Background

Stateless validation refers to a design of a blockchain protocol where validators do not need to store the state locally in order to validate blocks. Originally proposed by [Vitalik Buterin](#) in 2017 as a possible direction for Ethereum and subsequently explored by other researchers including [Dankrad Feist](#), stateless validation is a hot research topic because it provides a major scalability and decentralization upgrade for protocols. As the growing state ("state bloat") of blockchains contains more data over time, there is a greater urgency to address the question of how to store and manage state in the medium to long term.

In short, stateless validation is an elegant design to address the issue of state bloat by putting the onus of storing the state on relatively few nodes in the network. Other validators can then cheaply execute blocks without having to store the ever-growing state by annotating a block with the state witness required to execute the block. More importantly, those expensive nodes which do store the full state cannot do anything malicious, as the validity of blocks rely on approvals from those stateless validators.

In the design outlined above, the state-storing nodes with higher hardware requirements still need to store the full state, which means that at some point, they will still face the conundrum of state size outgrowing their hardware capacity. Sharding, on the other hand, is another scaling technique that addresses the state growth issue by dividing the state into multiple shards so that no validator ever needs to store the full state. It turns out that if we combine stateless validation with sharding, we can design a highly scalable blockchain that is also very decentralized.

Design Overview

The overall design is as follows:

- There are two types of validators: Chunk Proposers

, who are responsible for packaging transactions and creating chunks (shard blocks), including state witness needed to execute chunks. Chunk proposers need to have the state of the shard they are responsible for stored locally. Stateless Validators

, who do not store the state of any shards locally and instead rely on state witness provided by chunk proposers to validate chunks.

- Stateless validators are assigned to each shard randomly on every block. The assignment is determined by an onchain randomness beacon. When they receive a chunk, alongside its state witness, they execute the chunk using the state witness to verify whether the new state root matches the one claimed by the chunk proposer who produces the chunk. If the new state roots match, they would send an endorsement of the chunk to the next block producer. A chunk can be included in a block only if it has more than 2/3 endorsements from validators responsible for that chunk.

Security Analysis

A question that naturally arises in sharded blockchains is the security of the system. In this stateless validation design, the security of each shard relies on the random assignment of validators to shards and the frequent rotation of validators. More specifically, let's assume that there are a total of n

validators (with equal stake) and at most 1/3 of them are malicious. There are s

shards and k

validators are assigned to each shard at every block, so $n = sk$

. Then the probability of a shard getting corrupted can be analyzed as follows (here p

denotes the probability of a randomly chosen validator being malicious):

- If we only sample validators for one shard, then the probability of sampling l

malicious validator is the hypergeometric distribution

$$P(X=l) = \frac{{k\choose l} {n-k\choose k-l}}{{n\choose k}}$$

Using chernoff bound we know that the probability of this sampling being bad, i.e, having more than 2/3 malicious validators is

$$P(X\ge 2k/3) \le e^{-D(p + 1/3||p)k} = e^{-\frac{k}{3}}$$

- For s

samplings, the probability of at least one getting corrupted is

$$P_{bad} \le \sum_{i=1}^{s} P(\textrm{one shard is corrupted}) \le se^{-\frac{n}{3s}}$$

When n

and k

are large, i.e, there are many validators assigned to the same shard, the last term $se^{-\frac{n}{3s}}$

can be made negligbly small.

Our numeric calculation (based on multivariate hypergeometric distribution) shows that with 800 validators and 4 shards, the probability of the networking getting corrupted is roughtly 10^{-29}

, which means that in expectation it takes 10^{29}

blocks for the system to fail and assuming 1 block each second, that translates to 3×10^{21}

years! So in practice the blockchain can be considered secure.

Benefits

There are quite a few benefits of the proposed stateless validation design for a sharded blockchain:

- It addresses the security issue that is a challenge in designing sharded blockchains.

- The separation of chunk proposers and stateless validators make the network more decentralized as most validators can operate on relatively cheap hardware.

- The sharding design makes it possible to limit the state size of each shard. As a result, chunk proposers can store the state of a shard entirely in memory. If each shard is limited to 50GB, a machine with 64GB of RAM would allow chunk proposers to hold the state in memory.

- The design is also future-proof. With zk becoming more and more mature, it is possible to imagine that proof generation will at some point be fast enough for zk to be integrated into this sharded blockchain protocol, where the stateless validators only need to validate one zero-knowledge proof, instead of executing chunks against state witness. This would further improve the scalability and decentralization of this sharding design.

Outro

NEAR Protocol is about to finish implementing stateless validation on its sharded network in the spring of 2024. The design and implementation of this approach may also be of interest to researchers of Ethereum and other blockchains. We would like to get feedback from the Ethereum research community on the design and tradeoffs. Here is a link to the full technical paper (revised from NEAR's original sharding design, Nightshade, first published in 2019).