«Системні технології» 6 (143) 2022 «System technologies»
DOI 10.34185/1562-9945-6-143-2022-02
UDC 004.42

P. Borysenko, V. Gromova

USING SHARDING TO IMPROVE BLOCKCHAIN NETWORK SCALABILITY

Annotation. Blockchain is a distributed and decentralized database for recording transactions. It is shared and maintained by network nodes, which ensures its operations using cryptography and consensus rules that allow all nodes to agree on a unique structure of the blockchain. However, modern blockchain solutions face network scalability issues due to different protocol design decisions.

In this paper, we discuss sharding as a possible solution to overcome the technical limitations of existing blockchain systems and different forms of its practical realization presented in recent research spurred by blockchain popularity.

Keywords: blockchain, sharding, scalability, distributed networks, proof of work, multi blockchain, decentralized systems, directed acyclic graph.

1. Introduction. Blockchain — a centerpiece technology underlying Bitcoin and over thousand other cryptocurrencies that rapidly emerged in recent years — offers great potential and benefits in many applications. Not only does it fuel modern economic system changes, but also decentralized markets and governance models that benefit from lack of censorship and reliability of not having a singular failure point.

The main drawback of such systems is their performance. Most popular existing solutions have incredibly low processing bandwidth and confirmation latency: around 15 transactions per second (tps) and 12 minutes per transaction for Ethereum, 7 tps, and up to 60 minutes for Bitcoin [2]. This is incomparable with processing rates of traditional payment systems like Visa or Mastercard, standing at around 5000 transactions per second on average [2, 3]. Unfortunately, more than available computation capacity will be needed to solve this problem.

In this paper we want to discuss a demand and ways to achieve a more scalable solution with a throughput of a few orders of magnitude higher than existing systems while maintaining crucial blockchain benefits like security and decentralization, by looking at existing approaches to the problem in the industry. These in-the-

wild solutions are often lacking proper scientific rigor, but provide interesting insight into the problem.

2. Problem definition. A few modifications to existing protocols were proposed in recent years to mitigate these issues among the industry and academic communities [4-6], but no significant advancements are discussed in academic literature.

As first described in [1], blockchain is a decentralized, replicated, and transparent data store that allows anyone to read data and verify its correctness. Data is usually stored on the blockchain in the form of blocks (hence the name) as a linked list with cryptographic hashes instead of pointers. This list is stored on a series of (usually independent and generally untrusted) nodes in a network that must communicate without the presence of a trusted intermediary maintaining the consistency of data stored on the chain. This setup presents several challenges to the classical Byzantine generals' problem of finding consensus among the nodes, such as a lack of viable identity verification infrastructure or quadratic complexity on the number of messages needed.

Classical implementations of the blockchain employ a Proof-of-Work consensus method of selecting a random node as a designated trusted verificator from a pool of available nodes by producing hard-to-solve but easy-to-validate computationally intensive problems (e.g., Hashcash-like [7] — finding a SHA256 hash that satisfies a specific condition — in Bitcoin protocol). The node that finds the solution to a problem is then allowed to modify the blockchain with the new data. However, it's easy to see how this is a very limiting solution that scales poorly with the number of nodes and transactions in the system — reducing the underlying properties of the required computational puzzle (i.e., block time interval) leads to increased security risks. So does also increase the total transactions allowed per block (while further decreasing network propagation). This makes proof-of-work consensus neither particularly cost-effective nor ecologically viable [8].

Many blockchain projects propose a move to use a different consensus mechanism, among others:

– Proof-of-Stake, where block proposers are selected by their stake in the system (wealth, age, or other factors);

– Distributed Proof-of-Stake, where the stakeholders vote on block proposers;

– Directed Acyclic Graph that removes sequential limitations on block processing.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

These methods come with their own limitations in other critical aspects of the blockchain infrastructure. (For a more detailed overview of the modern blockchain consensus protocols, see [12].)

This leads to the drive for another way of tackling the network throughput and bandwidth scaling problems present in the existing protocols, preferably without massive security and decentralization drawbacks.

Note that a few commercial blockchain solutions are promising to solve the scaling problem by relying on trusted infrastructure or using federated identities, i.e., not targeting an open permissionless environment [9-11]. This is a concise decision on their part, and we are not discussing them within this paper.

3. Sharding architectures. One of the possible solutions to the scaling challenges of blockchain comes from traditional database design.

Sharding is a technique in which the main blockchain is divided into smaller parts, called shards or sub-chains, that exist in either separate sub-network or on the main blockchain network and processes a non-overlapping set of transactions.

This produces several challenges that need to be solved:

1) Cross-shard transactions: Cross-shard transactions enable interoperability between accounts or transactions on different shards but are hard to do securely when a consensus of more than one sub-chain is required. This is an essential property, even more complex when smart contracts are involved, especially in call chaining scenarios (i.e., when a cross-shard smart contract calls another cross-shard smart contract).

2) Masternode assignment: Each sub-chain must start from a primary node, but this process should be sufficiently random to prevent adaptive attacks and masternode collisions, especially in a non-proof-or-work system.

3) Shard reshuffling and data availability: By design, blockchain networks are not stable, and as such, there is a massive potential for network reconfigurations that will lead to synchronization issues. This synchronization during masternode reassignment may take a significant portion of network uptime.

Let’s look at a few examples of sharding architectures and compare their approach to the challenges posed above. Based on the similarities we can then deduce possible solutions to these problems and how this knowledge can be applied to other network architectures.

3.1. Elastico. Elastico [14] is one of the first protocols that utilized sharding and was developed specifically to solve scalability issues, initially of Bitcoin, that can be applied to any blockchain. (Zilliqa [15] was one of the first cryptocurrencies that

used Elastico protocol as the basis for development from the start.) Elastico scales linearly with the amount of available computational power in the network and can withstand up to 25% of its nodes being byzantine adversaries.

The protocol is uniformly splitting the network into separate sub-networks, called committees, that run a disjoint set of transactions in parallel using a classical byzantine consensus protocol internally. One of the committees is selected to combine shards selected by other committees, computing and broadcasting the digest and selecting randomness seeds.

Inside one committee, each node is responsible for solving proof-of-work challenge and providing identity to other members. Once most members sign the selected shard, that is then sent to a final committee that produces the final value (usually equal to a traditional block). This process, called an epoch, is then repeated with different sets of transactions, with different final committees on each epoch.

Despite the initially proposed algorithm using proof-of-work as part of the process to establish the node identity, which is central to the byzantine consensus protocols, it's possible to use other mechanisms.

The algorithm assumes a few drawbacks:

1) The protocol assumes the transactions are independent and subsequently doesn't try to restrict the number of transactions per shard, as traditional Bitcoin-like protocols do. Consequently, no additional interdependent data can be stored on the chain, and double spending can become an issue.

2) Elastico requires careful consideration of the controlling constants — number of committees and transactions per epoch — that greatly influence its viability and scalability. However, protocol authors propose a way to estimate these values.

3) The protocol is reliant on randomness generation by the final committee of each epoch, which assumes an easy vector for Byzantine and Sybil attacks.

3.2. Aspen. Aspen [16] is a protocol that proposes service-oriented sharding, the separation of blockchain data based on service preference. This is rooted in the idea of a multi-blockchain, where multiple separate chains have a common genesis block and share checkpoint blocks that contain data for the management of the overall structure.

This technique requires additional restrictions on the sharded blocks, such that they can belong to only one service. The added benefit is that intra-shard protocols can be completely separate and independent of the main blockchain. Aspen contains methods for introducing these new sub-chains (and, most importantly, their protocols) on the fly as part of the maintenance transactions in the checkpoint blocks of

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

the main chain, allowing for seamless integration with existing systems. This, combined with the ability to be built upon any existing blockchain platform, makes service-oriented sharding a very interesting protocol for non-payment related blockchain solutions.

The obvious drawback here is a possible non-uniformity of the service-chains. While the number of side-chains does not hinder the scalability, the existence of high-load services that require much computational power can draw it from less transactions-heavy services, resulting in their throttling, unless an additional mechanism for controlling per-service throughput is introduced.

Unfortunately, no practical implementations of this protocol exist, and no more developments have emerged since its conception, as well as no peer-reviewed data on its properties of it.

3.3. TON Blockchain. TON or Telegram Open Network, Blockchain [17, 18] is a blockchain system developed as part of the Telegram messaging platform to handle a large number of transactions. Practical implementation is currently in the testing phase, but theoretical papers describing its architecture are openly available.

TON's direction is wildly different from previous examples. TON proposes a so-called Sharding Paradigm, implying that not only main blockchain can be sharded, but every resulting subchain can be subsequently subdivided further into sub-subchains practically ad infinitum.

As such, the central architecture of the TON blockchain consists of 3 chains:

1) mainchain, which possesses general information about subchains, validators, and stakes;

2) workchains, which are subchains of the mainchain, that contain actual transactions, smart contracts, and so on;

3) shardchains, which are subchains of workchains and correspond to a self-contained set of accounts.

Each mainchain and shardchain block is considered a 1-block blockchain on its own. This allows for sharding particular shardchains if they become too big, scaling the system on the go.

TON Blockchain uses a proof-of-stake solution, designating specific nodes as validators and subdividing them among shards in a pseudorandom fashion, changing every 1024 blocks. For each validator, there is a set order of preference for a concrete block to be chosen to be committed. Validators use a variant of Byzantine fault tolerant consensus for selecting the new block in a shard. The mainchain block is generated on a timeout or when all new shardchain blocks are processed, containing

hashes for all shardchain latest blocks, by consensus of all validators to achieve persistence in the system.

What is different from protocols like Elastico is that each validator can be assigned to more than one shard, processing them in parallel. Moreover, invalid blocks in the shardchains can be corrected by subchaining the wrong block and generating a correct transaction in the top block of the subchain. This means that mainchain forks can be achieved only by incorrect behavior of the majority of validators, meaning significant stake losses for them.

Another interesting development is in the cross-shard messaging between shardchains. Messages are allowed to travel only between ``neighboring'' shardchains, i.e., shardchains that have only a small difference in their shard identifiers. Routing messages further is a special transaction. This creates a routing network between shardchains with the approximate number of intermediate transitions being $\lceil \log_{16} N \rceil - 1$, which scales incredibly well on the number of shardchains N .

Shardchain validators are encouraged to process messages from the output queues of their “neighbors,” ensuring the routing mechanism works. The negative side of such a system lies in its heavy reliance on the proof-of-stake mechanism to ensure cross-shard transaction delivery and validation.

4. Discussion. Sharding solutions presented in this paper use a wide variety of different consensus strategies, with Elastico and Aspen building upon a more traditional proof-of-work Nakamoto consensus and TON using a proof-of-stake system at its core. All of them mix it with Byzantine-based protocols to achieve a consensus process to increase the system’s security properties.

It's worth noting that there exist consensus-independent sharding protocols, such as OptChain, as well as mixed systems, that use different consensus strategies inside and outside shards.

In contrast to the traditional single-chained solutions, sharding protocols are much more reliant on randomness generation for security purposes, complicated by the distributed nature of the system. Most of the discussed protocols use Verifiable Random Functions, Verifiable Secrets Sharing, and Verifiable Delay Functions, which are a de-facto standard in the community.

Cross-sharding is still one of the most complex problems to solve. For example, in [19], it's expected that more than 90% of transactions are cross-sharded if using the traditional Unspent Transaction-Output model, where each transaction represents a unique atomic state of the ledger. Because of these, cross-shard transactions

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

need to ensure consistency of the produced state in all of the shards involved. This is usually referred to as an atomic commit property.

There is no agreed-upon common protocol for atomicity. Solutions include:

– use of a distributed set of trusted authorities [14, 20];

– use of locking mechanisms [19];

– splitting the cross-sharded operation into a series of intra-shard transactions and special atomic-by-construction cross-shard transactions [17].

Often cross-shard communications are the primary source of bandwidth in sharded blockchain systems, so choosing the correct algorithm is paramount to ensure proper scalability. Cross-shard communication can also be a significant vector of attack on the blockchain, as many atomic commit protocols are vulnerable to Denial of Service attacks. For a more detailed analysis, see [13].

We argue that, at least to some degree, the security, atomicity, and cross-shard communication (or, similarly, bandwidth of such communication) are mutually exclusive properties for now. We can think of them as of CAP-like triad for sharding, presenting a unique challenge in balancing out these properties. We also argue that most practical applications of these systems, like those mentioned above, are only focusing on solving one of the issues and generally do not care for the overall system's performance.

Finally, there are other venues for research in sharding, such as horizontal sharding (the influence of which we can see in TON Blockchain) and heterogeneous sharding. In these instances, we see each new instance of the blockchain, an equivalent of a shard, being configured separately in response to a specific demand, which can be more flexible in solving specific growth issues and balancing network bandwidth. Heterogeneous systems can consolidate the solution for the atomicity and communication problems.

Security still poses a problem but can be addressed where needed, specifically by sacrificing the overall performance. One way of doing so would be by introducing special "secure nodes" that process sensitive transactions. They would be bottleneck points and influence the network's overall bandwidth. Horizontal sharding has the same properties with a different technological basis.

Beside the sharding protocols, there are other alternatives to tackle scalability issues, like [21], that are conceptually similar to sharding.

For the sake of completeness, we need to mention other solutions. Off-chain solutions are outsourcing the transactions to a third-party entity, providing only de-

scriptions to the blockchain. These solutions are especially prone to identity attacks as they require additional mechanisms for validating the off-chain transactions.

Another popular method of achieving scalability is using a Directed Acyclic Graph when blocks are not connected as a chain, but as a graph, with validity being represented by outgoing edges. Throughput can be optimized by not requiring downloading the complete graph for all nodes. The obvious drawback is the additional dependency on validators, as well as the probability of valid transactions being appended to an invalid sub-graph.

It is also worth mentioning that there, of course, exist many other valid issues that factor into the scalability problem, such as environmental factors and socio-cultural acceptance of blockchain solutions that we only briefly mentioned in this article. Such matters may further hinder the development of the mentioned approaches. Environmental concerns are the most pressing in this regard, as the performance impact on the energy footprint of some algorithms is very high and can be unattainable when applied at scale. Such factors warrant additional research.

5. Conclusion. Among the many proposed solutions to achieve scale-out throughput and latency limitations of permissionless blockchain networks, like off-chain solutions, Directed Acyclic Graph, and sharding, the latter is the only one that seems most suited to overcome primary security and centralization issues posed by the problem.

Sharding solutions still imply the use of consensus protocols on top of traditional proof-of-X consensus, dominated by Byzantine fault-tolerant solutions, as well as a cross-shard communication protocol that requires additional atomic properties.

New developments in recent years include hybrid and non-traditional solutions championed by multi-blockchains like TON Blockchain.

We analyzed several sharding algorithms that present the most interest in solving scalability problems of classical blockchain. We concluded that sharding is the best possible solution to the scaling issues in most existing classical blockchain architectures based on practical data from in-the-wild usage. [16, 17, 19, 21]

We conclude that multi-chain solutions, such as the one proposed by TON developers, are the most effective for solving sharding problems in a decentralized setting. The trusted authority distribution can be performed in many different ways, and this problem is very similar to the classic database scaling problems. Solutions proposed by Elastico and RSCoin both have their strong sides under different conditions.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

The results of this research can be applied to a number of problematic scaling solutions in existing blockchain networks, such as Bitcoin and Bitcoin Classic, as well as modern blockchain infrastructures such as Ethereum.

Additional research is required regarding the non-algorithmic properties of these solutions, such as their environmental impact.

Acknowledgment. P.B. wants to thank Volodymyr Pasko for insightful conversations on different aspects of blockchain technology during the preparation of this paper.

REFERENCES

1. S. Nakamoto, “Bitcoin: a peer-to-peer electronic cash system,” [Online], 2009. Available: https://bitcoin.org/bitcoin.pdf.

2. Bitcoin Wiki. (2019) Scalability. Scalability targets, [Online], Available: https://en.bitcoin.it/wiki/Scalability.

3. Visa. (2019) Benefits accepting Visa, [Online]. Available: https://usa.visa.com/content library/modal/bene ̇ts-accepting-visa.html.

4. I. Eyal, A. E. Gencer, E. G. Sirer, and R van Renesse, “Bitcoin-ng: A scalable blockchain protocol,” arXiv: 1510.02037 [cs.CR], November 2015.

5. G. Andersen. (2015) Bitcoin improvement proposal 101, [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0101.mediawiki.

6. J. Garzik. (2015) Bitcoin improvement proposal 102, [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0102.mediawiki.

7. A. Back, “Hashcash - A Denial of Service Counter-Measure,” [Online], August 2002. Available: http://hashcash.org/papers/hashcash.pdf.

8. Digiconomist. (2019) Bitcoin Energy Consumption Index, [Online]. Available: https://digiconomist.net/bitcoin-energy-consumption.

9. Hyperledger. (2016) Sawtooth distributed ledger, [Online]. Available: https://wiki.hyperledger.org/display/sawtooth.

10. D. Mazieres, “The Stellar consensus protocol: A federated model for Internet-level consensus,” [Online], 2016. Available: https://www.stellar.org/papers/stellar-consensus-protocol.pdf

11. B. Chase, E. MacBrough, “Analysis of the XRP ledger consensus protocol,” arXiv:1802:07242 [cs.DC], February 2018.

12. S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, “SoK: Consensus in the Age of Blockchains,” in Proc. of the 1st ACM Conference on Advances in Financial Technologies (AFT ’19). Association for Computing Machinery, New York, NY, USA, October 2019, pp. 183–198.

13. G. Wang, Z. Shi, M. Nixon, and S. Han, “SoK: Sharding on Blockchain,” in Proc. of the 1st ACM Conference on Advances in Financial Technologies (AFT ’19). Association for Computing Machinery, New York, NY, USA, October 2019, pp. 41–61.
14. L. Luu, V. Narayanan, K. Baweja, C. Zheng, S. Gilbert, and P. Saxenna, “A secure sharding protocol for open blockchains,” in Proc. of the 2016 ACM SIGSAC Conf. on Computer and Communications Security (CCS ’16, New York, NY, USA, 2016, pp. 17-30.
15. The Zilliqa Team, “The Zilliqa technical whitepaper,” [Online], 2017. Available: https://docs.zilliqa.com/whitepaper.pdf.
16. A. E. Gencer, R. van Renesse, E. G. Sirer, “Service-Oriented Sharding with Aspen,” arXiv:1611.06816 [cs.CR], November 2016.
17. N. Durov, “Telegram Open Network,” [Online], March 2019. Available: https://test.ton.org/ton.pdf.
18. N. Durov, “Telegram Open Network Blockchain,” [Online], October 2019. Available: https://test.ton.org/tblkch.pdf.
19. E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, “Omniledger: A secure, scale-out, decentralized ledger via sharding,” in 2018 IEEE Symposium on Security and Privacy (SP), 2018, pp. 583-598.
20. G. Danezis, S. Meiklejohn, “Centrally banked cryptocurrencies,” arXiv: 1505.06895 [cs.CR], December 2015.
21. J. Wang, H. Wang, “Monoxide: Scale out blockchains with asynchronous consensus zones,” in 16th USENIX Symposium on Networked Systems Design and Implementation (NSD ’19), 2019, pp. 95-112

Received 07.11.2022.
Accepted 11.11.2022.

Використання шардингу для покращення масштабованості мереж

Блокчейн – це розподілена та децентралізована база даних для зберігання інформації про транзакції. Вона підтримується членами мережі, що гарантують її операції використовуючи криптографічні засоби підтримки консенсусу, що дозволяють членам мережі домовитися щодо унікальної структури блокчейну. Однак, сучасні блокчейн рішення стикаються із проблемами масштабованості через різноманітні проблеми із їх протоколами. У цій статті ми розглядаємо використання шардингу як можливого рішення для подолання технічних обмежень існуючих блокчейн систем і різні форми його практичного втілення у сучасних дослідження, викликаних популярністю блокчейну.

ISSN 1562-9945 (Print)
ISSN 2707-7977 (Online)

Громова Вікторія Вікторівна - Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського".
Борисенко Павло Борисович - Національний технічний університет України "Київський політехнічний інститут імені Ігоря Сікорського".

Gromova Viktoria Viktorivna - assistant Professor, Igor Sikorsky Kyiv Polytechnic Institute, Faculty of Applied Mathematics, Kyiv, Ukraine.
Borysenko Pavlo Borysovych - assistant Professor, Igor Sikorsky Kyiv Polytechnic Institute, Faculty of Applied Mathematics, Kyiv, Ukraine