# Theory TERA-sharding

Yuriy Ivanov (Vtools)
June 29, 2019
progr76@gmail.com
(ver: 0.01)

## abstract

concept blokcheyn and as a philosophy of decentralized transfer of values, it was deeply absorbed into society and it appeared the need to create such a global network that is completely decentralized and has no problems with scalability.

Based on the sharding protocol, you can create a network with millions of nodes and a total throughput of millions of transactions per second. The basic principle is that an unlimited number of equal blockchains are created in the network, combined with each other through a common hash network. Blockchains communicate directly with each other through cross-sharding transactions. Since there is no main network in the network through which

such transactions are carried out, there is no bottleneck, which allows for virtually infinite scalability. Network security is provided by the outer contour of the lightweight blockchain; it consists only of common hash headers. The network miners do the work of selecting such a common network hash and validate the affiliated blockchains.

## Introduction

To ensure maximum decentralization in the form of a virtually unlimited number of block producers, it is possible at the PoW consensus. But a simple partitioning of network nodes into shards in the form of individual blockchains, although leading to a multiple increase in overall performance, but also leads to a multiple loss of security. It becomes easier to attack 51% on each blockchain separately. Indeed, if we have broken the entire network into 100 mining shards that are equal in power, then to attack an individual blockchain, 0.51% of the power of the mining equipment of the entire network is sufficient.
To prevent such a threat, you can apply joint mining. In this case, miners count only the total hash of the network, the complexity of the hash remains the same and thus the security of the network does not deteriorate. But there is a problem how to properly organize the validation of shards, since resources of one node are limited (to guarantee mining decentralization, we will assume that one node can validate at most one shard).
The central idea of solving this problem is to use the social aspect of the miners - using the “theory of six handshakes”. Miners may have several nodes, miners may have connections (trusted friends).

## Analysis of the mechanics of sharding

In practice, a simple question arises - what should an ordinary ordinary user do if he downloaded a mining program that has only one computer and, accordingly, the resources for validation are only enough for one shard?

In the PoW consensus there is an interesting relationship between network security and time. The more time we spend creating the hash, the more secure the network and vice versa. This is something like a law that can not be circumvented, but which should be used. Hence the obvious conclusion: if there are not enough miners supporting the validation of the shard, then this shard will be less likely to have a confirmation block. Shards that have a fairly rare amount of confirmations will be less preferable for users, this will affect their market capitalization (since each shard is a separate blockchain with its own cryptocurrency).
It will look like this: the miner indicates the connection between his nodes and the shards that these nodes are validating. When mining,included in the winner's block tree **only the** hashes of the validated shards are. Users will consider the shard block as valid only if it is included in the shared network hash.
Thus, a miner who has only one node will validate only one shard, and a miner who has many nodes will have almost all shards. Because the probability of finding the blocks will be in the miner, who has more resources (nodes), then:

- If the network will be dominated by the nodes of professional miners with many nodes, then most of the blocks will have all validated shards.
- If the number of nodes (with conditionally equal powers) is equal, then on average all shards will be validated through one block.
- If there are more single miners, then the probability of validating a shard will be directly dependent on its popularity (and in the inverse of the total number of shards).

Thus:

- A new miner with a single node will be able to simply participate in mining, while he is motivated to validate other shards whenever he wants.
- To introduce a new shard, it is enough for it to be validated by at least one miner. The confirmation time for blocks of shards depends on the popularity and market efficiency of the shard.

All this together provides a smooth and organic mechanism for the formation and maintenance of the shard, which will be regulated by market methods - by changing the rate of the built-in cryptocurrency shard.

## Technical implementation

### Terms

**Network** - This term refers to a virtual (relay) network in the form of communication of nodes (computers) with each other in a specific sequence, grouped by different shards (blockchains) and working under the same data transfer protocol.
**Shard** - in this version of the document, a shard is understood as a separate blockchain, the peculiarity of which is that its validation is performed by the miners of the general blockchain.

### Coins

Each shard has its own built-in coin (cryptocurrency), cross-sharding transactions - these are actually cross-sharding swaps. No global cryptocurrency. Exchange of cryptocurrencies from different shards is possible only when creating smart contracts in the form of decentralized exchanges, the rate of which depends on supply and demand.
To use a single cryptocurrency, such as Thera, each shard can implement payment channels in the form of smart contracts that change one Tora from token from one shard to another tora from Thera in another shard (the token id depends on the smart contact itself, which Tera considers) . The overall balance of coins and tokens inside the shard remains constant.
This approach provides financial security for users of shards, if we assume that one shard is compromised, the other shards will not suffer.

## A node

is a basic network unit. All nodes are equal. They are combined with each other according to certain rules and form the so-called relay network, which allows you to optimally exchange information in the network. Nodes are not connected directly to each other, each node has a limited number of connections - it is proportional to the logarithm of the number of nodes in the entire network.

Functions of the node as a network participant:

1. General network information:
    a. Calculation (mining) and data exchange of the network's common hash
    b. Time
    c. synchronization Synchronization of code versions
2. Validation and data exchange of the shard (blockchain). Standardly, a node can validate only one shard.
3. Validation and data exchange of cross-sharding transactional buffers (maximum number is limited)

## Network creation rule

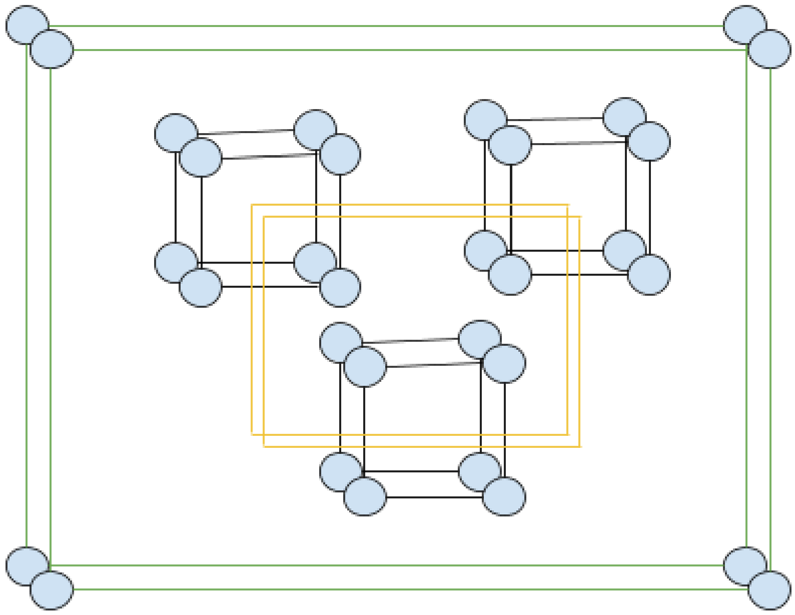
In order to efficiently transfer both common and intra-smart information within the network, several exchange paths are created. Such contours have the form of a multi-dimensional lattice. Since the operation of each node in the network is not guaranteed, the connections are of a dynamic nature - they are constantly maintained by connecting to other nodes and have a reserve.

The dynamic structure in the form of a multidimensional regular lattice is built on the basis of the similarity of node identifiers - depending on the degree of such similarity, its dimension number in a multidimensional cube is determined; this number will be referred to as the level of information exchange. Since many nodes can claim the same level, the following order exists:

1. At each level, in order of priority, the nodes of the current shard are first added to exchange shard transactions.
2. In parallel, at each level, nodes of other shards are added to exchange buffers of cross-sharding transactions.
3. If the level is empty (or the node is not sufficient), then nodes from the general list are added to exchange common network information.

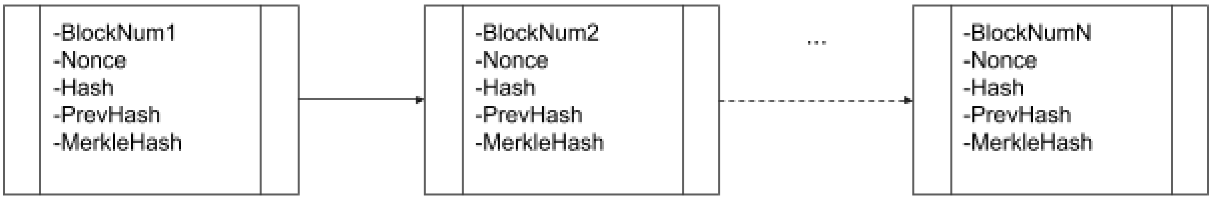
General information is contained in all nodes, so the rules indicate first filling levels with nodes with unique information, and then with more general ones.

Thus, the nodes will be connected to an external multidimensional cube to exchange common network information and internal cubes to exchange information within each shard.



## The structure of common network blocks The network

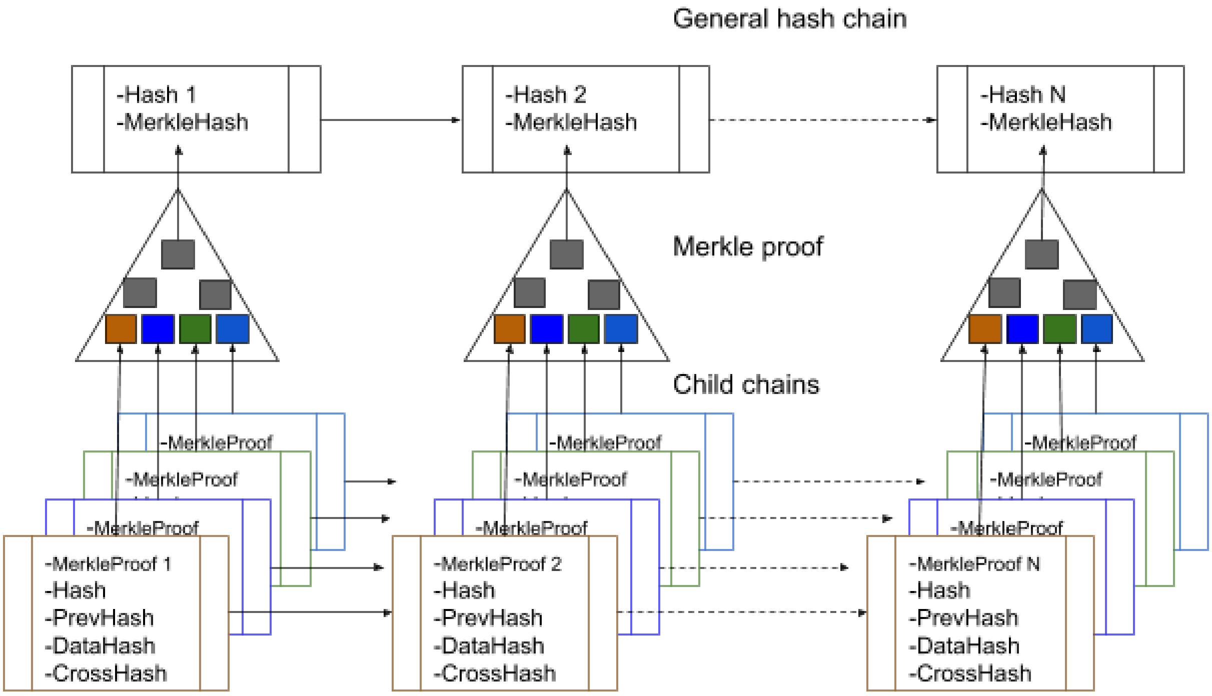
is a blockchain blockchain. The top level blockchain consists only of shards hashes. His goal is to cement the states of hashes of shards - the impossibility of rolling back blocks. The construction of this blockchain is classic: the hashes of the shards are grouped into blocks, the miners must calculate the hash of the new block which depends on the previous hash of the block (PrevHash), the number of Nonce and the hashes of the shards included in it (MerkleHash).



The header of the total block is 128 bytes, the size for the year will be 4 GB.

Miners who have found a common block should send to those shards that they support the Merkle proof - information on the connection between the hash of the shard and the common hash of the network. Technically, this is simply implemented due to the fact that the miner has an accurate list / tree of shards that he supports.

## Connection scheme of affiliated blockchains with a general



## Block formation rate The

consensus is based on the PoW algorithm, but since blocks are formed once per second, the concept of, is not used *target*, which is used in the classic Bitcoin blockchain and its copies. Instead, the job of finding the right hash is performed for exactly one second, the hash with the greatest complexity is sent to the network. When searching for a hash leader, the hashes are compared to each other in the network and are selected with the least amount of zeros from the left. numerically smaller.

## Horizontal scaling with joint mining

With a large number of shards (for example, thousands and hundreds of thousands), the validation of all shards by one ordinary miner is almost impossible - one node can check only one shard, and the number of nodes of the ordinary miner is limited.
Horizontal scaling allows you to increase these capabilities as follows:

1. The first way is to create a cluster from your own nodes. To do this, in the settings of each node, membership in one cluster is set by entering a shared secret password. In this embodiment, the number of supported shards is limited by the number of miner's own nodes.
2. The second way is to include an external cluster of nodes that the miner trusts (for example, his friend) in the tree of trusted clusters. This is done by adding the public key of the external cluster and specifying the level of the trust hierarchy (can it be trusted in turn to its child nodes). With this option, the number of supported shards can be many times greater than the number of miner's own nodes. And you can specify the degree

Miner can combine both options to achieve maximum reliability and profit.

*User strategy option:*

1. add your nodes (5 pieces one shard)
2. add a cluster of friends 'nodes (2 clusters with one hierarchy level, ie only friends' nodes)
3. add a cluster of 100 shards of a well-known company such as VISA / MasterCard

### Miner

Motivation Miner Motivation - it is receiving awards in every shard, so it is stimulated to validate and include at least one shard in the header. The algorithm allows for the inclusion of zero shards in this case, the hash shards (ShardsHash) has a zero value.
In order to maximize profits, the miner must strive to validate as many shards as possible, because awards he receives independently in each of them. Since a node can validate only one shard, a miner can run several nodes with different shard settings for validation, and in order to include information on several shards in a block, he can combine his nodes into a trusted cluster.
The optimal miner strategy will be to search for clusters that he can trust to include in his mining list. Ideally, he will strive for 100% coverage of Thera's shard network.

### Chain Selection Algorithm

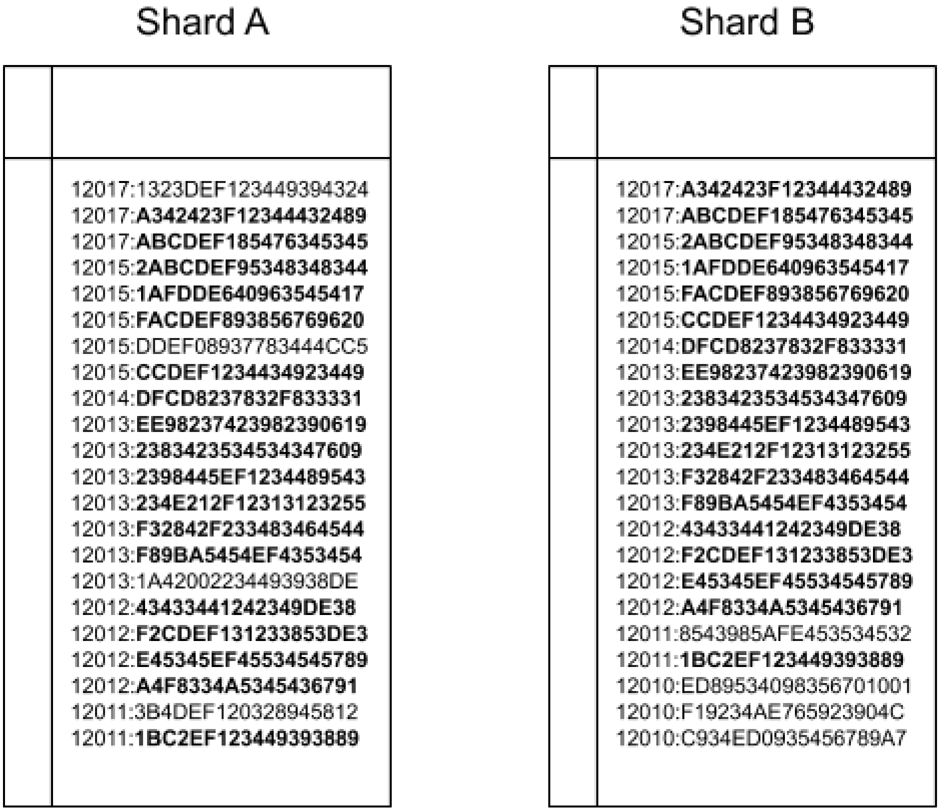
1. During the initial start, the full node as well as the light client loads the headers of the chains of the common network, determines the chain with the maximum amount of complexity of the hash, and further considers it as the main one.
2. After that, the nodes of the required shard are searched and the headers of the shard blocks are downloaded. They determine the chain of the required shard with the maximum amount of complexity of the hash (thus, protection from the fork is performed). Complexity is taken from common network hashes and only those blocks are added in which the block hash was included in the general header of the network block (checked through a merkl tree)

## Cross-sharding

### Cross-sharding buffer

To ensure cross-sharding transactions, there is a special buffer in each node It is an array of elements with a height of 1000 blocks. To enable quick validation from this array, a hash is calculated (CrossHash), which is written to the header of the shard block. Array elements are hashes of successful cross-sharding transactions. Thus, the external shard can quickly check whether the cross-sharding transaction in another shard has been successfully completed. If a transaction appeared simultaneously in all required buffers, then it is recorded as successful.

Buffer of two shards, in bold text, matching transactions are highlighted:



Shard A

12017:1323DEF123449394324
12017:**A342423F12344432489**
12017:**ABCDEF185476345345**
12015:**2ABCDEF95348348344**
12015:**1AFDDE640963545417**
12015:**FACDEF893856769620**
12015:DDEF08937783444CC5
12015:**CCDEF1234434923449**
12014:**DFCD8237832F833331**
12013:**EE98237423982390619**
12013:**2383423534534347609**
12013:**2398445EF1234489543**
12013:**234E212F12313123255**
12013:**F32842F233483464544**
12013:**F89BA5454EF4353454**
12013:1A42002234493938DE
12012:**43433441242349DE38**
12012:**F2CDEF131233853DE3**
12012:**E45345EF45534545789**
12012:**A4F8334A5345436791**
12011:3B4DEF120328945812
12011:**1BC2EF123449393889**

Shard B

12017:**A342423F12344432489**
12017:**ABCDEF185476345345**
12015:**2ABCDEF95348348344**
12015:**1AFDDE640963545417**
12015:**FACDEF893856769620**
12015:**CCDEF1234434923449**
12014:**DFCD8237832F833331**
12013:**EE98237423982390619**
12013:**2383423534534347609**
12013:**2398445EF1234489543**
12013:**234E212F12313123255**
12013:**F32842F233483464544**
12013:**F89BA5454EF4353454**
12012:**43433441242349DE38**
12012:**F2CDEF131233853DE3**
12012:**E45345EF45534545789**
12012:**A4F8334A5345436791**
12011:8543985AFE453534532
12011:**1BC2EF123449393889**
12010:ED89534098356701001
12010:F19234AE765923904C
12010:C934ED0935456789A7

### Cross-sharding transactions

For transferring values between blockchains are transactions that are calls to functions of smart contracts with a sign of cross-sharding. Such a transaction calls the function in two shards at once.

To ensure that the call is atomic in two shards at once, there is a special protocol, and each function in smart contracts has an additional parameter - the Call flag, which is filled systemically depending on the situation (0 is the initial call, 1 is successfully completed in both shards, -1 - transaction canceled).

The procedure is as follows:

1. A cross-sharding transaction is formed and sent immediately to two shards.
2. In each shard, a function call is performed with the flag parameter = 0. In this mode, the smart contract must reserve transferred values.
3. If successful, the transaction hash is sent to the cross-sharding transaction buffer. The success of the execution is considered if the code did not raise an exception.
4. After 100 blocks in each shard, the hash is checked in both buffers (his and the correspondent), if the transaction is present there and then the function is called again with flag 1. This flag in the code should remove the reserve and transfer the value.
5. If the transaction hash is not in both buffers, then the function is called with the -1 flag, but only in the blockchain where the hash is present in the buffer. This flag should only accomplish the removal of the reserve (i.e., return to the blockchain to the original value that was before the transaction).

### Variants of creating cross-sharding connections

Due to the fact that the resources of one node are limited by the number of shards to exchange at the same time point, they are also limited to a fixed value.
How to define with which shards you need to build data exchange?
To do this, you can apply the following rules:

1. Dynamic binding. Noda monitors the composition of cross-sharding transactions, if there is a transaction with a shard with which there is no connection, then such a connection is created. When creating a connection, the priority of both the exchange and belonging to the trust cluster is taken into account.
2. The list of cross-shards depends on the deposit. The
3. list of cross-shards depends on the vote of the miners.

## Conclusions

It is possible to achieve infinite scalability by using the trust in the validation of the shard. But unlike other blockchains, such a trusting system has no problems with decentralization, since by default, it is disabled, and if the miner asked a trust group and was wrong at that, the network as a whole will not suffer because other miners have different trust groups.