In another post it was highlighted that two ideas (namely multi-tries and partial statelessness) share similarities beyond both reducing witness sizes in the context of stateless clients. I will quickly recap the two ideas, and then show that multi-tries are actually better than partial statelessness in every respect I can think of.

- Multi-tries

: Instead of using a single state trie, use 2^n tries thereby partitioning the account space according to n-bit account prefixes.

- Partial statelessness

: Instead of storing just the state trie root, store the first n levels (aka the “top layer”).

Both ideas allow to reduce witnesses by n * 256 bits. However, below are reasons why multi-tries are better than partial statelessness:

1. Implementation complexity

: Multi-tries feel easier to implement than partial statelessness because a multi-trie is simply multiple instantiations of a plain trie, whereas partial statelessness requires distinguishing the top layer from the bottom layer and requires extra logic to handle the different cases.

1. Parallelism

: Multi-tries allow for highly parallel trie updates, whereas partial statelessness has a sequential bottleneck (the single trie root is a sequential bottleneck).

1. Storage

: Multi-tries require n * 256 bits of storage, whereas partial statelessness requires up to (2n - 1) * 256 bits of storage.

1. Future-proofness

: It is possible to implement partial statelessness optimisations on top of

multi-tries, but not the other way round.

TLDR

: Multi-tries FTW