

We analyse the time-taken to extract data from an Ethereum node. Block time of the Ethereum mainnet is ~15 seconds, if certain data extraction functions take more than 15 seconds to pull data then a block explorer using these functions will fall behind the ethereum chain.

We take one such heavy function, which is used commonly in block explorers to access the internal transactions, and demonstrate that if gas used increases to more than 12 million units, getting internal transactions will take greater than 15 seconds and the block explorers will start lagging behind the main chain. This means that we should constraint gas limit to 12M

, for a block explorer to easily follow the mainnet.

The following [notebook](#) can be used to rerun the analysis.

Key findings:

1. Even at the current 8M gas limit we sometimes overshoot the 15 second time limit.
2. The time taken to extract transaction traces depends on complexity of the contract codes and gas used in the block.
3. If gas used increases to more than 12M then we would consistently start hitting this time limit.
4. The above value of gas limit assumes that transactions in the block are not complex contract calls.
5. It is possible to find specific OPCODE's which might create complicated contracts making the extraction exceed the time-limit at lower gas limits.
6. One method to bypass this bottleneck is developing Ethereum nodes optimized in data-delivery.

Presentation:

How to rerun the notebook

1. Have an archive node synced upto 5M blocks.
2. Install python3
3. Clone the [repo](#)
4. Install dependencies from requirements.txt: \$ pip install -r requirements.txt
5. Open the notebook using: >>> jupyter notebook

Questions:

Tweet to @AnalyseEther