

I have a generic question and proposal about storage management and Eth. Forgive my ignorance on the topic and feel free to direct me to existing resources on this subject.

## Cloud Storage in an Eth-like Blockchain

Over the last decade or so, a large number of managed services for cloud storage have been created on top of the highly distributed, durable, and available block stores used by big tech companies. S3, Google Cloud store, Alibaba cloud, etc.

These services are neither censorship resistant nor trustworthy, but data structures such as Merkel trees can be stored and validated using only a minimal amount of locally stored trusted state. For example, an ethereum node could store block headers and execute transactions by fetching Merkel proofs from a distributed storage service.

If we imagine a world where all eth nodes use a service like this, then it also makes sense for the block storage to be bundled and shared between many nodes. Many nodes could share the same data with almost no sacrifice in durability. Data replication and durability is handled by the (untrusted) managed service while ethereum nodes independently verify Merkel proofs.

In such a system the data storage cost for each node would be the price of the managed service, which can be very cheap because of shared storage. Block producing nodes could be spun up and spun down with much lower friction.

## Objections

I see some obviously objections in terms of fragility. The managed storage service may increase concentration and reduce the system to a single point of failure. However, there's nothing to stop this from happening in Eth 1.x or even Eth 2.0 (correct me if I'm wrong here).

Second I can see perverse incentives for the storage providers. All of a sudden we would have actors financially motivated to increase the amount of data retrieved to validate each block. Like the last objective, I think there is also a problem with current protocols, if the cloud based node can exist.

Third, managed eth nodes already exist and this is very close to the same thing. I think there is a crucial difference though. Sharing untrusted state does not reduce decentralization nearly as much.

Lastly it's possible the cost of this is too high or performance (latency) too low. However I have a lot of experience estimating the cost of systems like this. My back of the envelope calculation is that storage services could provide eth validation data to nodes for less than \$450/month each, with no modifications or optimizations in geth. That's assuming 15M of gas per block, blocks consisting of only loads, each unjoinable (must be in a separate request). In reality each block will read similar data to other blocks, so heavy caching and batching by the service could reduce that cost by orders of magnitude.

Latency is also a solvable problem, many state reads can be speculatively issued in parallel, or the protocol could be amended to require transactions to declare what state they are going to read and write (this has other benefits for scaling because it makes it tractable to check commutativity relationships at transaction submission time without DOS).

I have been building services for the Chia blockchain that made me think some of this may be applicable to Ethereum.

## Thoughts?

I'm curious to learn and hear what others think. I unfortunately don't have much familiarity with the "state of the art" in ethereum research, so again feel free to link me to relevant discussion.