Weekly Report - Survey On Mining Algorithms

MSc Project
Runchao Han

November 14, 2017

1 Introduction

Currently Proof-of-Work (PoW) algorithm is the most popular consensus algorithm for public blockchains because of its scalability, where peers get coins by "mining" on the blockchain network. "Mining" in PoW based cryptocurrencies means to find hash values which satisfy the requirement to get the right to append a block on the blockchain. This report focuses on the first task of the MSc Project, which is to make a survey on state-of-the-art mining algorithms.

2 Background of PoW

This section introduces the theory of PoW from its cryptography basis to this probability based consensus, including Hash Functions, HashCash and PoW. In addition, its challenges and corresponding solutions are discussed.

2.1 Hash Functions

A hash function is any function that maps arbitrary size data to fixed size data. The values returned by a hash function are called hash values. An example of the the usage is a data structure called hash table, widely used in computer software for rapid data lookup. It can be expressed as:

$$H(x) = h$$

`H()` is the hash function which processes an arbitary string `x` then get a fixed length string `h`. Widely used hash functions are listed below:

1. Secure Hash Algorithm (SHA) family, including SHA-0 to SHA-3 with different parameters.
2. MD5 Message-Digest Algorithm

Since hash functions are essentially many-to-one functions for footprinting strings, some requirements should be met:

1. **Compression:** H can be applied to a block of data of any size, but produces a fixed-length output

2. **One-way property (pre image resistant):** H(x) is easy to compute for any given x. For any given h, it is hard to compute x such that H(x)=h

3. **Weak collision resistance (2nd preimage resistant):** Given x, it is hard to find y != x such that H(y)=H(x)

4. **Strong collision resistance (collision resistance):** It is hard to find two different messages, x != y, such that H(y)=H(x)

This online tool can do popular hash function calculations.

2.2 HashCash

HashCash was proposed to resist Denial of Service (DoS) Attack and spam emails. It requires a client to do a specific calculation which is hard to compute but easy to verify before invoking essential operations of the server. For example, a spam email sending machine should do this calculation every time before it sends an email, which will make the spam sending uneconomic, but not for normal individuals.

One-way functions like hash functions are suitable for implementing HashCash. For example, the server requires the client to find a value whose hash value starts with at least three zeros. Both clients and servers do not know answers at first. The client continuously generates random strings and computes hash values of them to find a valid string. Based on the random nature of hash functions, the probability of finding a valid value is fixed so that the required work of the client is also fixed.

2.3 PoW Consensus - Double SHA256 in Bitcoin as an Example

Supported by the HashCash mechanism, the consensus is obtained on a peer-to-peer (P2P), unstable and untrusted network by PoW. According to the example above, HashCash can change the difficulty simply by changing required number of 0 at the beginning of hash values. On the Bitcoin network, every node calculates HashCash values, where the difficulty is dynamic according to the total calculating power on the network. That is, if the total computing power is bigger, the difficulty will increase, which always keeps the average time to find a valid value about 10 minutes. This process is called “Mining” because it is similar to mine gold in a goldmine.

The time of 10 minutes is an average value based on probabilitic theories which is a practical way to avoid the double spending problem. 10 minutes is enough for nodes in the whole network to receive the latest block when no other valid hash values are found.

The hash function of Bitcoin’s PoW is Double SHA256:

$$H(x) = SHA256(SHA256(x))$$

2.4 Challenges of PoW - Centralisation

At the beginning of Bitcoin, only CPU is used for

2.5 Solutions

3 State-of-the-art Democrat PoW Algorithms

3.1 Scrypt

3.2 Ethash Based on DAG Generation

3.3 Cuckoo Based on Graph Theory

3.4 Momentum Based on Hash Birthday Collision

3.5 MemoHash

4 Planned Accomplishments

1. New item: give internal deadline for deliverable.
2. Person and Scheduled Task name:
   - hours spent by person
   - description of what was done (task doesn't need to be complete)
3. Scheduled Task name:
   - hours spent by person
   - description of what was done (task doesn't need to be complete)

5 Other Accomplishments

-
-

6 Next Week's Plan

-
-

7 Issues

-
-