International Journal of Engineering & Technology, 7 (4.36) (2018) 72-75

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Combination of Technology Blockchain and Multiversion Concurrency Control for Developing Database with Peer-to-Peer Atomic Safe Transactions

I.A. Ivanova[1], I.D. Kotilevets[2], E.S. Karbova[3], S.A. Pavelyev[4]

[1]*MIREA - Russian Technological University,Russia*
[2]*MIREA - Russian Technological University,Russia*
[3]*MIREA - Russian Technological University,Russia*
[4]*MIREA - Russian Technological University,Russia*

Abstract

The article discusses the mechanisms of consensus algorithms used in a blockchain without connection with crypto-currencies, compares with the functionality of competitive access control technology to the database, and combines the functions of blockcahin and MultiVersion Concurrency Control in databases to obtain peer-to-peer atomic safe transactions of strictly defined "facts", written down in an irrefutable audit journal.

Keywords: *technology, bitcoin, blockchain, algorithm, ϕnalysis*

1. Introduction

In recent years, the concept of blockchain technology has become popular. A blockchain is a ledger with "facts" that is replicated to several devices that are connected to a peer-to-peer network. Usually a money transaction is understood as "facts" (example - the Bitcoin crypto-currency).
However, "facts" can represent any information, for example, information of a specific registry, the signing of content etc. "Facts" are grouped into blocks and the only true chain of blocks is replicated to the network, where each block refers to the previous one. Before they are added to the block, the facts are pending, i.e. not confirmed.
Blockchain is the transfer of two types of messages - transactions and blocks (which are lists of transactions). Transactions are formed by participants of the system, and blocks are the main product of the consensus algorithm and determine in which order the transactions will be included in the journal1.

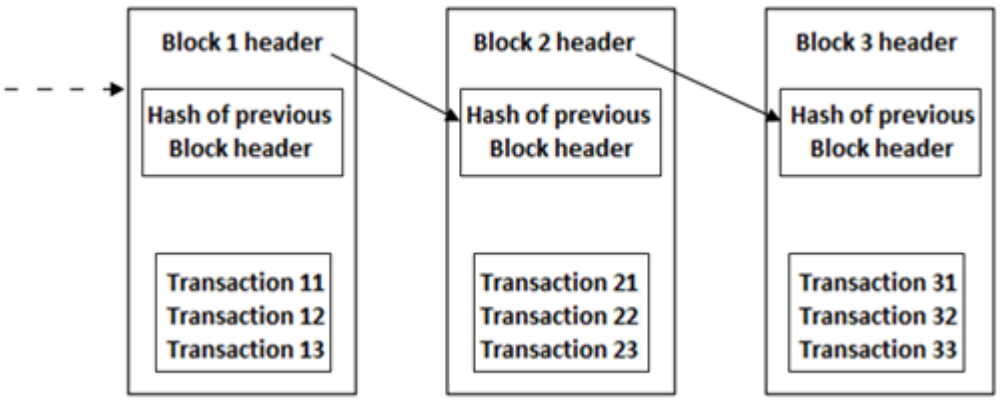


Fig. 1. Simplified Blockchain

The consensus algorithm allows building reliable systems from unreliable parts. This allows to use this algorithm in fault-tolerant databases, group systems etc.
Consensus must meet the following requirements:

1) Consistency: all the corrected working nodes must take the same value;
2) Correctness: the selected value must be one of those that was suggested by some properly working node;

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

3) Finiteness: every correctly functioning network node must make a choice in a finite number of steps.
Consensus algorithms also appear in the context of the use of distributed finite state machines, which allows the system to run the identical command while in the same state, and continue to work even if part of its servers fail. Thus, distributed state machines are used to solve problems of fault tolerance in distributed systems2.
Consensus algorithms have the following properties:
1) Ensuring security;
2) Functioning until most nodes can interact with each other;
3) Do not depend on time to ensure consistency of the journals;
4) The completion of the command occurs when most of the nodes responded with a successful acknowledgment of the execution of the command.
The consensus mechanism in the blockchain ensures that distributed registries are exact copies, which reduces the risk of fraudulent transactions, since extraneous interference can occur in many places at the same time. Then in the blockchain-network all participants are equal and are connected by the same protocols, and the system records the chronological order of transactions with all nodes of the network that recognized the validity of transactions through the chosen consensus model and using a hash function. The result is non-cancellable transactions agreed upon by all network participants in a decentralized manner.
In ordinary consensus algorithms, network nodes have certain identifiable attributes expressed through digital signatures, and the list of nodes is known in advance or varies rarely, but predictably. In the blockchain it's the other way around. The participants of the network are not only unknown in advance, but they are also free to connect or disconnect from the network. Because of this, the usual consensus algorithms for the blockchain are not suitable. Therefore, many different algorithms have been proposed, the most popular ones are: Proof-of-Work and Proof-of-Stake. Today other consensus algorithms in the block are actively developing, for example, Proof-of-Importance, Proof-of-Burn, Proof-of-Space and so on. There are many areas of application of the technology in various system, so different companies from many spheres are interested in developing blockchain.
The use of blockade technology3:
1) Storage on the distributed cloud;
2) Identity management;
3) Registration and verification of data;
4) Automatic execution of contracts;
5) Tracking of supplies and proof of origin;
6) Notary services;
7) Automated security;
8) Lease of property and collaborative economy;
9) Voting on the Internet;
10) Electricity market without intermediaries;
11) Application in the media;
12) Application in the military sector;
13) Decentralization of the Internet of Things;
14) Application in the field of insurance;
15) Use on the Internet.

2. Research Method

To ensure data integrity and security competitive database access control technology is used. There are many variants of this technology, but the following principle unites them: operations on a database are grouped into "transactions", which are processed atomically. Competitive access management maintains consistency of data by blocking parts of the database for the duration of the transaction so that other transactions cannot at the same time access the same data.
An effective competitive access control scheme allows parallel operations, blocking as little data as possible for as little time as possible. One of the popular methods of parallel processing of transactions is called multiversion concurrency control, or MVCC. In the MVCC each transaction consistent snapshot at a certain point in time, even if some of the data is processed in parallel by another transaction. This property guarantees, for example, that statement showing the total balance across multiple accounts will always be correct even if some amounts are currently being transferred from one account to another4.
One transaction will affect the data available to another transaction only if the second transaction begins after the successful application of all changes within the first transaction. Importantly, MVCC prevents conflicts of write operations. We can say that the blockchain system plays the role of a distributed MVCC system. Blockchain performs the function of a unified mechanism for detecting and preventing conflicts throughout the network. If two transactions try to delete the same version of the line, then only one of them will be accepted. Managing competitive access with the help of multi-versioning functions as a unified mechanism for detecting and preventing these conflicts in the database.
When mentioning the multiversion concurrency control, it cannot go without the definition of ACID (Atomicity, Consistency, Isolation, Durability)5
Consider this abbreviation in more detail:
1) Atomicity - either all changes to the transaction are fixed, or all are rolled back;
2) Consistency - transactions do not violate the consistency of data, they translate the database from one correct state to another;
3) Isolation - transactions that works at the same time do not affect each other, multithreaded transaction processing is performed in such a way that the result of their parallel execution corresponds to the result of their consecutive execution;
4) Durability - if the transaction was successfully completed, no external event should result in the loss of the changes it has made.
Examples of databases using these technologies - MongoDB, Cassandra, RethinkDB, PostgreSQL. All of them are able to work with a large number of replicas that are clustered together. The client works with one of the replicas, and the data is automatically synchronized with the others. Also shading can be used for load balancing, when part of the data is stored only on a part of the replicas. Adding a new replica to the cluster linearly scales the cluster, and certain implementations allow the replica to automatically assume part of the cluster work.
For example PostgreSQL uses consecutive transaction numbers6. These numbers determine the order of events.
The unit of multiversion is the rows of tables. The table block contains a set of line versions (tuples), for each of which the numbers of two transactions are stored: initial (xmin) and final (xmax). When row is inserted, the transaction number is written to it as the initial transaction number. When a row is deleted, it is not erased from the block. Instead, the number of the deletion transaction is written as the final. Updating a line works just like deleting and inserting a new one. Thus, within the same block there can be different versions of the same line, and it is known when the version appeared and when it disappeared.
Index blocks do not contain any versioning information, index entries refer to each version of the rows.
The system has a list of all transaction statuses (CLOG). Fixing or canceling transactions is performed by changing the status in this list. The initial and final transaction numbers in the lines can become irrelevant, for example if the transaction was canceled. Affected blocks are not corrected immediately, but later checking with CLOG. Correcting affected blocks immediately would be too costly.
A snapshot of the data consists of the nearest transaction number (which determines the current time) and a list of active transactions that are not completed at the moment. When a transaction reads data, it must see in the block only those lines that have already been committed and have not been deleted at the time the snapshot was created (as well as the lines created by the transaction itself). The information in the picture and lines is just enough to calculate this condition.

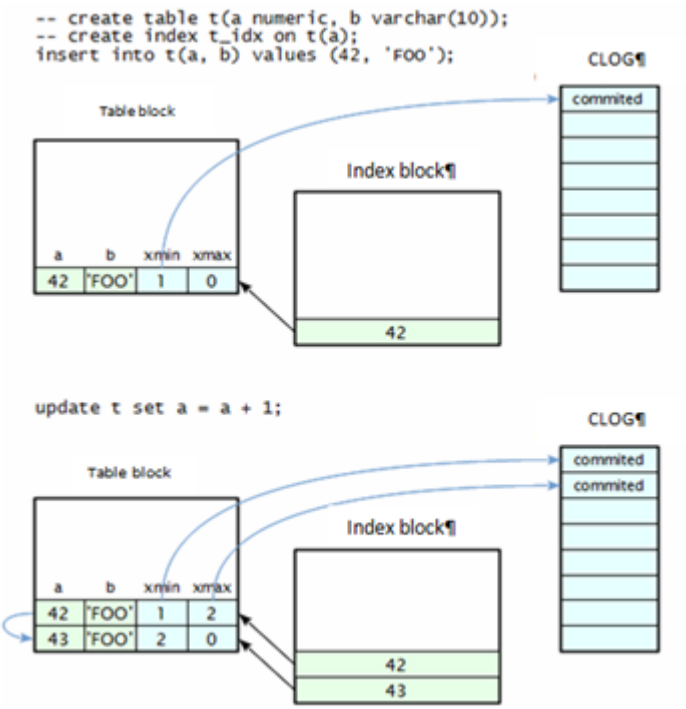


Fig. 2. Example of MVCC in PostgreSQL7

Also PostgreSQL, MongoDB, Cassandra, RethinkDB provide "Eventual Consistency" - the data becomes consistent after a time when individual replicas are synchronized. For this reason they are similar to blockchain - confirmation of the transaction is more likely, the more time has passed8.

Apparently, the ACID requirements correlate with the requirements of consensus algorithms. Moreover, comparing the functionality of MVCC and the blockchain, you can see the similarity. Repeating the above:

If two transactions try to spend one output, then only one of them will be accepted. Blockchain performs the function of a unified mechanism for identifying and preventing these conflicts throughout the network.

If two transactions try to delete the same version of the string, then only one of them will be accepted. Multiversion concurrency control functions as a unified mechanism for detecting and preventing these conflicts in the database.

3. Results Abd Analysis

Continuing to expand this idea, we can consider the blockchain-node (for a specific example, the bitcoin-node) as a set of unspent transaction outputs, then the database will be a table in which each string is one unreleased output. It turns out:

1) Each transaction changes only a few strings at a time.

2) The size of each string in the database is quite small.

In such case a transaction does not simply point to the outputs of some previous transaction and creates new outputs, but also performs two additional functions necessary to prevent the increase in the size of the blockchain transaction.

First, the rules of transaction verification contain applied logic of the bitcoin database, the requirement that the total amount of funds at the inputs of the transaction should exceed the total output. In other words, bitcoin-transactions cannot increase the total number of bitcoins in the database. Such a restriction exceeds the stored procedures of a regular database, because it cannot be violated under any circumstances.

Second, each bitcoin-transaction output encodes the conditions under which it can be spent. For normal outputs, these conditions are based on public key cryptography: the public address is embedded in the "output script" so that it can be spent only with a private key corresponding to that address. In analogy with databases, this output corresponds to a string with cryptographic permissions set for it. Moreover, each transaction provides an open confirmation that its creator had the right to delete or change the previous strings.

The combination of MVCC and blockchain can give the following properties for the database:

1) The database is public, the user is identified by the public key, which is the user ID.

2) Each user can send transactions to the database, each transaction must be signed by this user.

3) A new record created by the user remembers that it is its owner.

4) Only the owner (or a user for whom certain permissions are installed through various mechanisms, for example, a smart contract) can change the record after creation.

5) Everyone can read all the entries.

6) To prevent conflicting keys of their records between different users, all the keys of user records are prefixed with: user ID.

7) All permissions are checked for both transactions and replicas.

4. Conclusion

Integration of blockchain and multiversion concurrency control allows achieving high data safety in many spheres, from the financial sphere to the Internet of Things, due to the fact that in such a database at the lowest level it is guaranteed that "facts" (money, data, etc.) are not can be created from nothing. Moreover, the use of peer-to-peer atomic safe exchange transactions makes it possible to avoid identification of the exchange side.

Also, an audit log will play an important role in building the system based on these technologies, showing that each transaction was authorized by the owner of the "facts". Moreover, this audit log is irrefutable. As a result, at the output we can get a database that satisfies the following properties:

1) Distribution. The database supports an unlimited number of replicas, each of which can be a coordinator. Referring to one of them, the user gets access to all data.

2) Publicity. The database is designed to work in a public environment. New nodes can be added to the network and take on some of the load at any time.

3) Resistance to the problem of Byzantine generals and other types of attacks in the public network9. Given that all data placed in the database is signed by their owner, the nodes cannot change the

data at their discretion, nor can they corrupt data when replicating on other nodes. Attempts to substitute are immediately detected using the electronic signature mechanism. For attempted substitution, the offending node may be stripped of the registration deposit and expelled from the network.
4)Speed. Principles of data storage suggest that the speed of writing and reading data in database will not be very different from the current implementations of such databases.
5) Ability to store structured data. Data in database supports the structure. This can be a document with a structure that is user-friendly.
6) Ability to delete data. Data removing is supported. Cannot be guaranteed instant removal, but in the end, with good behavior of nodes, the data will be deleted. A malicious node can deliberately store all data that is deleted. However, it will not be able to do this for all data, because it only receives requests in a certain range of primary keys.
This work was partially supported by motivational payments system faculty MIREA10.

References

[1] Author,"Title of the Paper", *Journal name*, Vol.X, No.X, (200X), "Consensus Algorithms": Proof of share and proof of work. (Electronic Materials) - https://habrahabr.ru/company/bitfury/blog/327468/
[2] Mechanisms for achieving consensus in blockchain. (Electronic Materials) - https://geektimes.ru/company/waves/blog/286896/
[3] Blockchain – an opportunity for energy producers and consumers? (Electronic Materials) - https://www.pwc.com/gx/en/industries/assets/pwc-blockchain-opportunity-for-energy-producers-and-consumers.pdf
[4] Voit A, Stankus A, Magomedov S, Ivanova I 2017 Big Data Processing for Full-Text Search and Visualization with Elasticsearch Int. J. of Adv. Comp. Science and App. (DOI: 10.14569/IJACSA.2017.081211) 8/12:76-83
[5] Wu Y, Arulraj J, Lin J, Xian R, Pavlo A 2017 An Empirical Evaluation of In-Memory Multi-Version Concurrency Control Proceedings of the VLDB Endowment 10/7:781-92
[6] Zendaoui F, Hidouci W K 2015 Performance Evaluation of Serializable Snapshot Isolation in PostgreSQL 12th Int. Symp. on Prog. and Sys. (ISPS'2015) (DOI: 10.1109/ISPS.2015.7244971)
[7] MVCC in Oracle and PostgreSQL. (Electronic Materials) - https://postgrespro.ru/blog/pgsql/17758
[8] Holzschuher F, Peinl R 2016 Querying a graph database – language selection and performance considerations J. of Comp. and Sys. Sciences 82/1A:45-68 (https://doi.org/10.1016/j.jcss.2015.06.006)
[9] Popov G., Magomedov Sh. Comparative analysis of various methods treatment expert assessments. International Journal of Advanced Computer Science and Applications, 2017, vol. 8, no 5, pp. 35-39. (DOI: 10.14569/IJACSA.2017.080505)
[10] Pankov V The effectiveness of incentive mechanism, and the potential level of satisfaction of the needs of the employee. Russian Journal of Technology. 2015. № 4. p. 288-91 (in Rissian).