Calculating Costs in Ethereum Contracts

Danny Ryan Jun 28, 2017

Thinking, coding, and explaining Ethereum and blockchain technologies.

GAS PRICE PSA *(2017–08–23): The median gas price at the time of writing this article was 28 Gwei, and continues to be in the realm of 20 Gwei. This is far greater than the typical average and safe-low found on EthGasStation (4 and 0.5 Gwei respectively). The median is so high because of bad gas-price defaults found in many wallets. I* ***highly recommend*** *using EthGasStation's average gas-price or lower in order to not pay high fees and to help drive down the market rate for gas-price.*

UPDATE *(2017–09–6): I ported the Google Spreadsheet of OPCODES to a github repo. This repo will be maintained and updated as the yellow paper evolves.*

What are users storing when they hold Ether? In one sense, they are storing the ability to perform computation on the Ethereum network. This computation is done in a decentralized fashion:

> *A miner executes the computation associated with each transaction being included in a block, resulting in an updated state. Upon successfully mining a block, a miner broadcasts the block to the network. Each of the other miners and non-mining nodes verify the validity of the transactional computation and resulting state change before accepting the block as valid, incorporating the block into their copy of the blockchain, and moving on to the next block.*

You may have noticed that there is incredible amount of redundancy for every bit of computation on the network. Namely, each node verifies the results of each transaction—read: *every node runs all of the computation*.

I've been researching Ethereum and other blockchain application

platforms for a long time now, and rarely, if ever, do people outright say this. Once you get into the more technical side of things, it becomes an obvious feature of the system, but to the less initiated this is not so obvious. Wouldn't this be far more costly than just running the computation on a server? Yes, by the nature of the protocol, it has to be. The following is likely a fundamental principle of economics or computation or both:

> *More machines running code == More money spent on running code*

We must remember that executing code and updating state on Ethereum is far different than doing the same on a simple server. The resulting state changes in Ethereum have different properties than that of a server — namely: immutability and public verifiability. Let's first take a look at how much computation actually costs on the network and then we can try to decide if it's worth it.

How much does it cost?

Gas

Each low level operation available in the EVM (Ethereum Virtual Machine) is called an OPCODE. These include operations such as ADD — adding two integers together, BALANCE — getting the balance of an account, and CREATE — creating a new contract with supplied code. Each of these OPCODEs has a number called "gas" associated with it. *Gas* is an abstract number that represents the relative complexity of operations. For example, ADD uses 3 gas while MUL (multiply two integers) uses 5 gas, so MUL is more complex than ADD.

I've compiled the gas required for each OPCODE in the EVM (Ethereum Virtual Machine) here. This information is pulled from the most recent version of the Ethereum Yellow Paper (EIP-150 Revision)— the formal specification of the Ethereum Protocol.

It is important to note that all transactions cost 21000 gas as a base. So if

you are just transferring funds and not interacting with a contract, your transaction takes 21000 gas. If you are interacting with a contract, your transaction takes 21000 gas plus any gas associated with running the contract.

Gas Price

While gas is fixed per operation, the amount a user pays per gas—*gas price*—is dynamic and dictated by market conditions. Gas price is a value representing how much Ether the user is willing to pay per gas. When a user sends a transaction, they specify the gas price in Gwei/Gas (1 Gwei equals 0.000000001 ETH), and the total fee that they pay is equal to *gas_price * gas_used*. Miners are paid out this fee and so they prioritize transactions with a higher gas price. The higher gas price you are willing to pay, the faster your transaction will be processed.

ETH Gas Station is a great resource for understanding the current gas market conditions. "Recommended User Gas Prices" shows the range of gas prices you might pay and the expected transaction times.

No Really, how much does it cost?

Operations in Ethereum cost gas_price * gas_used, but what does this translate to in both ether and dollars? I've compiled a spreadsheet of some example operations and associated costs at the current median gas price (28 Gwei) and current USD/ETH exchange rate ($295/ETH). Each line shows a task, the gas required for the task, the cost in ETH and USD, the number of this task you can perform with 1 ETH and 1 USD, the number of tasks you can perform per block, and the number of blocks it takes to perform this task.

Adding Numbers

The following is a sample from the spreadsheet. This shows the costs associated with adding or subtracting two integers.

Task	Gas Required	Cost (ETH)	Cost (USD)	Ops per ETH	Ops per USD	Ops per Block	Blocks to complete Op
Add or subtract two integers	3	0.00000009	0.00002655	11111111.11	37664.78343	1566666.667	0.0000006382978723
Add or subtract two integers 1 million times	3000000	0.09	26.55	11.11111111	0.03766478343	1.566666667	0.6382978723

Cost of Adding Integers

We can see here that it costs 0.09 ETH or $26.55 to add two numbers together 1 million times. Compared to running on a local computer or cloud server this seems pretty high. Let's do a quick price comparison to AWS.

I can add two numbers together 1 million times in python in 0.04 seconds. Amazon charges $0.0059/hour for their cheapest EC2 instance—t2.nano. This costs $0.000001639/second or $0.000000066 for the operation. Compared to $26.55, this is about 400Million times more expensive (or 40Million if you are willing to pay a low gas price). Whoa!

Storing Data

Another common operation we might be interested in is storing data whether it be storing a single value such as the number of days until a contract expires or something a bit more ambitious such as the contents of a short story.

Task	Gas Required	Cost (ETH)	Cost (USD)	Ops per ETH	Ops per USD	Ops per Block	Blocks to complete Op
Save a 256-bit word to storage	20000	0.0006	0.177	1666.666667	5.649717514	235	0.004255319149
Save 1 MB to storage (31250 256-bit words)	625000000	18.75	5531.25	0.05333333333	0.000180790960	0.00752	132.9787234
Save 1 GB to storage (1000 MB)	625000000000	18750	5531250	0.000053333333	0.000000180790	0.00000752	132978.7234

Cost of Storing Data

We can see from the above snippet that storing data to the blockchain is extremely expensive, but for good reason! When you store data in the blockchain, you store data into an immutable database replicated across 10s of thousands of nodes. Doing operations like uploading your favorite movie to the blockchain is and should be entirely cost prohibitive to keep the growth of the blockchain manageable. This intuition is captured by the fact that storing a single 256-bit word requires 20000 gas, over 6000 times more expensive than adding two numbers together.

Another bottleneck of storing large amounts of data is the current Block Gas Limit of approximately 4700000 gas/block. At this cap of gas per

block, it would take over 132 blocks to write 1 MB of data to the blockchain, and that is assuming you can manage to hog all of the gas per block and that there are no other operations required!

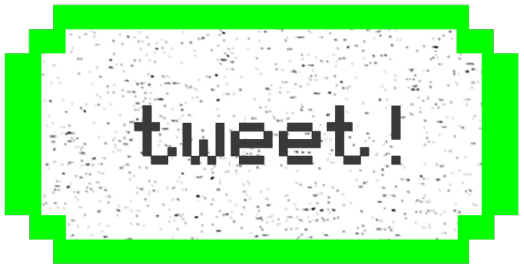
What does it mean?

To be fair, adding two numbers together 1 million times is a bit contrived. A well written contract would likely move such computational complexity off-chain and deal more with updating state in the contract. Storing vast amounts of data to the blockchain is also not an ordinary task. Depending on the task, a user would likely store a cryptographic reference (a hash) of the data on-chain and keep the rest of the data off-chain.

That said, we as developers need to be aware of these costs, and design dApps accordingly. We need to find the balance between on-chain and off-chain complexity, while still leveraging the decentralized capabilities of the blockchain.

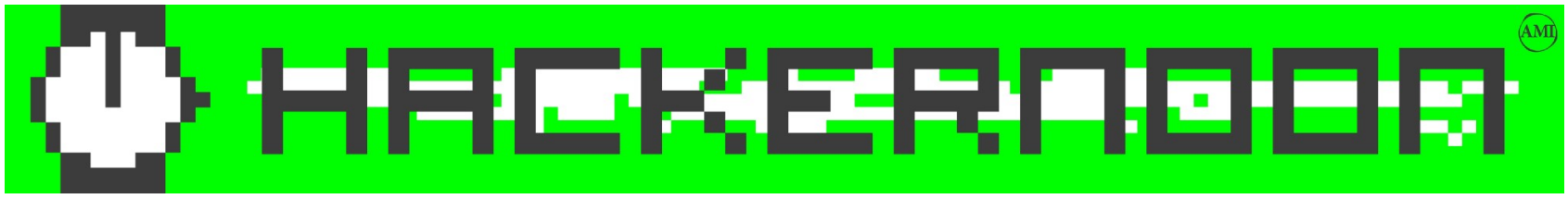
We also need to understand that at the end of the day we are comparing apples to oranges. With the increased cost and inefficiencies of the blockchain, we gain guarantees of open, censorship resistant code execution and publicly available, immutable data. We have never had such properties in computation before, and we do not yet fully understand the financial and societal gains that we might see.

Over the next year, I expect increased adoption and hopefully some long-awaited launches on the Ethereum blockchain. Only when we get significant activity on the blockchain will we truly be able to assess the costs and benefits.



> *Hacker Noon is how hackers start their afternoons. We're a part of the @AMI family. We are now accepting submissions and happy to discuss advertising & sponsorship opportunities.*

> *If you enjoyed this story, we recommend reading our latest tech stories and trending tech stories. Until next time, don't take the realities of the world for granted!*



Like what you read? Give Danny Ryan a round of applause.

From a quick cheer to a standing ovation, clap to show how much you

enjoyed this story.

- **Danny Ryan**

Thinking, coding, and explaining Ethereum and blockchain technologies.