Worksheet 20 dec 2019

Calculating the Hashrate of Ethereum

Method

From Ethereum yellow-paper follows that the proof-of-work algorithm used for the consensus algorithm is specified as follows:

$$(n, m) = PoW(H, H_n, d)$$

Here H is the header of the block without the nonce and mix-hash, H_n is the header with included 64-bit nonce and d is the DAG used. n is a 256-bit hash value resulting from the ethash algorithm and m is the 256-bit mix-hash. The mix-hash originates from the memory-hard loop in the ethash algorithm. This mix-hash is computed as a means of proof-of-work instead of finding a correct SHA256 hash value, like in Bitcoin. The mix-hash is computing using memory-hard highly interdependent operations to achieve ASIC-resistance although numerous ASIC hardware has been developed last few years specifically for the ethash algorithm.

The following conditions much satisfy to mine a block.

$$n \leq \frac{2^{256}}{H_d} \land m = H_m$$

The the above equation, the hash value n must be lower than a target value $\frac{2^{256}}{H_d}$ specified by the difficulty H_d at that time. The probability of finding a correct nonce is

$$p_{targetFound} = \frac{\frac{2^{256}}{H_d}}{2^{256}} = \frac{1}{H_d}$$

Each nonce value behaves like an independent trial such that the number of tries is geometrically distributed. Which means that the number of tries to find a block with given difficulty can be computed as

$$N = \frac{1}{p_{targetFound}} = p_{targetFound}$$

The time to mine a block is kept relatively stable by adjusting the difficulty according to previously mined block's timestamp and difficulty.

`Note that computing the difficulty at a given time is not trivial as the difficulty is adjusted every Block instead of every 2016 blocks in Bitcoin`

The time to mine a block can be queried using various API's from mining pools and is kept track of by Ethereum as a timestamp t at the inception of a new block.

Since the time consumed to mine a block $E[i]$ is simply computed as the difference in timestamps between two existing blocks $E[i] = t_{B_i} - t_{b_{i-1}}$ and it's difficulty can be easily queried, the hashrate R can be easily computed. The hashrate R denotes the amount of PoW (ethash) operations performed by the network per second.

$$R = \frac{N}{E[t]} = \frac{\frac{1}{(\frac{(\frac{2^{256}}{H_d})}{2^{256}})}}{E[t]} \frac{H_d}{E[i]} = \text{hashrate H S}^{-1}$$

Calculating the Energy usage

To calculate the energy usage of Ethereum we construct a **lower bound**, **upper bound** and **best guess**. As most blockchains, Ethereum is a public blockchain which means that blocks containing transactions are publicly available. Useful information about blocks is available through API's.

Smart Contracts

Ethereum, like most cryptocurrencies supports transactions between addresses. In addition to that, Ethereum acts more or less like a distributed code execution engine. It features the EVM (Ethereum Virtual Machine) which acts as an environment for so-called 'smart contracts' te be executed which can perform transactions or interact with other contracts on the blockchain. The execution of the code is then validated by the miners as-well. It's very hard to estimate the amount of energy that is consumed by the EVM so this share is disregarded at first.

Proof-of-Work

The Proof-of-work algorithm represents presumably the majority of the energy usage of Ethereum as it is deliberately computationally hard. The proof-of-work algorithm in Ethereum is called Ethash and is designed to be ASIC-resistant. This means that (originally) the most common hardware to mine Ethereum were GPU's. Since the last few years, ASIC developments have made progress on Ethash and the efficiency has gone up tremendously. As feared, this will cause decentralisation eventually.

Proof-of-Stake

As a solution, The Ethereum Foundation is moving towards a Proof-of-Stake consensus algorithm which means that not computational effort is used as a mean of trust, but a 'stake'. This stake represents an amount of ETH, the share of ETH you have in the Network determines how much blocks you can mine. The Ethereum network is likely to consume negligible amounts of energy compared to now when the switch to PoS happens.

Data collection

To calculate the **upper bound**, **lower bound** and **best guess**, publicly available data from various API's is scraped and aggregated using scripts in *calc.py*.

Block data

Source: https://etherscan.io/apis#proxy
Starting from block 9123600 (12/18/2019), data from 500 blocks (approx. 2 hours) going back in blocknumbers with an interval of 40000 blocks (approx. 7 days) will be aggregated to gather:

- $avg(H_d)$ (difficulty
- $avg(E[i])$ (block time)
- $avg(R)$ (hash rate)
- UncleRate

- $gasUsed$

Ethereum Price

Source: https://etherscan.io/chart/etherprice
For each data point the value of 1 ETH in USD $ETHPrice$ will be added to the data set.

Hardware data

Source: Krause & Tolalymat (2018)
The set of Ethereum mining hardware from Krause & Tolalymat (2018) will be used and supplemented with later developments in Ethereum mining hardware since the dataset contains data until Q4 2017.

Lower bound

To calculate the lower bound of the energy usage of Ethereum, the most common (accepted) method, also used by Stoll etc is to assume the network hardware is made up of the most state-of-the-art efficient hardware currently available (CBECI, Bevand, Digiconomist).

When the hashrate R $(\mathrm{MH\ S^{-1}})$ of mining hardware is know together with its power usage P (W), the power efficiency PE can be calculated as $PE = \frac{P}{R}$ $(\mathrm{J\ MH^{-1}})$

Ethereum was designed to counter ASIC hardware by using a dagger-hashimoto-like algorithm wich is memory-hard. Therefore the most efficient hardware remained GPU's for a long time. Recent developments of ASIC for the ethash algorithm changed this. The table below is made up from mostly ASIC machines as well as some popular/most-efficient GPU's.

Wednesday 4 dec: (outdated)

Mining Hardware	Hashrate $(\mathrm{MH\ S^{-1}})$ (advertised)	Power Consumption (W)	Efficiency ($\mathrm{J\ MH^{-1}}$)	Release data
Antminer E3	190	760	4	Jul 2018
Innosilicon A10(1)	432	740	1.713	Sep 2018
Innosilicon A10(2)	485	850	1.753	Sep 2018
Innosilicon A10(3)	500	750	1.500	Sep 2019
Canaan xxxxxxx	2200	1500	0.68	TBA
Radeon RX470	24	150	6.250	Aug 2016
AMD Vega 64	31	295	9.516	Aug 2017
Titan V	77	237	3.078	Jul 2018
Tesla V100 16GB HBM2	90	250	2.778	Jun 2017
GTX 1080 ti	50	250	5	Mar 2017

sources:
- https://www.asicminervalue.com/

- https://www.reddit.com/r/ethereum/comments/d9rh4r/canaans*new*asic*is*a*pipe*dream*not*an_ethereum/

From this table it becomes clear that ASIC developments have dramatically improved GPU mining efficiency.

Results (outdated)

Using data from publicly available miningpool API's (Ethermine, Nanopool), the difficulty and blocktime (time to mine a block) are known. This way we can calculate the needed hashrate R needed **for one block**. The results are in line with commonly cited sources like https://etherscan.io/chart/hashrate.
Of course, this is the hashing power needed to complete **one previously mined block**, assuming geometric distribution of probability of mining a block. So it might be better to take the average of blocks mined during one day (or other time period).
Knowing this hashing rate R together with the most efficient available hardware efficiency from the above table, the theoretical power usage at a given moment can be calculated.
Queried on *5 dec*, using a PE value of 1.713 (Innosilicon A10) the power usage is around $300\ \mathrm{MW}$ or $2471816.557939\ \mathrm{MWh\ Year}^{-1}$ which is roughly $2,5\ \mathrm{TWh\ Year}^{-1}$
Using the most efficient GPU (Titan V) this number is $530\ \mathrm{MW}$ and $4.4\ \mathrm{TWh\ Year}^{-1}$

A new, more accurate lower bound can be calculated when enough data points have been gathered by the new API-scraping script. The above calculation only take into account data from a single block. Single block can vary alot and since a block is mined approx. every 15 seconds, this data is not very representative.
The new data points take into account 500 blocks (2 hours) and not only blocks mined by a singly mining pool but from **all** miners in Ethereum.

Using data this new method and from 500 blocks mined on (17/12/2019), the lower bound would be 210 MW (using the efficience of the Innosilicon A10 (500 MH)).

Upper Bound

In recent literature about the power usage of Bitcoin,some methods of calculating the upper bound are proposed.

- Method of Bevand(2017):
  - See what mining hardware is just/barely profitable for what time period. Then, assume 100% of the mining power in that period originates from such hardware and corresponding efficiency.
  - Assume that the hardware is never upgraded to more efficient hardware.
  - Uses known numbers of sold hardware to estimate more precisely
- Alex de Vries (Digiconomist)(2017):
  - Assume 100% of revenue of mined ETH is equal to the expenditures on electricity.
- CBECI
  - Uses Bevand method of selecting hardware.
  - Assumes a PUE of 1.20

Bottum-up, Bevand/cambridge method

Like Bitcoin, the PoW algorithm in Ethereum is (mostly) the source of the Energy consumption. The abovementioned methods for calculating the upper bound all use a profitability threshold to see what efficiency is needed at what time to mine profitably. We can construct such a profitability threshold for Ethereum.

Profitability threshold

The profitability threshold is defined as the break-even efficiency to which mining is still profitable at a given moment because of changing difficulty, global hashrate, ETH/USD price.

For a specific time period we can calculate this efficiency as follows:

We need to know the value of ETH mined (in USD), the average hashrate, average blocktime, the average electricity price.

The Break-even efficiency is calculated by assuming all generated ETH in dollars is equal to the electricity expenditures.
For a given time period t this gives:

$$\text{ETH revenue in USD}(t) = \text{NrofBlocks}(t) \cdot \text{BlockReward}(t) + (\text{UncleRate} \cdot \text{NrOfBlocks}(t) \cdot \text{uncleReward}) \cdot ETHPrice(t$$

$$\text{Elec. costs in USD}(t) = \frac{t \cdot \text{EthereumHashRate}(t) \cdot \text{HardwareEfficiency}(t)}{3600000} \cdot \text{KWhPrice}$$

To calculate the break-even hardware efficiency we assume the electricity costs are equal to the revenue from mining: $\text{ETH revenue in USD}(t) = \text{Elec. costs in USD}(t)$. We can thus calculate the $\text{BreakEvenEfficiency}$ as follows:

$$\text{BreakEvenEfficiency}(t) = \frac{\frac{\text{ETH revenue in USD}(t)}{\text{KWhPrice}} \cdot 3600000}{t \cdot \text{EthereumHashRate}(t)}$$

Results

Using the naive method of Bevand and the current profitability threshold (17/12/2019), the upper bound would be 910 MW.

Hardware set (work in progress)

Using the above profitability threshold for historical time periods of Ethereum, we calculate the upper bound by selecting the least efficient but still profitable hardware of that time period and assume that hardware is never upgraded until hardware is no longer profitable and upgrades must be made to the next least efficient but profitable hardware.
We thus need a set of mining hardware with corresponding efficiencies for the lifetime of Ethereum. For a given moment t we specify a function $\text{MiningHardware}(t, \text{HardwareEfficiency})$ which returns the efficiency of known hardware which meets the above aforementioned requirements. The dataset from Krause & Tolalymat(2018) contains efficiencies of common hardware used to mine Ethereum for the years 2012-2017. We extend this dataset with hardware from 2018-present to include the very important ASIC-developments of the last years to construct $\text{MiningHardware}(t, \text{HardwareEfficiency})$

Best guess

Ideas

Using data from Krause & Tolylamat (2018) with given PE values.

Bevand's approach of hardware-mix (adoption time, economics)

Using the Api from large miningpools

Sparkpool, nanopool and ethermine make up around 60% of total hashing power, source: https://www.poolwatch.io/coin/ethereum.

Miner data such as workers (hardware devices actually doing the mining) and corresponding hashrates can be an indicator of which class of hardware is used (GPU or ASIC) and maybe say something about the mining efficiency. At least it would give a good idea about how much is being paid for electricity or the likely PUE. In Stoll (2019), the miners in a pool are grouped into small, medium and large scale miners, with corresponding PUE values to give a more accurate power consumption.

Mining pools

Getting a reward for mining a block is rare when mining using a rig which usually represents a fraction of the total hashrate of the network. Mining pools bundle the hashing power of rigs across the world to win the race of finding a block and getting the reward. The mining pool acts as one Ethereum address/miner for the Ethereum network. The chances of a mining pool finding the correct nonce of a block is thus much greater than mining on your own. The rewards of a mined block is then distributed amongst participants in the mining pool, proportional to their computing effort. Mining pools dominate the Ethereum landscape as rewards are somewhat guaranteed as opposed to mining on your own. The largest share of Ethereum hashing power comes from only a handful mining pools [1]. These are sparkpool (32%), ethermine (22%), f2pool (11 %), nanopool (9%).
Ethermine and nanopool provide useful API's from which we can deduce alot of information about the miners. A miner can allocate his mining rig(s) effort to the pool and use the information from these API's to monitor his equipment. Equipment like ASICs or general purpose computers, outfitted with (alot of) GPU's are assigned to 'workers' by the mining software such that they can be monitored independently.

The API's of ethermine and nanopool allow anyone, who knows the miner address to monitor these devices. We can gather these mining addresses by seeing what transactions the mining pool address makes. Since the transaction history of Ethereum is public, the outgoing transactions of the mining pool's address must be the payout transactions to its miners (in most cases they are). Using the Etherscan.io API, two datasets have been collected containing the last 100,000 outgoing transactions that the ethermine and nanopool pool-addresses made. Using these addresses and the monitoring API's of ethermine and nanopool we can collect the hashing rate of workers of miners of both pools. Most workers are even conveniently named after the hardware it represents. It must be noted that it is hard to distinguish an ASIC compared to a GPU-cluster when not such convenient names are **not** used.

The result of this data collection will hopefully give a good estimate of mining hardware used and thus a realistic PE value of hardware used to mine Ethereum. This outcome will be somewhat representative as ethermine and nanopool together make up around 30% of all hashing rate of the Ethereum network.

Observations (work in progress)

The distribution of large, medium and small scale **miners** still needs to be analysed, but:

Average hashrates for **workers** in the above-mentioned mining pools are:
Sparkpool: 270 MH/s
Nanopool: 180 MH/s
Ethermine: 170 Mh/s

With the Api's I can estimate what the distribution of GPU vs ASIC is and the scale of individual **miners**.

Verifier Energy usage

A remaining question is whether the computational power of executing smart contract (EVM) code by verifier nodes is significant compared to the effort mining. This is where Ethereum differs drom Bitcoin as implements a Ethereum Virtual Environment (EVM) which is a distributed effort comprised of verifier nodes executing invoked contracts.

Gas

Ethereum has a system of **gas** which resembles the computational power needed for the execution/interaction with smart contracts. An amount of gas is incorporated in transactions to reserve an amount of computational effort. The gas (which is translated into ETH) is the fee a verifier node receives after executing the EVM bytecode. Gas should closely resemble the computational power needed for *opcodes* specified by EVM. Different arithmetic operations are assigned different values of gas.
The gas scores assigned to opcodes are not entirely accurate and actual power consumption of executing opcodes are highly dependent on hardware used and implementation of client software.
The main motivation of implementing gas was the prevention of DoS-attacks by stopping expensive smart contracts which could loop forever, not to give a fair fee for the computational effort needed for executing smart contracts.

A few studies have been conducted on the energy usage of opcodes [2] [3] [4] [5]

Currently working on:

- Waiting to gather enough data, this takes a long time since the limited API throughput
  - This data is used to make a better estimate of the **lower bound** and will be used to compute the **upper bound** and **best guess** numbers.
- Constructing the hardware set (almost done)
  - Can be found in working directory/DATA/GPUDATA.xlsx
- Methods for calculating the Best-guess
- Assumptions in Ethereum which might be different from Bitcoin (energy price, realistic profitability threshold),
- Energy usage of EVM
  - Using GasUsed to make estimates?

---

[1] https://miningpoolstats.stream/ethereum ↩

[2] Sankaran, S., Sanju, S., & Achuthan, K. (2018, July). Towards realistic energy profiling of blockchains for securing internet of things. In 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS) (pp. 1454-1459). IEEE. ↩

[3] Loghin, D., Chen, G., Dinh, T. T. A., Ooi, B. C., & Teo, Y. M. (2019). Blockchain Goes Green? An Analysis of Blockchain on Low-Power Nodes. arXiv preprint arXiv:1905.06520. ↩

[4] Aldweesh, A., Alharby, M., & van Moorsel, A. (2018, October). Performance benchmarking for Ethereum opcodes. In 2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA) (pp. 1-2). IEEE. ↩

[5] Aldweesh, A., Alharby, M., Solaiman, E., & van Moorsel, A. (2018, September). Performance benchmarking of smart contracts to assess miner incentives in Ethereum. In 2018 14th European Dependable Computing Conference (EDCC) (pp. 144-149). IEEE. ↩