Many thanks @Canhui.Chen for discussing the idea.

Goal

The goal of the proposal is to reduce the challenge-response interactive times of optimistic fraud-proof protocol to 1-3 times using

- a DA BLOB consists of 4096 challenge points (e.g., statehashes of intermediate states to be challenged)
- a zkVM verifier (e.g., zkWASM in GitHub - DelphinusLab/zkWasm) that can verifier a 100k or more sequence instructions on-chain

Background

Current interactive fraud-proof challenge protocol such as Optimism fault proof system uses

- binary search to narrow down the specific step/instruction of disagreement between the sequencer and the challenger (https://github.com/ethereum-optimism/optimism/blob/e7a25442ae03c2858076b9df6ea7f638287adb2e/specs/fault-proof.md)
- when the specific step of disagreement is found, a one-step on-chain executor to determine whether sequencer or challenger is correct (https://github.com/ethereum-optimism/optimism/blob/develop/packages/contracts-bedrock/contracts/cannon/MIPS.sol).

Considering about 40+B steps (instructions) per block transition (e.g.,https://github.com/ethereum-optimism/optimism/blob/e7a25442ae03c2858076b9df6ea7f638287adb2e/cannon/README.md), the protocol will take about 36 interactions,

which is costly in both time and gas.

Proposal

The proposal solves the problem by allowing a challenger to submit 4096 intermediate execution results (aka, statehashes)

in a single challenge transaction. The 4096 statehashes are not submitted by calldata nor stored on-chain - the statehashes are uploaded in a DA BLOB defined in EIP-4844, and only the datahash of the 4096-statehashes is stored on-chain. Since a BLOB has 128KB size, we can put 4096 statehashes in a single BLOB (proper hash-to-field-element mapping is required). With this, we would expect much lower gas cost (given that EIP-4844 and the following danksharding upgrade will significantly reduce the cost a DA BLOB vs calldata).

To answer the challenge, the sequencer will pick up one of 4096 statehashes, where the sequencer disagrees with the statehash but it agrees on the previous statehash. Therefore, a single interaction will reduce 4096x fold computational steps in challenge.

Further optimization is to employ a multi-step on-chain verifier to determine the winner. This can be done by a zkVM verifier when the computational steps (trace) between previous (agreed) statehash and current (disagreed) statehash is smaller enough for a zkVM prover to generate a proof of the multi-step execution.

Expected Results

Consider a +40B steps verification, using 4096 statehashes per interaction will be done in ~3 times

- an one-step VM verification. Suppose a zkVM can verify 4000+ steps on-chain, then the iterations will be reduced to ~2 times
- a multi-step zkVM verification. If the zkVM can verify ~10M steps, then only one challenge-response interaction + a multi-step zkVM verification is good enough.