By block level sharding I mean having a single chain with large blocks, which are broken up into shards. Validators would run a cluster of servers, and each server would be assigned a range of shards. Validators could scale without any practical limits using traditional distributed systems technologies, like sharded databases. If clusters grow very large and fault tolerance is desired, validators could use consensus-based storage systems with strong consistency, such as Spanner or CockroachDb; I elaborated a bit here.

This would lead to some centralization, but I'd like to explore those concerns a bit, since I don't find any of them very compelling. I wonder if there are stronger arguments that I'm missing.

Benefits

The main benefit is keeping the protocol simple. There's no need for locking/yanking, and no signature aggregation. It's trivial to find servers with data for a certain shard, since servers never switch shards. There's only one type of chain, rather than one main chain and many collation chains.

The lack of locking/yanking also keeps the EVM model simple, saving contract authors some pain. It also reduces the latency of regular payments, since there's no need to wait for lock/yank operations to finalize. For certain contracts requiring lots of cross-shard communication, block level sharding could give vastly better performance.

Centralization concerns

Obviously, block sharding would raise the barrier to running a validator node. One of the rationalizations for small block sizes, e.g. in the Lightning Network paper, is that it lets small players fully validate the blockchain. However, this becomes a non-issue with Casper FFG or any BFT consensus algorithm. Only validators need to download transaction data; light clients can download just the votes along with Merkle proofs that specific transactions are included. In either model, we assume that an invalid block will never get 2/3 votes.

There would be some increase in centralization, though a degree of centralization seems inevitable. In the long term, regardless of protocol design, the vast majority of coins will likely be held by specialized custodians, exchanges, banks and other large institutions. But for the sake of argument, let's suppose that block-level sharding would significantly increase centralization.

Centralization could make 34% attacks somewhat easier to coordinate. On the other hand, with chain-level sharding we would have SHARD_COUNT

random samples of the validator pool at each block height, rather than just one. That means more opportunities for an attacker with <34% stake to get lucky and get 34% representation for a single block height and shard ID. We can use the binomial CDF to calculate the probabilities.

Another risk of centralization is that it makes soft forks, such as censoring a certain account, easier to coordinate. With BFT consensus systems, however, we can design the protocol such that once honest validators see a valid block, they will always vote for it (or its descendants) unless they see a conflicting supermajority. This way, soft forks would require a sustained 67% supermajority. This would be even less of a concern with systems like Zcash, where censorship is impractical since account data is private. There could still be soft forks based on public fields like transaction fees, but that's less of a concern.