大数据与云计算期末报告
Mining

王笑
前沿交叉学科研究院
1701214262

June 25,2018

Outline

1. What is Mining?
   - Mining
2. Why to Mine?
   - Monetary Supply Mechanism
   - Decentralized Emergent Consensus Mechanism
3. How to Mine?
   - Hardware
   - Algorithm
4. Who is Mining?
   - Mining Nodes

Outline

Mining

Mining

Mining is the process of hashing the block header repeatedly, changing one parameter, until the resulting hash matches a specific target.

Prepare:

- One Block Example
- Block Header

Block

Block #528786

Summary	
Number Of Transactions	1485
Output Total	3,160.1394646 BTC
Estimated Transaction Volume	593.57869142 BTC
Transaction Fees	0.28929455 BTC
Height	528786 (Main Chain)
Timestamp	2018-06-23 04:59:34
Received Time	2018-06-23 04:59:34
Relayed By	BTC.com
Difficulty	5,077,499,034,879.02
Bits	389508950
Size	846.327 kB
Weight	2981.622 kWU
Version	0x20000000
Nonce	2472214566
Block Reward	12.5 BTC

Hashes	
Hash	00000000000000000210c34666e6636eb92e470f5c5d7cf11fb57b11f1c1848
Previous Block	00000000000000000002ddaa3aa588fe06bff8857d4f31db79a650015f08845e5
Next Block(s)	
Merkle Root	1375b3a02c1f1b681b89edbd1996e27b2d555b6ecb8fbe0898a97bebfd011599

Transactions

f6282fc4bf42b87d12f008797f5ec52d2ae8c3066d6b43a987f0e3208d4f81b4 (Size: 243 bytes) 2018-06-23 04:59:34

No Inputs (Newly Generated Coins) → 1C1mCxRukix1KfegAY5zQQJV7samAciZpv - (Unspent) 12.78929455 BTC
Unable to decode output address - (Unspent) 0 BTC

12.78929455 BTC

5f38e303c45740ef2ca3fe8e04aae9a78080a8bec700162c5c0d5f082ae2a8d0 (Fee: 0.0031551 BTC - 306.92 sat/WU - 1,227 sat/B - Size: 257 bytes) 2018-06-23 04:47:51

1L5Cc8wfH48xfpjapFEPejsjWurky16tjS (1.15367468 BTC - Output) → 1L5Cc8wfH48xfpjapFEPejsjWurky16tjS - (Spent) 1.15051412 BTC
Unable to decode output address - (Unspent) 0 BTC
1Bhi27LbYB3Le4UZdpdoCbPnoq82o1d2Xu - (Unspent) 0.00000546 BTC

1.15051958 BTC

Ref :https://blockchain.info/block

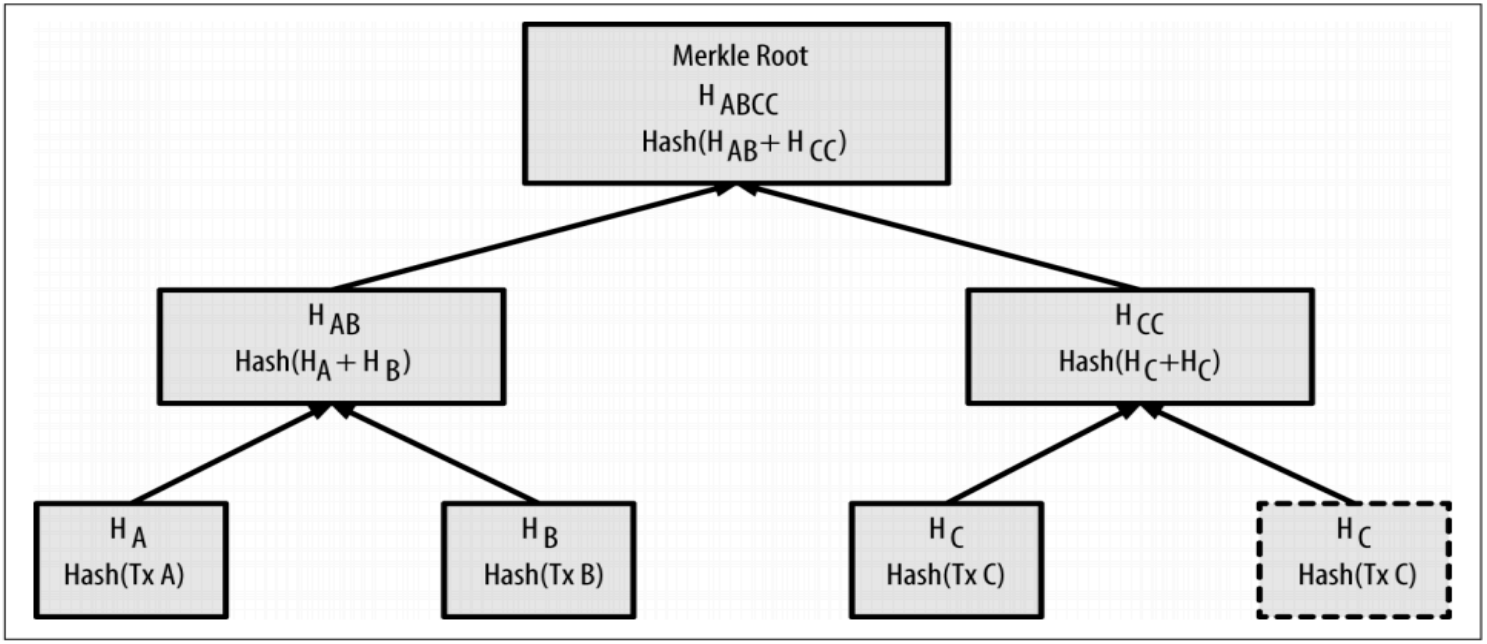
Block Header

Field	Description	Size
Version	A version number about software/protocol upgrades	4 bytes
Previous Block Hash	The hash of last block (only the header)	32 bytes
Merkle Root	the hash of the root of merkle tree of this block's transactions	32 bytes
Timestamp	The approximate creation time of this block(Unix Epoch)	4 bytes
Difficulty Target	The difficulty of this block	4 bytes
Nonce	A random number	4 bytes

- The block header is 80 bytes totally
- Previous Block Hash and Merkle Root 均为 32 bytes(SHA256)

Block Header

- **Merkle Root**: merkle tree is a binary tree



- **Timestamp**: seconds from Unix Epoch: 自 1970.1.1 00:00:00 时 (UTC/GMT) 以来的秒数
  Time : 2018-06-23 04:59:34 Actual: 2018-06-23 12:59:34
  Epoch timestamp:1529729974
- **Difficulty Target**: also "difficulty bits" (we'll see later)
- **Nonce**: a random number that miners to find

Extra Nonce

- **The Reason**
  - nonce in block header: 4 bytes $2^{4*8} = 2^{32} = 4.2$billion
  - As difficulty increased, miners often cycled through all 4 billion values of the nonce without finding a block
- **The Solution**
  1. **Updating the block timestamp**

     When mining hardware exceeded 4 GH/s, it failed.
  2. **Using the coinbase transaction as a source of extra nonce values**

     The coinbase transaction is included in the merkle tree
     No "scriptSig" field⇒"coinbase" data(2-100 bytes)
     change the coinbase data ⇒ change merkle root

     Extra Nonce(8 bytes) + "Nonce"(4 bytes) $\Rightarrow 2^{96} = 7.92 * 10^{28}$
  3. **Future**: more space in the coinbase script

Outline

Monetary Supply Mechanism

供币机制

- The only way to create new BTC
- In Coinbase
- That's the incentive system for miners
- Total Reward = Mining Reward + Transaction fees

Mining Reward

- **Mining Reward**: The initial reward is 50 BTC, it halves every 210,000 blocks

```
CAmount GetBlockSubsidy(int nHeight, const Consensus::Params& consensusParams)
{
    int halvings = nHeight / consensusParams.nSubsidyHalvingInterval;
    // Force block reward to zero when right shift is undefined.
    if (halvings >= 64)
        return 0;

    CAmount nSubsidy = 50 * COIN;
    // Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years.
    nSubsidy >>= halvings;
    return nSubsidy;
}
```

Ref: bitcoin/src/validation.cpp line 1180

- typedef int64_t CAmount
- Total $\approx 21,000,000$; acturally, $210000 * 50 * (1 + \cdots + 1/2^{63})$
- in 2140

Transaction Fees

$$\boxed{\textbf{Transaction fees} = \text{Sum(Inputs) - Sum(Outputs)}}$$

- 按照 Priority 的大小添加交易至区块

$$\textbf{Priority} = \frac{\text{Sum(Value of Input * Input Age)}}{\text{Transaction Size}}$$

- "High Priority" 交易可以不交 Fee

$$\text{High Priority} > \frac{100,000,000\text{satoshis} * 144\text{blocks}}{250\text{bytes}} = 57,600,000$$

即 value = 1BTC(10^8 satoshis), age = 1 day(144 blocks), size = 250 bytes 的 Priority。

- Block has 50KB space for transactions with High Priority and Carrying no fees

Outline

Decentralized Emergent Consensus Mechanism

去中心化自发共识机制

- Mining is the main process of the decentralized clearinghouse.
- verify blocks
  - The block header hash is less than the target
  - The block timestamp is less than two hours in the future (allowing for time errors)
  - All transactions within the block are valid

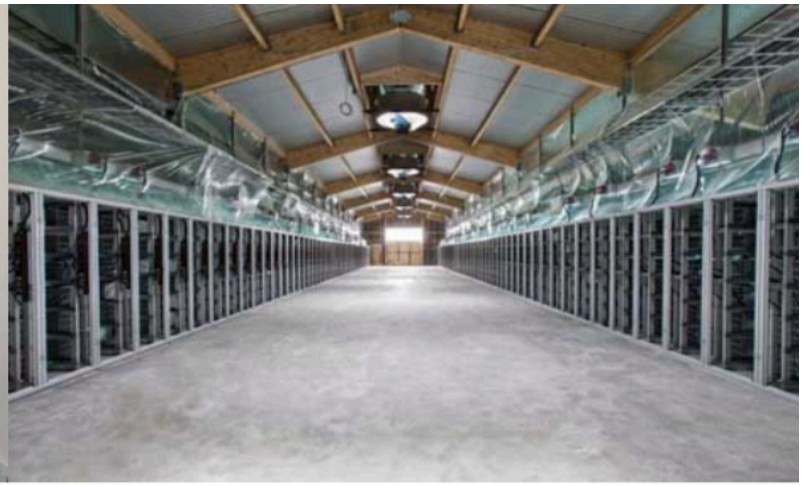
Outline

Hardware

CPU

GPU

FPGA



ASIC

Hardware

Total Hash Power

2009
0.5 MH/sec–8 MH/sec (16× growth)

2010
8 MH/sec–116 GH/sec (14,500× growth)

2011
16 GH/sec–9 TH/sec (562× growth)

2012
9 TH/sec–23 TH/sec (2.5× growth)

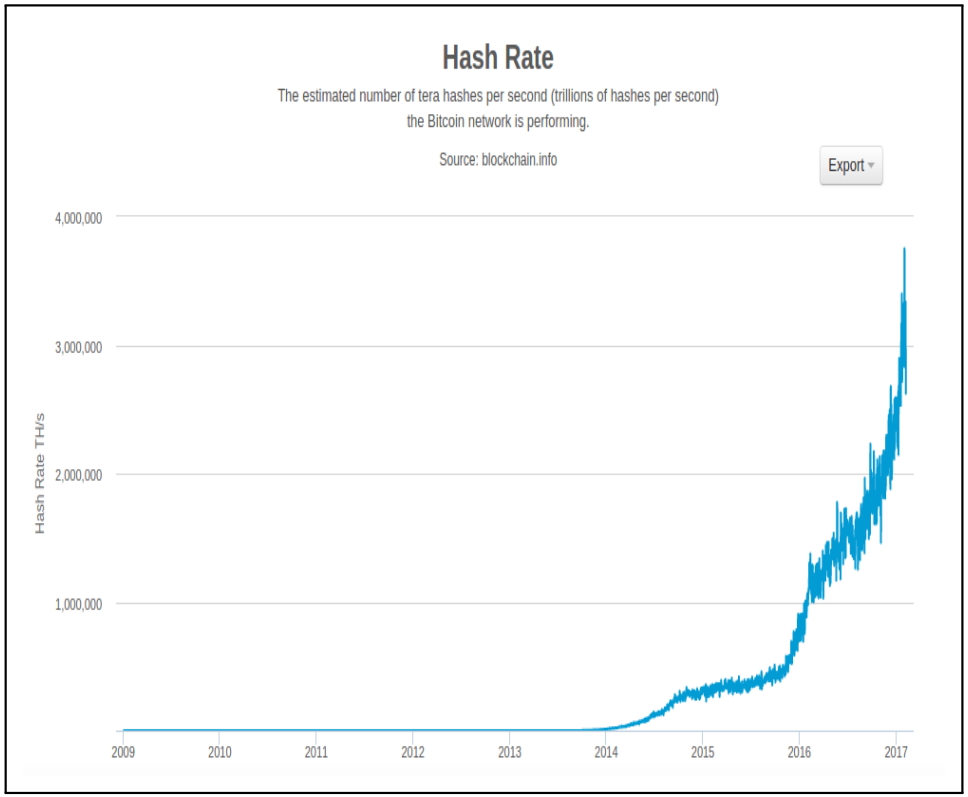
2013
23 TH/sec–10 PH/sec (450× growth)

2014
10 PH/sec–300 PH/sec (3000× growth)

2015
300 PH/sec-800 PH/sec (266× growth)

2016
800 PH/sec-2.5 EH/sec (312× growth))

Ref : Mastering Bitcoin 2nd Edition P247



Ref :`https://blockchain.info`

Outline

Proof-Of-Work Algorithm

- **POW**:Try different random number, to get a hash of block header that is less than the target.

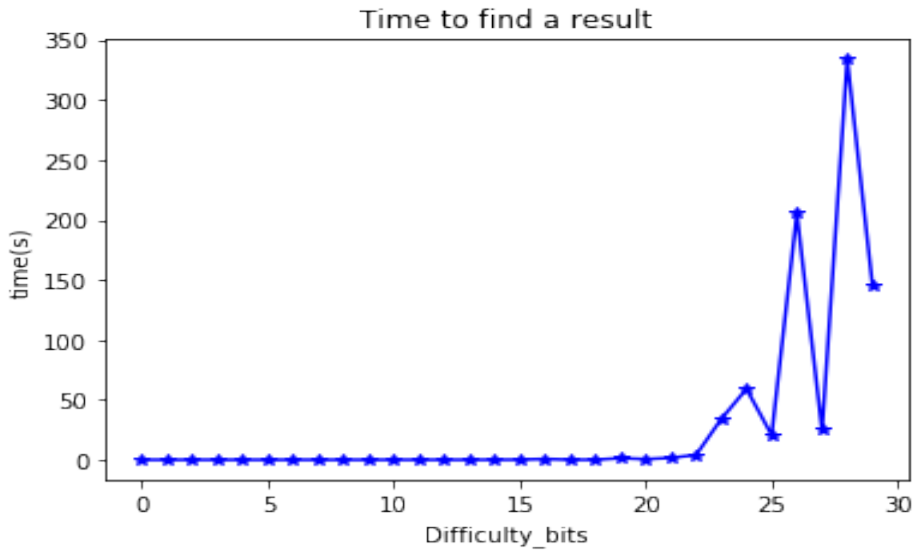
```
max_nonce = 2 ** 32 # 4 billion
def proof_of_work(header, difficulty_bits):
    target = 2 ** (256-difficulty_bits)
    for nonce in range(max_nonce):
        hash_result = hashlib.sha256((str(header)+str(nonce)).\
                                     encode('utf-8')).hexdigest()
        if int(hash_result, 16) < target:
            return (hash_result,nonce)
    print("Failed after %d (max_nonce) tries" % nonce)
    return nonce
```

Ref: Master Bitcoin

Proof-Of-Work Algorithm

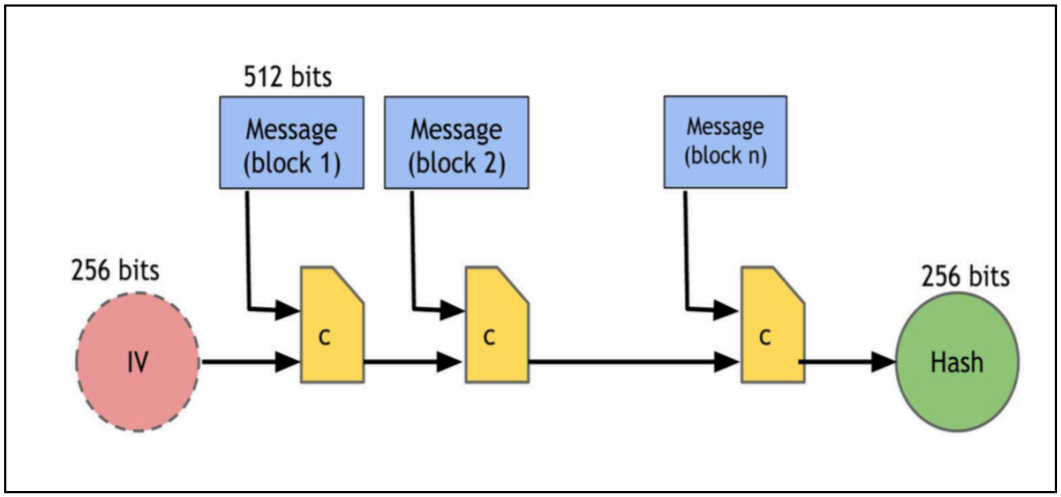
Experiment in Python:

```
nonce = 0
hash_result = ''
for difficulty_bits in range(30):
    difficulty = 2 ** difficulty_bits
    print("Difficulty: %ld (%d bits)" % (difficulty, difficulty_bits))
    start_time = time.time()
    new_block = 'test block with transactions' + hash_result
    (hash_result, nonce) = proof_of_work(new_block, difficulty_bits)
    end_time = time.time()
    elapsed_time = end_time - start_time
    print("Elapsed Time: %.4f s" % elapsed_time)
```



- increasing the difficulty by 1 bit causes an exponential increase in the time to find a solution
- Each time you constrain one more bit to zero, you decrease the target space by half.

SHA256 Algorithm



Ref : Bitcoin and Cryptocurrency Technologies

- IV⇒ Initialization Vector
  c⇒compression function
- The compression function takes 768-bit input and produces 256-bit outputs

5.3.3 SHA-256

For SHA-256, the initial hash value, $H^{(0)}$, shall consist of the following eight 32-bit words, in hex:

$$H_0^{(0)} = \texttt{6a09e667}$$
$$H_1^{(0)} = \texttt{bb67ae85}$$
$$H_2^{(0)} = \texttt{3c6ef372}$$
$$H_3^{(0)} = \texttt{a54ff53a}$$
$$H_4^{(0)} = \texttt{510e527f}$$
$$H_5^{(0)} = \texttt{9b05688c}$$
$$H_6^{(0)} = \texttt{1f83d9ab}$$
$$H_7^{(0)} = \texttt{5be0cd19}$$

https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.180-4.pdf

- Divide the message into blocks of length 512
- only 8 IV

Difficulty Target

- The difficulty target, called "difficulty bits" or just "bits"
- 4 bytes⇒ 8 hexadecimal digits
  first two hexadecimal digits⇒ the exponent
  the next six hex digits ⇒ the coefficient
- **target** = coefficient * 256 ^(exponent - 3)
- **Example**:
  Block # 528786 389508950
  To hexadecimal: 0x17376f56
  17⇒23 376f56 ⇒ 3632982
  $3632982 * 256^{23-3} = 3632982 * 256^{20}$

Target	0000000000000000376f560000000000000000000000000000000000000000
Hash	0000000000000000210c34666e6636eb92e470f5c5d7cf11fb57b11f1c1848

Difficulty Adjustment

- **The Reasons**
  - Computer power continues to increase at a rapid pace
  - The number of miners constantly changes
- **The Aim**
  - To make the block generated every 10 minutes , on average
- **The Method**
  - Every 2,016 blocks, all nodes retarget the proof-of-work difficulty.

$$\text{New Difficulty} = \text{Old Difficulty} * \frac{\text{Actual Time of Last 2016 Blocks}}{\text{1,209,600 seconds(2 weeks)}}$$

Difficulty Adjustment

```
// Limit adjustment step
int64_t nActualTimespan = pindexLast->GetBlockTime() - nFirstBlockTime;
if (nActualTimespan < params.nPowTargetTimespan/4)
    nActualTimespan = params.nPowTargetTimespan/4;
if (nActualTimespan > params.nPowTargetTimespan*4)
    nActualTimespan = params.nPowTargetTimespan*4;

// Retarget
const arith_uint256 bnPowLimit = UintToArith256(params.powLimit);
arith_uint256 bnNew;
bnNew.SetCompact(pindexLast->nBits);
bnNew *= nActualTimespan;
bnNew /= params.nPowTargetTimespan;

if (bnNew > bnPowLimit)
    bnNew = bnPowLimit;
```

Ref: bitcoin/src/pow.cpp line 49

- The retargeting adjustment must be less than a factor of 4 per cycle

Outline

Mining Nodes

- Solo Miners: Must be a full node
- Miners in Pool: Don't have to be a full node

Mining Pools

- **What?**
  Miners collaborate to form mining pools, pooling their hashing power and sharing the reward, pay the reward to a pool address(Coinbase)
- **Why?**
  - reducing uncertainty
  - the ability to mine without running a full node
- **How?**
  - The miners
  - The pool server: a company or individual

Managing Pools

The pool server:

- To do:
  - must be a full node, validate blocks and transactions
  - constructs the header of the candidate block, send to miners
  - sets a lower difficulty target for earning a share, typically more than 1,000 times easier
- To get:
  - charges a percentage fee of the rewards for providing the pool-mining service

The miners:

- To do:
  - Calculate the nonce to meet the pool target
- To get:
  - Share the block reward in proportion to the number of shares.

P2Pool

Peer-to-peer mining pool (P2Pool)

- The reason why P2Pool emerged:

  The pool operator maybe cheat, might include double-spend transactions or invalidate blocks

- Almost similar to solo miners, miners are full nodes

Summary

- What?
  - To find a nonce making the hash of block header less than target
- Why?
  - Monetary Supply Mechanism
  - Decentralized Emergent Consensus Mechanism
- How?
  - Evolution of Machine
  - POW
- Who?
  - Solo Miners
  - Mining Pools

Reference Book I


Andreas M. Antonopoulos
Mastering Bitcoin.
O'Reilly Media, 2017.


Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, Steven Goldfeder
Bitcoin and Cryptocurrency Technologies.
Princeton , Feb 9, 2016.

Thanks for Attention!