596.

[6] Elias M. Bitcoin: Tempering the Digital Ring of Gyges or Implausible Pecuniary Privacy. Available at SSRN 1937769, 2011.

[7] Yan Li. (March 21 2013) http://www.kuaiyilicai.com/201305/3fef1960-c1cf-11e2-888d-4f29d41a7088.html

[8] (July 22 2013) https://en.bitcoin.it/wiki/Transactions.

[9] (July 22 2013) https://en.bitcoin.it/wiki/Blocks.

[10] Grinberg R. Bitcoin: An Innovative Alternative Digital Currency. Hastings Science & Technology Law Journal, 2011, 4: 160.