Comparison between Ethereum gas and Matic Gas

I. Gas in Ethereum

Gas is Ethereum’s unit for measuring the computational and storage resources required to perform actions on the Ethereum blockchain. In contrast to Bitcoin, whose transaction fees only take into account the size of a transaction in kilobytes, Ethereum must account for every computational step performed by transactions and smart contract code execution. Each operation performed by a transaction or contract costs a fixed amount of gas. Some examples, from the Ethereum Yellow Paper:

- Adding two numbers costs 3 gas
- Calculating a Keccak-256 hash costs 30 gas + 6 gas for each 256 bits of data being hashed
- Sending a transaction costs 21,000 gas

Gas is a crucial component of Ethereum, and serves a dual role: as a buffer between the (volatile) price of Ethereum and the reward to miners for the work they do, and as a defense against denial-of-service attacks. To prevent accidental or malicious infinite loops or other computational wastage in the network, the initiator of each transaction is required to set a limit to the amount of computation they are willing to pay for. The gas system thereby disincentivizes attackers from sending ”spam” transactions, as they must pay proportionately for the computational, bandwidth, and storage resources that they consume.

Gas Accounting During Execution

A gas supply equal to the amount specified by the gas limit in the transaction is given when an EVM is needed to complete a transaction, in the first instance. Every opcode that is executed has a cost in gas, and so the EVM’s gas supply is reduced as the EVM steps through the program. Before each operation, the EVM checks that there is enough gas to pay for the operation’s execution. If there isn’t enough gas, execution is halted and the transaction is reverted.

If the EVM reaches the end of execution successfully, without running out of gas, the gas cost used is paid to the miner as a transaction fee, converted to ether based on the gas price specified in the transaction:

miner fee = gas cost * gas price

The gas remaining in the gas supply is refunded to the sender, again converted to ether based on the gas price specified in the transaction:

remaining gas = gas limit - gas cost

refunded ether = remaining gas * gas price

If the transaction “runs out of gas” during execution, the operation is immediately terminated, raising an “out of gas” exception.

Gas Accounting Considerations

The relative gas costs of the various operations that can be performed by the EVM have been carefully chosen to best protect the Ethereum blockchain from attack. You can see a detailed table of gas costs for different EVM opcodes in Table C-1. More computationally intensive operations cost more gas. For example, executing the SHA3 function is 10 times more expensive (30 gas) than the ADD operation (3 gas). More importantly, some operations, such as EXP, require an additional payment based on the size of the operand. There is also a gas cost to using EVM memory and for storing data in a contract's on-chain storage.

Gas Cost Versus Gas Price

While the gas cost is a measure of computation and storage used in the EVM, the gas itself also has a price measured in ether. When performing a transaction, the sender specifies the gas price they are willing to pay (in ether) for each unit of gas, allowing the market to decide the relationship between the price of ether and the cost of com- puting operations (as measured in gas): transaction fee = total gas used * gas price paid (in ether)

When constructing a new block, miners on the Ethereum network can choose among pending transactions by selecting those that offer to pay a higher gas price. Offering a higher gas price will therefore incentivize miners to include your transaction and get it confirmed faster.

In practice, the sender of a transaction will set a gas limit that is higher than or equal to the amount of gas expected to be used. If the gas limit is set higher than the amount of gas consumed, the sender will receive a refund of the excess amount, as miners are only compensated for the work they actually perform.

It is important to be clear about the distinction between the gas cost and the gas price.

To recap:

- Gas cost is the number of units of gas required to perform a particular operation.
- Gas price is the amount of ether you are willing to pay per unit of gas when you send your transaction to the Ethereum network.

Negative gas costs

Ethereum encourages the deletion of used storage variables and accounts by refund- ing some of the gas used during contract execution. There are two operations in the EVM with negative gas costs:

- Deleting a contract (SELFDESTRUCT) is worth a refund of 24,000 gas.
- Changing a storage address from a nonzero value to zero (SSTORE[x] = 0) is worth a refund of 15,000 gas. To avoid exploitation of the refund mechanism, the maximum refund for a transaction is set to half the total amount of gas used (rounded down).

Block Gas Limit

Block Gas Limit The block gas limit is the maximum amount of gas that may be consumed by all the transactions in a block, and constrains how many transactions can fit into a block. For example, let's say we have 5 transactions whose gas limits have been set to 30,000, 30,000, 40,000, 50,000, and 50,000. If the block gas limit is 180,000, then any four of those transactions can fit in a block, while the fifth will have to wait for a future block. As previously discussed, miners decide which transactions to include in a block.

II. Matic Network

Matic Network is a scaling solution for public blockchains. Based on an adapted implementation of Plasma framework (Plasma MoreVP).

The term "transaction" is used to refer to the signed data package that stores a message to be sent from an externally owned account to another account on the blockchain. Can be categorised into two main categories:

Value Transfer

- A value (in Ether) is transferred between two externally owned accounts

Execution of Contracts

- Here, a function/method call is made from an externally owned account to a contract account. These types of transactions can be further chained eg., the contract account can make several internal transactions before calling another contract account or returning back. Any type of transaction execution on Ethereum costs money - or gas.

"Gas" is the name for a special unit used in smart contract enabled blockchain networks. It measures how much "work" an action or set of actions takes to perform: for example, to calculate one Keccak256 cryptographic hash it will take 30 gas each time a hash is calculated, plus a cost of 6 more gas for every 256 bits of data being hashed. Every operation that can be performed by a transaction or contract on the platform costs a certain number of gas, with operations that require more computational resources costing more gas than operations that require few computational resources.