Intro

We have a bunch of blockchains that are not able to talk to each other. So centralized solutions have appeared that let you transfer tokens or information from chain 1 to chain 2.

We can do atomic swaps between chain 1 and chain 2. But this does not allow you to make a token on chain 1 that represents an asset on chain 2. In order to do this we need a two way bridge. Chain 1 validates the consensus of Chain 2 and Chain 2 to consensus on Chain 1.

This has been done in the past for bitcoin. But in order to keep the chains synced on eth it cost about 3,000,000 gas. So this one way bridge has not been used a lot due to cost.

Here we propose a 2 way bridge where verification happens optimistically. The cost is ~5000 gas per block. The fraud proof costs 3M gas.

Optimistic verification

The prover commits to the full block header (~200 bytes) the nonce

and the result

that is the only on chain data.

We then have a fraud proof that given that block_hash

and nonce

that produces that result

. The prover provides a bond that is slashed if the fraud proof shows their commitment to be incorrect.

The cost here per block is (200 + 64)*16 = 4224

gas. So ethereum can validate another ethhash proof of work chain for a cost of 4224 gas per block.

Ethhash Fraud proof

The main loop of ethhash algorithm is defined https://eth.wiki/en/concepts/ethash/ethash.

Basically we read from a database 64 times hash the result together with some weird mixing fnv

function.

So our main cost is 64 reads from memory.

There are two approaches

1. Read from the dataset of about 2^32 entries 64 times.
2. Read from the cache of 2^25 items 64 times and 256 reads in order to construct the data item for the cache.

Seems like 1 is the better approach. So during our fraud proof we need to

1. Prove that a given location in the dataset contains a given input
2. Do the actual hash which is basically more fnv

calls on that data.

1. Do this 64 times.

Gas cost

Approach 1 costs ~1B gas. Due to the cost of calculating the cache on the fly dominates.

Approach 2 costs ~ 3.5M gas.

As there are no cache calculations just 64 reads merkle proof reads

reads:

128 * 32 * 32 * 16 = 2097152

flv:

64 * 64 * 345 = 1413120

So lets do approach 2 with a and fraud proof of ~3.5 M gas.

cache calculation

Traditionally the cache generation part was seen as a huge problem. But it turns out that the cache is totally deterministically generated which means we can precalculate the cache for future epochs as the cache gets updated every 30000 blocks about every 5 days.

So at deploy time we can create deploy a list of all checks for the next 10 years at a cost of 32*16 gas for every 5 days. Which is 37376 gas per year. ~300k for 10 years and 3M for 100 years worth of datasets seems to be a nice round number.

Note: this is contradicted by the yellow paper and the eth.wiki but i think both of these resources are mistaken. I confirmed this with core devs and made issue to fix https://github.com/ethereum/eth-wiki/issues/47 never the less this question seems to be the first things we should test experimentally as we try and build this.

Conclusion

Now that we can validate ethhash in the EVM. We can build a two way bridge from ethereum to other ethhash chains

We are also half way to bridging ethereum to all other turning complete chain with the only remaining requirement is to add a fraud proof to validate the proof of work from the target chain inside ethereum.

This should also be achievable using optimistic verification so we can do this at a lost cost.