Motivation

Gas estimation is a key piece of infrastructure that ensures transaction will have enough gas in the runtime. Currently, the community relies on the eth_estimateGas

RPC endpoint to determine the minimum amount of gas needed for transactions to succeed. However, this poses a challenge when computing gas for contracts that catch unsuccessful external calls

in their logic.

Current Solution

The eth_estimateGas

endpoint currently employs a binary search algorithm to find the optimal amount of gas required for transaction execution. It lowers the amount of gas if the simulation is successful and raises it if the simulation fails with an out of gas

error.

Proposal

To improve gas estimation, we propose tracking internal transaction failures caused by out of gas

errors and changing the condition for lowering the amount of gas. Instead of lowering the gas only if the simulation is successful, the gas will only be lowered if the simulation is successful and doesn't encounter out of gas

errors in any internal transaction.

In go-ethereum, this can be implemented by introducing a newlogger that only listens to CaputeFault

and records an error if it occurred due to insufficient gas.