Since every full node is required to verify that a smart contract was correctly executed, gas limits prevent centralization. An alternative is to allow offchain execution. Perhaps a hybrid approach that allows miners to calculate computationally heavy transactions on chain but without requiring replication by all nodes would simplify the scaling debate somewhat.

Proposal

A new keyword is introduced, lazy

, which allows only the miner to evaluate a function and publish the inputs and state changes, signalling to the network to accept the calculation without need for replication.

Implementation

1. When publishing a block that contains, a lazy function, a miner deposits ether into a mediation smart contract equal to twice the sum of the gas value of the entire block. This deposit transaction is included in the block before publishing.
2. The deposit is locked for 1 day, allowing anyone to challenge the state.
3. Any user reruns the block manually and notices the miner cheated by supplying incorrect outputs for the given inputs to a lazy function.
4. The flagging user submits a deposit equal to the miner deposit to flag the block.
5. The miner of the next block has to validate the transaction fully, spiking the gas limit for that block only. If the flagging is wrong, the flagging user's deposit is deleted. If the miner did cheat, the miner's deposit is sent to the flagging user.

Benefits

By allowing large lazy functions to mostly go uncontested (because of the economic penalties), the gas limit per block can be dramatically increased without affecting block size or network speed. Only in the case of dispute (which will hopefully be rare, given the costs), will the block under dispute be potentially far larger than previous blocks since the lazy function will now become an active function. DDOS attacks are too costly.

Drawbacks

In the case of a fraudulent block detected, after punishing the user, the nodes have 2 choices:

1. Punish the bad actor but accept the erroneous result because re-mining the blocks will punish future actors who acted on the erroneous block.
2. Hard fork at the erroneous block.

Neither of these is particularly desirable and if the results of the lazy function can lead to an outcome that benefits the miner more than the lost deposit, then the validity of the blockchain is always in doubt. In that case, it might serve to allow the contract creator to specify the size of the deposit. Miners can then collectively set a deposit requirement upper limit much like a gas limit but in this case, contract designers must be cognizant not to create lazy functions whose outcomes are more valuable than the required deposit. Instead the best case for lazy functions are low value, high computation calls that would otherwise be performed by trusted off-chain oracles.