Hi, I'm Ian, applied cryptography researcher and soon to be professor at the University of Maryland. I've been working on applied cryptography and cryptocurrency for a while (Zcash was my Ph.D. thesis), but am rather new to Ethereum and its low level details.

Ethereum writes data for one contract instance to many random locations in the Merkle Patricia Trie. By my understanding, this was intended to limit the damage an attacker can do by reducing everything to the average case. However as result, any update to the contract's data 1) causes many random writes to disk and far worse 2) forces a number of different hash chains to be recomputed for the Trie. The second one of these is a major cost. Optimistic and zk-roll up work around this by, in CS terms, batching writes to one location and letting people dispute it.

In general, when designing high performance systems, one of the major things is to limit random IO operations as this causes performance issues. This is common research problem in systems. Here of course there is also a security dimension in terms of resource exhaustion. The fact that, in blockchains, this is done in authenticated data structures makes the issue far worse as it costs computational resources to recompute the authentication (regardless of the on disk/in memory representation).

What are the plans to address this in Eth2? Is the plan to still write data to random locations even within one contract? It seems there are a number of ways to limit the damage an attacker can do that do not come with the IO and hash computation costs of doing random writes.