Mining

Notice

Becoming a miner is not recommended. Ethereum is going to transition to proof-of-stake, making mining obsolescent. Becoming a miner would involve investing in a mining rig (several GPUs, plus maybe other hardware if needed, like a compatible computer), which is unlikely to get a return on investment before PoS is implemented.

Introduction

The word mining originates in the context of the gold analogy for crypto currencies. Gold or precious metals are scarce, so are digital tokens, and the only way to increase the total volume is through mining it. This is appropriate to the extent that in Ethereum too, the only mode of issuance post launch is via the mining. Unlike these examples however, mining is also the way to secure the network by creating, verifying, publishing and propagating blocks in the blockchain.

- Mining Ether = Securing the network = verify computation

So what is mining anyway?

Ethereum Frontier, like all blockchain technologies uses an incentive-driven model of security. Consensus is based on choosing the block with the highest total difficulty.
Miners produce blocks which the others check for validity. Among other well-formedness criteria, a block is only valid if it contains **proof of work** (PoW) of a given **difficulty**.
Note that in Ethereum 1.1, this is likely going to be replaced by a **proof of stake** model.

The proof of work algorithm used is called Ethash (a modified version of Dagger-Hashimoto) involves finding a nonce input to the algorithm so that the result is below a certain threshold depending on the difficulty. The point in PoW algorithms is that there is no better strategy to find such a nonce than enumerating the possibilities while verification of a solution is trivial and cheap. If outputs have a uniform distribution, then we can guarantee that on average the time needed to find a nonce depends on the difficulty threshold, making it possible to control the time of finding a new block just by manipulating difficulty.

The difficulty dynamically adjusts so that on average one block is produced by the entire network every 12 seconds (ie., 12 s block time). This heartbeat basically punctuates the synchronisation of system state and guarantees that maintaining a fork (to allow double spend) or rewriting history is impossible unless the attacker possesses more than half of the network mining power (so called 51% attack).

Any node participating in the network can be a miner and their expected revenue from mining will be directly proportional to their (relative) mining power or **hashrate**, ie., number of nonces tried per second normalised by the total hashrate of the network.

Ethash PoW is memory hard, making it basically ASIC resistant. This means that calculating the PoW requires choosing subsets of a fixed resource dependent on the nonce and block header. This resource (a few gigabyte size data) is called a **DAG** (directed acyclic graph). The DAG is totally different every 30000 blocks (a 100 hour window, called an **epoch**) and takes a while to generate. Since the DAG only depends on block height, it can be pregenerated but if it is not, the client need to wait the end of this process to produce a block. Until clients actually precache DAGs ahead of time the network may experience a massive block delay on each epoch transition. Note that the DAG does not need to be generated for verifying the PoW essentially allowing for verification with both low CPU and small memory.

As a special case, when you start up your node from scratch, mining will only start once the DAG is built for the current epoch.

Mining Rewards

Note that mining 'real' Ether will start with the Frontier release. On the Olympics testnet, the Frontier pre-release , the ether mined have no value (but see Olympic rewards).

The successful PoW miner of the winning block receives:

- A **static block reward** for the 'winning' block, consisting of exactly 3.0 Ether
- All of the gas expended within the block, that is, all the gas consumed by the execution of all the transactions in the block submitted by the winning miner is compensated for by the senders. The gascost incurred is credited to the miner's account as part of the consensus protocol. Over time, it's expected these will dwarf the static block reward.
- An extra reward for including Uncles as part of the block, in the form of an extra 1/32 per Uncle included

Uncles are stale blocks, i.e., with parents that are ancestors (max 6 blocks back) of the included block.
Valid uncles are rewarded in order to neutralise the effect of network lag on the dispersion of mining rewards, thereby increasing security.
Uncles included in a block formed by the successful PoW miner receive 7/8 of the static block reward = 2.625 ether
A maximum of 2 uncles allowed per block.

Ethash DAG

Ethash uses a **DAG** (directed acyclic graph) for the proof of work algorithm, this is generated for each **epoch**, i.e every 30000 blocks (100 hours). The DAG takes a long time to generate. If clients only generate it on demand, you may see a long wait at each epoch transition before the first block of the new epoch is found. However, the DAG only depends on block number, so it CAN and SHOULD be calculated in advance to avoid long wait at each epoch transition. `geth` implements automatic DAG generation and maintains two DAGS at a time for smooth epoch transitions. Automatic DAG generation is turned on and off when mining is controlled from the console. It is also

turned on by default if `geth` is launched with the `--mine` option. Note that clients share a DAG resource, so if you are running multiple instances of any client, make sure automatic dag generation is switched on in at most one client.

To generate the DAG for an arbitrary epoch:

```
geth makedag <block number> <outputdir>
```

For instance `geth makedag 360000 ~/.ethash`. Note that ethash uses `~/.ethash` (Mac/Linux) or `~/AppData/Ethash` (Windows) for the DAG so that it can shared between clients.

The Algorithm

Our algorithm, Ethash (previously known as Dagger-Hashimoto), is based around the provision of a large, transient, randomly generated dataset which forms a DAG (the Dagger-part), and attempting to solve a particular constraint on it, partly determined through a block's header-hash.

It is designed to hash a fast verifiability time within a slow CPU-only environment, yet provide vast speed-ups for mining when provided with a large amount of memory with high-bandwidth. The large memory requirements mean that large-scale miners get comparatively little super-linear benefit. The high bandwidth requirement means that a speed-up from piling on many super-fast processing units sharing the same memory gives little benefit over a single unit.

Formal Requirements

TODO: Content from formal requirements doc.

Design Decisions Taken

TODO: Content from design decisions doc.

Infrastructure Overview

Mining will be accomplished in one of two ways: either on CPU (and possibly the GPU, to be confirmed) with the Mist client or on the GPU though a combination of the Ethereum daemon and sgminer.

An sgminer module for Ethash is expected to be released at some point during, but not necessarily before the Frontier Genesis.

JSON-RPC

Communication between the external mining application and the Ethereum daemon for work provision and submission happens through the JSON-RPC API. Two RPC functions are provided; `eth_getWork` and

`eth_submitWork` .

These are formally documented on the JSON-RPC API wiki article.

Comments

Write a new comment...

Your Name

Your Email Address

Markdown Format

Post Comment

Loading comments...