The problem with Proof of Work and how we can solve it

Christian Felde - nupow.fi

Proof of Work is great, but has some major flaws.

By its very design it's meant to be inefficient, to force computers to slow down. Only by slowing the miners down can a decentralized system manage to come to an agreement, otherwise they would be flooded with block proposals. As more computers join, the system slows them all down more, by increasing the mining difficulty.

But this inefficiency is also its greatest problem. It leads to high energy usage, which isn't good for the planet or for power consumers, as the source is often dirty, and we could surely use this energy for something better? [1][2]

Having said that, Proof of Work is simple, easy to understand, and easy to implement. While we today have more energy efficient methods, like Proof of Stake, these bring with them other drawbacks.

Besides the added technical complexity, Proof of Stake also limits who can participate by requiring an upfront capital investment. It's no longer enough to boot up a miner and let it run, you need to boot up a validator, add the required capital, and ensure you manage the technical complexity that comes with this. If your Proof of Work miner stops running, you risk not making money. If your Proof of Stake validator stops running, not only do you risk not making money, you also risk your capital.

Introducing NuPoW

NuPoW, or New Proof of Work, is the next generation Proof of Work algorithm. Realizing that we can build on top of existing blockchain systems, we can change the incentive mechanism while keeping the algorithm simple.

With NuPoW, miners participate to find the secret value, just like with traditional Proof of Work. But, if too many miners are involved, and hence, the energy use becomes too large, the algorithm will automatically lower the rewards.

As the reward is lowered, less tokens are minted, and it becomes economically unprofitable to run a large mining rig. Thereby, we are incentivizing mining on smaller devices, like a cheap PC or a small Raspberry Pi computer. The algorithm automatically maintains the required balance between rewarding miners and keeping total resource usage under control.

Doing this allows everyone to participate, without the need for big investments in mining hardware. This again allows NuPoW tokens to be mined in a more decentralized manner, as the typical economies of scale that apply to traditional Proof of Work mining no longer apply to NuPoW mining.

We could maybe label this approach Proof of Indifference, as the average miner has no compulsion to be overly invested in mining hardware. It's important to understand that there's no clear link between the efforts needed to mine a token or coin, and their underlying value. The value of a token is driven by supply and demand. However, if anything, it could be argued that NuPoW tokens are harder to mine than traditional Proof of Work, because we can't simply spend more money on hardware to obtain more of them. They are harder to mine without this implying more energy usage. Spend too much energy mining a NuPoW token, and you'll end up with less minted tokens than if you didn't go past the indifference threshold. And the indifference threshold is a collective threshold, keeping the whole mining community in check.

The NuPoW mechanism

At the core of Proof of Work is a mechanism that looks to find a hash value, with this value being smaller than a given limit. This is how PoW works in for example Bitcoin [3] and Ethereum [4] (before switching to PoS).

The hash value depends on a cryptographic hash function, where its output hash is predictable for a given input value only. This property means the only way to find a desired hash value is through brute force, trying many inputs, until you stumble upon a desired output.

NuPoW uses the same mechanism. And much like with PoW, it is using this mechanism for two comparable reasons: Firstly, we must recognize that blockchains are shared utilities with limited resources, and we should avoid spam transactions. Secondly, we want to randomize, in a decentralized manner, who gets to mint the next NuPoW token.

In traditional PoW, the difficulty is a function of overall mining power. While is it practically impossible to directly measure the total mining power, we can indirectly measure it by setting a target for how often a new block is produced. If valid blocks are proposed too frequently on average over a given time period, the protocol collectively increases the difficulty by reducing the hash limit, and vice versa.

Because of this, there is no upper bound on how much energy we're willing to spend on PoW, and as long as the rewards justify the expense, this will unfortunately keep on increasing.

In NuPoW, the target is not to mint tokens at a given frequency, but to mint tokens at a given energy consumption. If the number of tokens minted over a given time period is more than a given threshold, the amount minted is reduced going forward, and vice versa.

Because we can't directly measure the value attributed to NuPoW tokens, we indirectly measure this value through a chain length target. At each step of the chain, it becomes progressively more difficult to find the correct input, as the last hash represents the next limit.

While the chain will eventually stall due to difficulty, throwing more energy at the problem will help progress the chain. Due to short term greed, some will be motivated to continue mining the chain past its target length. While this benefits them in the short run, it will in the

long run reduce the number of minted tokens, making their energy intensive setup unprofitable.

We can through the chain length target mechanism observe a self-regulating collective enforcement of total energy usage. Whatever the unobservable value attributed to a NuPoW token is, the reward adjustment will eliminate overuse of mining power.

Limiting wasteful energy

Tokens and coins have gained much interest and momentum since Bitcoin was first introduced in 2008. This helps secure PoW chains, as attackers must outspend the majority. This is wasteful, and Proof of Stake provides a more efficient method for driving the chain forward. [5]

The problem with both PoW and PoS is that they, through different means, allow for an ever-increasing concentration of wealth. PoW does this through economies of scale around mining operations. PoS does this by rewarding those with the most capital with new tokens more frequently than those with less.

NuPoW builds on the security model of existing chains by running on top of them, and this removes the security requirement we find in PoW and PoS chains. Limiting the reward like done in NuPoW would be a security concern had NuPoW been used directly to drive the chain forward. But as said, that's not the purpose of NuPoW.

Instead, the purpose is to enable a decentralized and distributed allocation of new tokens, through a process that anyone can participate in. Removed is the need for energy hungry specialized hardware, and large upfront capital investments.

NuPoW can be tuned with 3 different parameters: Chain length target, a duration after which the chain is assumed stalled, and a max token minting quantity.

The chain length target is the one we need to decide on, and it depends on the stalled chain duration and, as an upper limit, how much hardware we're willing to throw at the mining efforts. The token minting quantity then adjusts accordingly based on chain length activity, starting at the initial max token minting quantity.

With this, assuming actors want to maximize minting payouts in the long run, we can formulate a total token payout per chain run as

new tokens = max token minting quantity * chain length target

If the actors don't want to maximize minting payouts, as a collective, and instead are motivated by short term greed, or other factors, they would drive the expected chain length beyond the chain length target, and hence reduce the token minting quantity. If this continues for too long, the minting quantity would reduce to zero, before increasing again if the expected chain length is reduced to be below the target length.

Some could argue that a slight reduction in the token minting quantity could be justified by a longer chain, but this runs into two challenges. The first is that subsequent overrunning chain targets would continue to reduce the token minting quantity. The second is that NuPoW halves the reward per chain target overrun, requiring a doubling in the next chain length, which is challenging to achieve without also dramatically increasing the computing power from one round to the next. The process is symmetric, with the minting quantity doubling if the chain length is below the target, up till the max token minting quantity.

With that in mind, we can then start formulating a total token payout over time, spanning multiple chain runs. A chain run is one where miners have produced a number of valid input values, each input value reducing the previous hash limit, till we get to a point where the chain is considered stalled. It is considered stalled if no new valid input is produced before the end of the stall duration.

For the official NuPoW tokens [6], the max token minting quantity is set to 0x1000000000000000 (1.152921504606846976) and the stalled duration set to 30 minutes. Assuming a block time of 12 seconds, as only one minting event can take place per block, we can find the total chain run time to be

chain run time = (chain length target - 1) * [12 seconds, 30 minutes] + 30 minutes

So, what should the chain length target be?

The aim is to allow many to participate, to allow for a well distributed and decentralized allocation process. And, while keeping that in mind, the aim is to do this while limiting the energy to something rather insignificant.

Your answer would be different from mine, but I have landed on allowing for 10000 overclocked Raspberry Pi 4 Model B as the reference of choice. A Raspberry Pi 4 Model B is a relatively cheap computer, that doesn't consume much energy, and 10000 of these sound like a decent amount of participation to ensure a wide allocation distribution.

Could miners pick more expensive and powerful hardware? Of course they could, and they probably will. But keep in mind that if too many starts throwing too much energy at this token, they will reduce the reward, and their fancy hardware isn't going to be worth it. A Raspberry Pi is something you can buy for the fun of it, and if it doesn't work out as a miner, there's plenty of other fun projects to use it on. If you don't want to buy new hardware, just use some old computer you have lying around instead.

Calculating max chain length target

Having said we want to limit the chain length per chain run to be comparable to the power of 10000 overclocked Raspberry Pi 4 Model B, we need to run some experiments to find the relationship between hash rate and chain length.

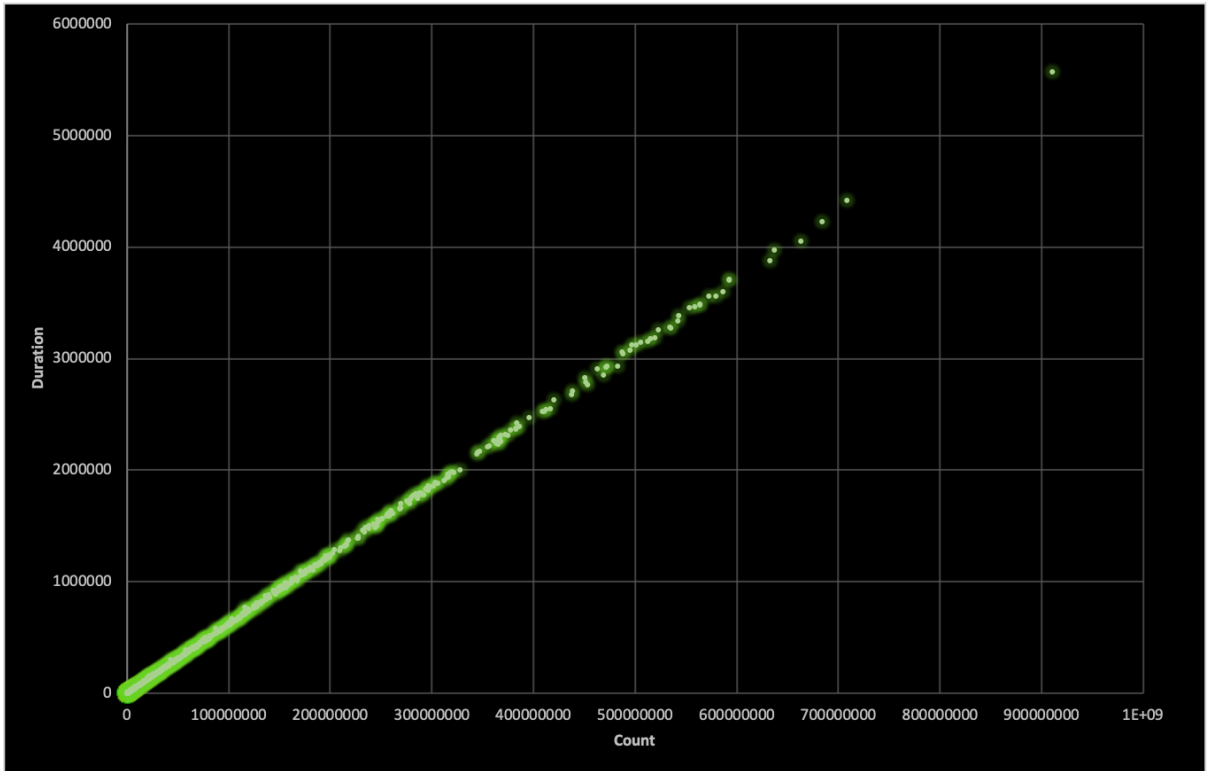
Because each participating miner is interested in committing their valid input as part of a chain, in order to be rewarded with tokens, it is in their interest to do so as soon as possible

after finding the first possible valid next input. Because of this, it means we can have some certainty around the fact that the least amount of effort is being spent before the next valid input is presented.

For example, after a chain has stalled and a new chain is started, the difficulty is practically non-existent on the first round, and it would not be in any miner's interest to spend time finding a more difficult first hash. Also, on the first few rounds, it will be trivial to find a valid input value, and it would be expected that the challenge is more focused on getting a valid transaction in before others.

Over time, efforts shift from getting the first transaction in, to finding a valid input value. The frequency of subsequent transaction reduces for each chain link.

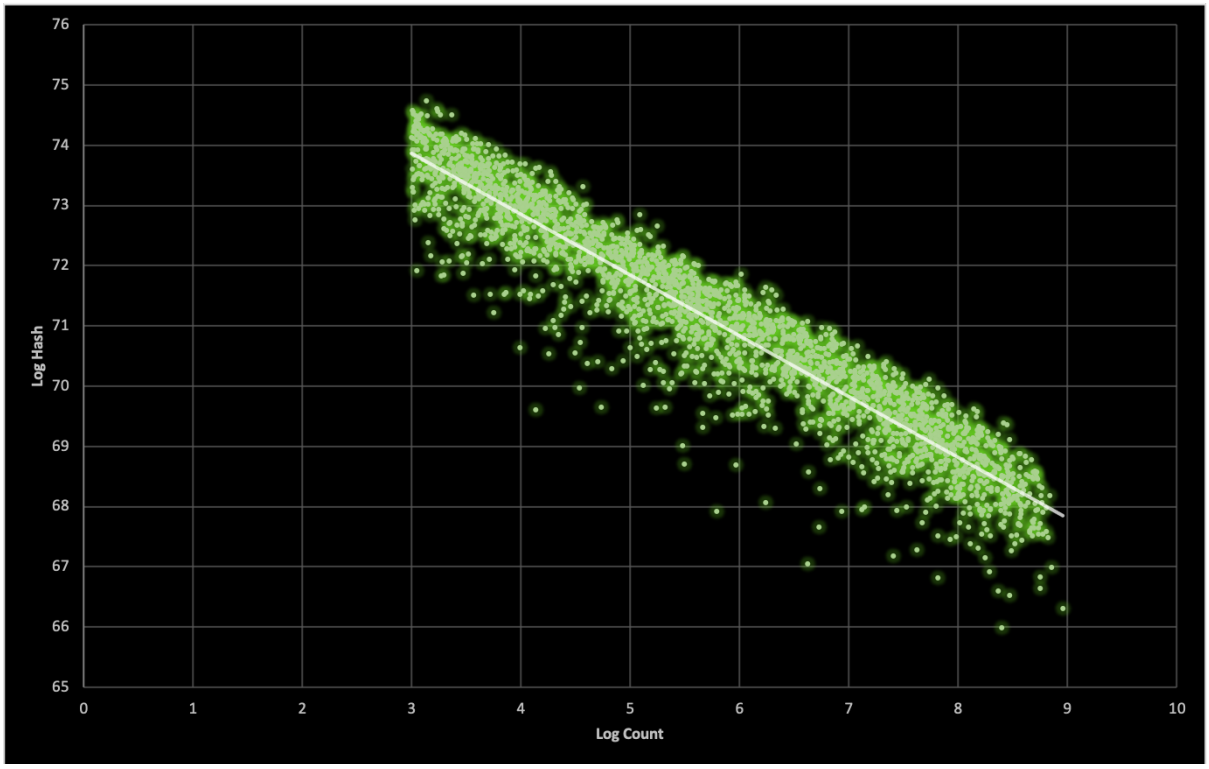
This in turn mean we have a valid expectation that each iteration of the chain will get progressively more difficult to participate in, but no more difficult that the least amount of effort required. This produces a predictable system, which in turn allows us to simulate the expected chain length when 10000 overclocked Raspberry Pi 4 Model B participate. This is good, as I have no intention of buying 10000 of these for my own experiment.



Miner RPi: Total hash count vs duration

As is known, the process of finding a hash is constant and independent. [7] With the number of hashes calculated going up, the total duration goes up with a linear dependency on the count. The relationship depends on the hardware and software used. For this experiment the NuPoW miner [8] HashPerf software is used to run the calculations. This is run in a single thread on a Raspberry Pi 4 Model B, overclocked to 2.1 GHz running in 64-bit mode.

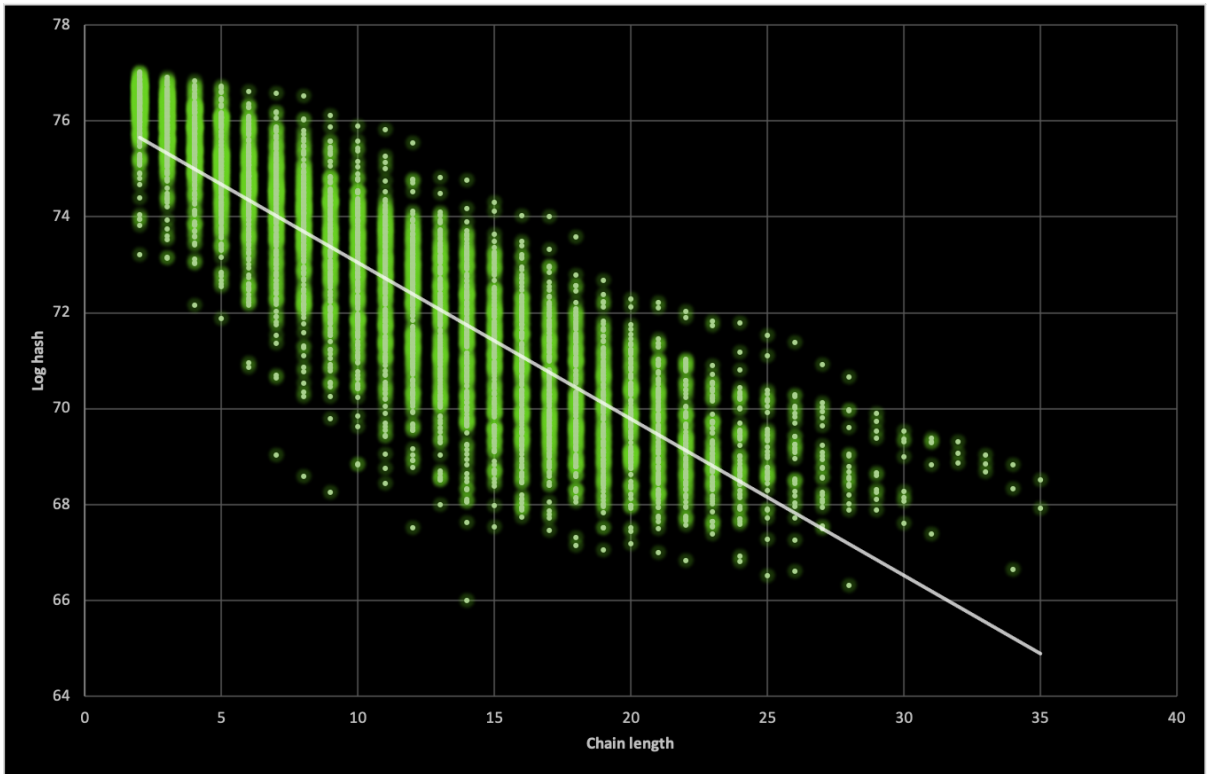
Using this software and hardware setup, and OLS, we get a slope of 0.00619604 with an intercept of 302.5575668. Since we have defined the stall duration to be 30 minutes, we can then calculate that the expected number of hash calculations possible on this setup is close to 290.5 million. [9]



Miner RPi: Total hash count vs hash

As can be expected, we also find that there's an exponential relationship between the total hash count and hash value. Log-linearizing this we find a slope of -1.0077178 with an intercept of 76.88239446. We found previously that a single thread on our setup could do close to 290.5 million hashes within the stall duration. As one Raspberry Pi 4 Model B has a quad-core CPU, we can then assume 10000 of these would be able to do 1.16184E+13 hashes within 30 minutes, under optimal circumstances. This gives us a minimum expected hash value of 5.20493E+63.

Since we can expect miners to perform the minimum amount of effort to find a valid next input, as outlined earlier, this then allows us to find a relationship between hash value and expected chain length for that hash value. This is again an exponential relationship which we log-linearize to find a slope of -0.326362478 with an intercept of 76.31517111. From this we calculate a chain length value of 38.6035689.



Miner RPi: Chain length vs hash

Some remarks about this calculated chain length for 10000 overclocked Raspberry Pi 4 Model B:

The observations from the experiment were taken over a 7-day time frame, with the HashPerf software running 24/7. The Raspberry Pi had otherwise spare CPU capacity across its 4 cores, so I have confidence in the number and quality of the observations.

It is assumed that the number of smaller hash values would be somewhat underrepresented in the data because they appear less frequently in this setup than they are expected to be had we actually had 10000 of these computers running. Because of this, I would assume the calculated chain length target to be biased, and should in reality be somewhat smaller, as more observations of the smaller hash values would drag the slope down. This would also explain why, when running the HashPerf software on a more powerful z1d.large AWS instance, also 24/7 over a 7-day period, we observe a slope almost 10% steeper.

With these observations, calculations, and the assumed bias, the official NuPoW chain length target value is defined as 37.

Encouraging valid and active mining participation

In addition to what's outlined already, certain other dynamics are at play to encourage valid and active participation.

Firstly, we must be careful to avoid that anyone be able to premine a chain, and through that be able to mine a chain where more than the least amount of effort is used. This because it would put them at an unfair advantage, hence centralizing the allocation process.

We avoid this by randomizing the starting point. When randomizing we use the hash of the previous block, combined with the CFELDE constant. This ensures 144 bits of randomness, while at the same time ensuring a high enough starting point to avoid stalling the chain early.

Secondly, we want to encourage active participation, and do so by introducing a counteracting motive. The official NuPoW tokens all support the ERC-3156 flash loan extension. Anyone making use of this extension can temporarily mint large quantities of NuPoW token for short term use within their transaction. Often, among other things, used to unlock arbitrage opportunities and remove token exchange inefficiencies, this brings utility. But this utility comes with a fee. The fee is set to be equal to the next token minting quantity.

This utility fee and the ability to influence it hence brings with it a motivation to reduce it, which can be done by overrunning the chain length target. While it is in no one's interest to drastically overrun the target, there exists motivation for slightly overrunning it, bringing the fee down. This then keeps us within the desired maximum energy usage, while encouraging mining activity.

References

[1] A. De Vries, 'Bitcoin’s growing energy problem', 2018

[2] H. Ritchie and M. Roser, 'Energy mix', https://ourworldindata.org/energy-mix, 2020

[3] S. Nakamoto, 'Bitcoin: A Peer-to-Peer Electronic Cash System', 2008.

[4] V. Buterin, 'Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform', 2014.

[5] ethereum.org, 'Proof-of-Stake (PoS)', https://ethereum.org/en/developers/docs/consensus-mechanisms/pos/, 2022.

[6] C. Felde, https://github.com/cfelde/nupow, 2022

[7] I.B. Damgård, 'A Design Principle for Hash Functions', 1990

[8] C. Felde, https://github.com/cfelde/nupow-miner, 2022

[9] C. Felde, https://github.com/cfelde/nupow-stats, 2022