This post is a continuation of a series on stateless clients and log accumulators. For context see:

1. History, state, and asynchronous accumulators in the stateless model
2. A cryptoeconomic accumulator for state-minimised contracts
3. Batching and cyclic partitioning of logs
4. Double-batched Merkle log accumulator
5. Log shards and EMV abstraction

TLDR

: We detail a cryptoeconomic mechanism for EMV executions with sublinear use of the state trie. It is an alternative to cryptoeconomic accumulators with the benefit that users do not need to post collateral, and only have to push logs (the cheapest kind of onchain activity).

Construction

Given a “normal” stateful contract C

we construct a state-minimised equivalent contract C’

. The contract C’

has a “virtual state” maintained as follows:

- The contract stores a single confirmed virtual state root at a corresponding collation height.
- Users can push logs of the form [LOG T]

, called “virtual transactions”. (Log shards are an ideal substrate for such logs, providing cheap log ordering, friendly witnesses, and real-time data availability.)

- Virtual state transitions for C’

given a virtual transaction [LOG T]

happen like state transitions for C

given a transaction T

.

- Collaterised “executors” can suggest unconfirmed virtual state roots at more recent collation heights than the current confirmed virtual state.
- Whistleblowers can challenge unconfirmed virtual state roots and engage in a TrueBit-style protocol with executors.
- Whistleblowers earn a share of the collateral of adversarial executors.
- Non-adversarial executors advance the virtual state root and are rewarded with an internal fee system that mimics coinbase rewards and/or gas.

Conclusion

The construction takes the traditional notion of a transaction and decouples data availability (via logs on log shards) and validity (via TrueBit-style cryptoeconomic execution). The end result is a state-minimised execution protocol where the cost of validation is pushed away from (onchain) validators onto (offchain) executors.

Note also that transactions corresponding to virtual transactions can assume a stateful model for executors (as opposed to a stateless model), so virtual transactions do not

need to includes witnesses. In such a setup users get both short transactions (improving upon the standard stateless model) and cheap transactions with logs (improving upon standard execution).