Methods of Developing Systems Based on Blockchain

Mikhail Gorodnichev, Alexandra Kukharenko, Elena Kukharenko,Tatyana Salutina
Moscow Technical University of Communication
Moscow, Russia
m.g.gorodnichev@mtuci.ru, a.m.kukharenko@mtuci.ru,
e.g.kukharenko@mtuci.ru

***Abstract*—This article explores the concept of distributed systems and the technology of blockchain. At the beginning the reader may find out the short historical note about developing of blockchain since 20th century. Further there is the short description about functional principles of blockchain systems. The protocol for transferring the information was also created as a technology of blockchain application example.**

I. Introduction

Considering the term "blockchain", it becomes clear that in the modern world this concept is almost also important and irreplaceable as the Internet.

In recent years the principle of work of many large companies making different products considerably changed. In the course of decrease in expenses companies use different methods to create the cheapest production of products or minimize their cost value. Cost value depends on such factors as location of the company that affects the cost of the earth and rent of rooms, the cost of human resources in the specific region, the cost of resources, etc. In the modern world thanks to information technologies there are an opportunity to connect many aspects of production activity despite their location and also to manage effectively production without the company management.

Today many large corporations work at the world market and use production capacities in the different countries and continents that considerably reduces their expenses. Because of this fact, it is possible to conclude that the majority of the information systems and technologies used on production are distributed.

Such phenomenon can be proved by the following reasons also:

- Geography: Large corporations and the companies are geographically distributed all over the world, nevertheless the connection of elements of the company should not be broken;
- Parallelism: Computer technologies develop fastly, multi-core processors and computer clusters are used. This fact opens new opportunities of using technologies;
- Reliability: Information is distributed on several carriers to reduce risk of its loss;
- Usefulness: Information is distributed on several carriers also to provide it is quick access and to reduce number of delays and failures.

The problem of this article is the need to explore the blockchain system as it becomes more and more popular in the technical sphere [7]. The main aim of the article is to clarify the concept of working of the system based on blockchain.

II. Analyzing

Having analyzed the listed reasons, it is possible to select a number of advantages which distributed systems have: increase in volume of data storage, computing power, an opportunity to connect spatially separated locations.

In addition to these advantages we should remember that the system should be fault-tolerant. Its work should not be broken because of failures of separate components. Otherwise it will not be able to be inefficient for the user.

The meaning of a fault-tolerant system is that the each component can be a failure, but at the same time functioning of other components and also the whole system will not be broken. It is also necessary to deal with the meanings of such concepts as "client" and "server".

The each element of a system is calling a node. In the simplest client-server model each computer is an equivalent node. The quantity of nodes in one system can be any (n). In model with transmission of messages each node can make a number of calculations and transfer messages to other nodes in this system.

There can be only two nodes in the simplest distributed system: client node and server node.

Further there is the way of their interaction. The client sends a command to the server which should proceed the received commands and send the confirmation to the client. If receiving was not confirmed for the specified time, the node client sends a command once again. The classical operation algorithm of a client-server system with the probability of distortion of information consists in this simple process. This algorithm can also be adapted for several servers at the same time. The command will be considered if the client receives confirmations from each server.

According to the described algorithm, the commands received by servers from different clients are processed in random order depending on time of their receiving. As a result of a command in several nodes servers are carried out

ISSN 2305-7254

randomly and have no dependent sequence. In some cases it may become a problem when there is no opportunity to control the proceeding processes and because of that their meaning is lost.

The concept of replication was entered to solute this problem. The point is that commands on all nodes are executed in the same order. The concept of replication of commands is one of key in science of distributed systems. It formed the basis of “blockchain” technology and became a synonym of this process.

There are two servers used for automatic achievement of such result. They are assigned the parallel-to-serial converter or the serializer are used. The command is sent by the client and automatically processed by the serializer. Then it is sent to other servers, further the confirmation of command is sent to the client. Thus, the serializer is some kind of intermediary between clients and the server and help a node client not to send messages to each of nodes servers. That allows to minimize possible losses in case of failure of one node.

Nevertheless, many specialists consider that using of the serializer is unreliable while using of the two-phase protocol leaves a possibility of recovery of nodes after the failure. The idea of the two-phase protocol is applied in several variations which have insignificant differences, in spite of that there are two different concepts. The first phase is called preparation for a transaction, the second – a transaction committed/aborted. Feature of this process is that it is started by the coordinator.

The algorithm Paxos is also used for the systems of unreliable calculations. It consists in obtaining result by the group of servers. Depending on it the offered command is carried out or deviates. As a rule, the proposers, the acceptors and learners participate in this algorithm.

The two-phase protocol is used for years. Despite that its technology is not perfect as there is no alternative to compare and improve it. The Paxos was offered L. Lemport in 1989 for the first time. Later this algorithm slightly changed and became simpler, but there is no alternative which could be compared to this development.

It is important to remember that there is no compromise reached in the process of using Paxos. The events are not accompanied by time in the asynchronous model. Thus each message will have a delay on more than one temporary unit. At the same time the maximum delay will not be able to be considered by the algorithm as it needs to work irrespective of time delays. All this will lead to failure in work of asynchronous model.

This problem reflects the need of searching of the algorithms as there is an opportunity to reach the consensus. Let's assume that the system has a configuration C which includes all dependent nodes of a system and all messages which are transferred between them. The configuration can be univalent or bivalent depending on the quantity of the events. Different configurations can integrate in the system (tree) of the configurations. In this case there will be the transitions between configurations at the edges of this tree.

One of the methods of improvement of algorithm’s work is the shared coin. It consists in introduction of this command to the code. This is used for acceleration of an operating cycle of the search algorithm to reach the consensus. This concept was offered by G. Brach in 1987.

Discussing the problems of work of distributed systems, it is necessary to specify a problem of random behavior of nodes. It is accepted to call such nodes "Byzantine" or incorrect. They do not transfer the message at all, or transfer the wrong messages. Such nodes cannot work separately. it is set of several faulty nodes as a rule. Byzantine agreement is a search of consensus in a system with the Byzantine nodes.

The main danger of the Byzantine nodes is that it is extremely difficult to recognize them as externally messages from them are correct, but they have wrong content which occasionally cannot be revealed quickly.

For the first time the problem of the such error was studied by scientists of NASA within the SIFT project. The result of these researches was the representation of an algorithm of the Byzantine agreement. Also it was proved that it is impossible to solve this problem for some algorithms. Today there is a set of algorithms and protocols for fight against a problem of the Byzantine nodes.

The Zyzzyva protocol is one of the solution. The principle of its work is that it will allow to accelerate work process if the nodes are correct and to slow down it for correction of mistakes. It allows to generate the order of tasks that were entered by the client. One more task of the considered protocol is the evidence that the node is Byzantine.

The safety is one of the main operating conditions of the system. If the system is safe, the executed commands cannot be reset or mixed.

The main feature of the Zyzzyva protocol is that its use improves productivity in case of lack of mistakes. At the same time there are protocols which do the same thing in case of mistakes.

There are situations when one server cannot service all their clients because of decreasing its power. The Quorum-system is applied in such cases.

Quorum is a subgroup of the overloaded nodes. The simplest Quorum system consists of one subgroup with one server. Such system is called single.

Historically, quorum is a minimum number of members in the council to create the business by group of people. This concept passed into computer science in the late 1970th.

On an equal basis with nodes, such systems can be incorrect or Byzantine. The feature is that being Byzantine, the node becomes more difficult in work. The work with them considerably becomes more simple, when they pass into discharge of quorums. Nevertheless, they continue to be incorrect.

Despite it, work with these nodes will become much safer, if quorums are used. The Byzantine nodes are dangerous not only because of high risk of failure, but also because of the possibility of data falsification. There will be one correct node in subgroup which will help to recover the reliable information.

Quite often there are problems with connection in the course of work with distributed systems. Network division is considered as the most common problem - it is failure at which the network is divided into two parts between which it is impossible to set connection. Any uncommon distributed system is not capable to continue to work in the normal way.

There are some of the most common causes of such gaps:

1) Physical shutdown;
2) An error in the work of the operating system;
3) Incompatibility of protocols that are used.

In terms of nodes, the gap is the same, as loss of the transferred message.

The distributed system will be able to work if the following conditions are satisfied:

1) Sequence. All nodes in a system are agreed with its current status;
2) Availability. During its functioning the system processes all incoming requests instantly;
3) Possibility of the gap. In case of a gap the distributed system has an opportunity to continue to work correctly even at the divided network.

The main feature of this principle is that the distributed system can carry out only with two of these three conditions at the same time.

If there are no updates are offered for a system, the system remains in a current status and the message between nodes stops. Such phenomenon is called final coherence.

The cryptocurrency “Bitcoin” system is the example of such a phenomenon.The Bitcoin network – the network which is randomly connected to the thousands of nodes controlled by the owners. All nodes carry out the same operations. The network is homogeneous so it is not necessary to control it all the time.

The users can generate any quantity of personal keys from which the public key indicating the owner will be received. Couple of such keys (personal and public) allow to identify the owner of the address. The point of using the address is that its interpretation is shorter, than a personal key.

The Bitcoin network spontaneously keeps track of quantity of bitcoins on balance of each address.The Bitcoin network was developed by Satosi Nakamoto and presented in 2008 for the first time. The name of the developer is the nickname, so the real name or names of founders of this network are unknown. This development was presented during the global financial crisis therefore the Bitcoin system very quickly gained popularity.

It is possible to compare a virtual system in which nodes change every second to classical distributed system. They are included into this network and leave in others, they also have the information about the status of nearby nodes, but not all system in general. Such distributed system is called a virtual area network or overlay network.

Its main features are:

1) Homogeneity. There is no main node;
2) ID existence;
3) The small number of connections at each node of a system;
4) Small diameter of network, simple routing.

Blockchain is a continuous set of linked blocks, which keep an information built in accordance with settled rules. Every node in the block system stores independent copies of the block chains.

A new-created block keeps an information about ordered records (transactions), and an initial record (header). Once a block has been created, it is verified by other network nodes, and if every node confirms the legality of a block adding, connects it to an end of the chain then. Any block modification becomes impossible after it. The database record is updated to all connected nodes and devices of the network automatically.

An information about any event of a human being’s activity, like buying a real estate or vehicle, taking a credit loan, changing of his personal status, etc., is taking and kept by government information storage systems, or business companies ones. It particularly may cause to a hacker’s attempt to get an un authorized access to a database and make undesired information modification to it. The blockchain technology makes possible to amend the access to an information stored, fully changing the approach to it, because if a system is built on blockchain technology, the data is stored not in a single server, but is widely distributed among lots of devices. A chance to have all servers in a blockchain network disable simultaneously is really small. Until an only one network server is on duty, a network system based on the blockchain technology is still working.

As was marked already, any single center database can be get under attack and lose information integrity. Blockchain technology makes it impossible. A single block corruption really makes no sense - all the blocks must be attacked and modified, as well as all copies of the data base on all devices connected, and it needs big computing resources to be engaged to. moreover, a cipher encryption algorithm which operates with hash functions and an Electronic Digital Signature (EDS) would prevent a fraud activity. EDS operates with two keys - open key and closed key. An open key is needed to confirm the signature itself, and a closed key is used to settle a digital signature and is kept secretly. The hash function makes confirmed that all data recorded are kept uncorrupted.

III. Software implementation

A blockchain network is created by persons which want to use and store some certain data. There are two types of network participants:

1) Regular users;

2) The creators of the blocks (miners).

Regular users can create new records on the network, they are called transactions. For instance, "transfer to the user 20 conventional units", and miners create a block from these records. Records are verified and added to the block after majority confirms it only. Invalid records that can't pass the test are ignored and considered as unreliable. The owner of the key that opens access to it can use this record only.

Blockchain technology is intented to form different types of systems. Among them can be found public systems with easy access, where any users and miners can join. Besides, there are private network shutdowns, which can be supported and controlled by developers only. To become a private network member, it is necessary to fulfill certain conditions settled by organizers. True delimited users only can create new blocks in private systems.

Using the blockchain account, people have the opportunity to make transactions in any sphere of life: real estate transactions, financial transactions, transportation organization, etc.

The main disadvantage of this system is the scalability. Nowadays, the system is not being able to provide the large amount of transactions in a short time. The bitcoin can operate only 7 transactions per second, while the Visa or MasterCard systems are able to deal with more than 50 000 transactions at the same time. Moreover, the database increases daily, which should be stored on the computers of each user of the network.

More than that, experts pay much attention to the possibility of a so-called "51% attack" on the blockchain system. This attack implies that if the participants of the network occupy 51% of the computing power, then members of this group will be able to start acting in their own interests, for example, to confirm only profitable transactions for them. However, it will not be able without powerful resources, so it should be extremely difficult to implement this idea.

For the creating of the protocol, the programming language Python 3.6.5 was used with the built-in libraries Flask and Requests. To test the correct operation of the program, was used the HTTP client Postman.

The creating of a protocol for transferring the information can be divided into the following stages:

The first stage is the creating a Blockchain Class. This step begins with the generating an empty list and another class for storing transactions.

```
class Blockchain:
def __init__(self):
self.current_transactions = []
self.chain = []
self.nodes = set()
self.new_block(previous_hash='1',
proof=100)
def register_node(self, address):
parsed_url = urlparse(address)
if parsed_url.netloc:
self.nodes.add(parsed_url.netloc)
elif parsed_url.path:
self.nodes.add(parsed_url.path)
else:
raise ValueError('Invalid URL')
@property
def last_block(self):
return self.chain[-1]
```

The first example shows a part of the code where a method is implemented to create a new block and add it to the chain, a method for adding a new transaction to the transaction list, a method for hashing blocks, a method that returns the last block in the chain.

The class which is responsible for the chain management is a blockchain class. This can help to safe transactions and other methods for adding blocks in a chain. Each block contains index, the time mark, the list of transactions and the previous block's hash. Each block is presented in the JSON size.

There is the storage in each block of the hash function of the previous block that makes the constancy of the chain of blocks possible. If an hacker tries to break the one of the initial blocks, the entire block sequence will contain an incorrect hash.

The new_transaction () method lets the user to add transaction to the block:

```
def new_transaction (self, sender,
recipient, info):
self.current_transactions.append({
'sender':sender,
'recipient': recipient,
'info':info,})
return self.last_block['index'] + 1
```

This way adds the transaction to the list and returns the index of the block which is used for adding the new transaction. After the creating the new block it is necessary to fill it with the source block which was used at the beginning. The methods new_block and the hash () were used for creating the source block in the block class.

```
def new_block(self, proof,
previous_hash):
block = {
'index': len(self.chain) + 1,
'timestamp': time(),
'transactions':
self.current_transactions,
'proof': proof,
'previous_hash': previous_hash or
self.hash(self.chain[-1]),}
with open(blockchain_dir + 'block.txt',
'a') as file:
json.dump(block, file, indent=4,
ensure_ascii = False)
self.current_transactions = []
self.chain.append(block)
return block
```

The second stage is the development of the Proof-of-work algorithm.

The Proof-of-work algorithm is an algorithm which allow to create the new blocks in a chain. It is necessary to find the number which can become the problem solution. The number should be difficult to find and easy to check at the same time.

The algorithm works with the principle of finding the number y which results in a hash ending 0 after the multiply by the hash of x. The changing of the number of conditions influences on the time and complexity, because of that it is necessary to choose the level of complexity of the calculation from the security requirements.

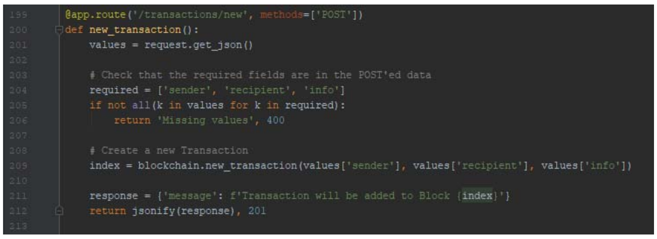
In this article the number p is the hash with two leading zeros which was hashed with the solution of previous block.

```
def proof_of_work(self, last_block):
last_proof = last_block['proof']
last_hash = self.hash(last_block)
proof = 0
while       self.valid_proof(last_proof,
proof, last_hash) is False:
proof += 1
return proof
```

The third stage is the setting up the blockchain as an API. The platform Flask is used for accessing to the block system through the web-connection. In this example the three methods are used for the work with the blocking.

- /transactions/new – create the new transaction in a block;
- /mine – the information about the necessity of “mining” block;
- /chain – an output of the whole blockchain.

The function for adding transactions should check up that all of the fields are submitted in the request and after that the new transaction is created.

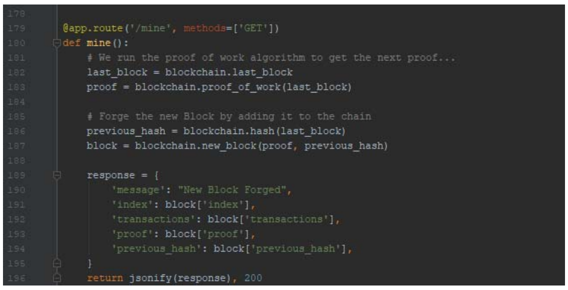


```
@app.route('/transactions/new', methods=['POST'])
def new_transaction():
    values = request.get_json()

    # Check that the required fields are in the POST'ed data
    required = ['sender', 'recipient', 'info']
    if not all(k in values for k in required):
        return 'Missing values', 400

    # Create a new Transaction
    index = blockchain.new_transaction(values['sender'], values['recipient'], values['info'])

    response = {'message': f'Transaction will be added to Block {index}'}
    return jsonify(response), 201
```

Fig. 1. Method of creating a transaction

The considered blockchain system is used for the secure connection of the devices of the Internet of things, so the extra graphs for this task is not necessary.

The block mining function begins with the PoW algorithm, then forms a new block by adding it to the chain.



```
@app.route('/mine', methods=['GET'])
def mine():
    # We run the proof of work algorithm to get the next proof...
    last_block = blockchain.last_block
    proof = blockchain.proof_of_work(last_block)

    # Forge the new Block by adding it to the chain
    previous_hash = blockchain.hash(last_block)
    block = blockchain.new_block(proof, previous_hash)

    response = {
        'message': "New Block Forged",
        'index': block['index'],
        'transactions': block['transactions'],
        'proof': block['proof'],
        'previous_hash': block['previous_hash'],
    }
    return jsonify(response), 200
```

Fig. 2. Method of forming a new block

The forth stage is testing of the correct operation of the blockchain.

For this step the Postman program is used. The Postman program is the client HTTP, which was developed for testing the web-sites. This program may help to compose and edit simple or complicated HTTP requests. The generated requests are automatically saved for the reuse in the future. There is also the built-in query editor with the ability to encode queries, load from different files, and send binary data.

It is necessary to use the GET request to the address, in order to “mine” the block http://localhost:5000/mine.

The POST request gives user an opportunity to add a new transaction:

http://localhost:5000/transactions/new with the body containing the transaction structure:

```
{
"sender":"42b1f3bb5677454ab9c07919aa
95b2oo",
"recipient": "Someon111e",
"info": "Some info from device"
}
```

By sending a GET request to the address http://localhost:5000/chain, user gets the whole chain of blocks contained in this blockchain.

The last stage is an addition of new nodes to the blockchain.

There are two methods which allow implementing blockchain on multiple devices:

•/nodes/register – registration of new nodes;

•/nodes/resolve – algorithm for solving conflict situations, so that each node contains a correct chain.

Conflict situations – situations, when the one node contains the chain which is different from the same chain in the other node. For solving this problem there is the rule that the longest chain is correct, so it is used in the whole process.

For the chain test, the valid_chain () method should be used. It checks the entire block and compares the hash and proof of each block.

The resolve_conflicts () is a method that traverses all neighboring nodes in a loop, downloads node chains and checks them using the valid_chain method described above.

IV. Conclusion

In conclusion, it is necessary to say that the issue was made of the short historical node about the development the system which became the principle of blockchain, particular qualities of its work. As a result, the basic principles of the functioning of the blockchain system were considered. All the features were described with the specific example.

The blockchain technology is one of the most perspective spheres of the web-science development nowadays. IT corporations all over the world pay much attention to this area as it could result in a radically different competitive future for the variety of services.

It is too early to tell exactly where the cryptocurrency and blockchain system will end up. But to get there, it is extremely important to remember that without users, there is no network effect.

So in the next future it is going to develop actively and to affect all spheres of the human life.

Research objectives are achieved with the system method of creating the blockchain protocol. The problem of the topic was successfully solved during the topic.

References

[1] Gaston C. Hillar Internet of Things with Python. Packt Publishing. 2016.

[2] Dirk Slama, Frank Puhlmann, Jim Morrish, Rishi M Bhatnagar Enterprise IoT. O'Reilly Media. 2015.

[3] Dawid Borycki Programming for the Internet of Things. Microsoft Press. 2017.

[4] Bob Familiar Microservices, IoT, and Azure. Apress. 2015.

[5] Gorodnichev M.G., Nigmatulin A.N. Technical and program aspects on monitoring of highway flows (case of Moscow city)

[6] Roger Wattenhofer. The Science of the Blockchain. 1st edition. CreateSpace Independent Publishing Platform. 2016.

[7] Malenie Swan. Blockchain: Blueprint for a new economy. 1st edition. O'Reilly Media. 2015.

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

[9] Antony Lewis. The basics of bitcoins and blockchains: an introduction to cryptocurrencies and the technology that powers them. Mango. 2018.

[10] Daniel Frumkin. Understanding blockchain: learn how blockchain technology is powering bitcoin, cryptocurrencies, and the future of the Internet. Independently published. 2018.