Faster Ethash check against difficulty - HackMD

Check against boundary using integer division

Ethereum Yellow Paper in section 11.5 defines PoW difficulty check as

n \leq \frac{2^{256}}{H_d}

where n

is Ethash final hash and H_d

is current block difficulty.

The \frac{2^{256}}{H_d}

is called boundary

. In practical implementations integer division is used to validate the check.

n \leq \left\lfloor\frac{2^{256}}{H_d}\right\rfloor

This is correct because of the following fact (proof):

x \in \mathbb{R}, n \in \mathbb{Z}: n \leq x \iff n \leq \left\lfloor x \right\rfloor

Check against difficulty using multiplication

We can avoid using big integer division (which is slow and complex to implement) by transforming the orignal check formula into:

nH_d \leq 2^{256}

Benefits

1. Integer division has been replaced with multiplication. The 256 x 256 → 512 multiplication is straight forward to implement.
2. Degenerated values of difficulty (0

and 1

) do not require special handling.

Implementation

This has been implemented in ethash 0.8.0.

Side notes

1. The check nH_d \leq 2^{256}

can be further decomposed into nH_d \lt 2^{256} \lor nH_d = 2^{256}

where the first part is 256-bit multiplication overflow check and the second part is very unlikely (or even impossible considering the difficulty update formula).

1. The difficulty values on Ethereum Mainnet safely stay within 64-bit boundaries. Therefore, optimized path can be used for such values for both integer division and multiplication.