Lesson 21

MINING POOLS
BLOCKCHAIN BEYOND BITCOIN

Laura Ricci
30/05/2017

some slides taken from
http://www.chainfrog.com/
THE MINING PROCESS

- C++ code for mining:


  The core cycle
  
  ```
  while (1)
    HDR[kNoncePos]++;
    if (SHA256(SHA256(HDR)) < (65535 << 208)/ DIFFICULTY)
      return;
  ```
**MINING HARDWARE**

- SHA256 computation:
  - takes 256 bits blocks and performs **80 rounds** of basic encryption operations
    - several long chains of 32-bit **additions, rotates, and bitwise functions including xors, majority, max functions**.
  - chain of dependencies: **each round is dependent on the last round**
    - separate rounds cannot be executed in parallel
  - ...but, each separate nonce trial can be performed in parallel and some of the operations inside a round are parallelizable.
MINING HARWARE DIFFICULTY

Difficulty

- GPU
- FPGA
- ASIC

Dipartimento di Informatica
Università degli Studi di Pisa
FIRST GENERATION OF MINING: CPU

- CPU mining on general purpose computers
- Sequential search of right nonces or a few core parallelization
- SHA 256 computed in software (no specialized hardware)
- Even if a high-end desktop computer is exploited, it would take several hundred of thousand years on average to find a block

139,461 years to find a block in 2015!
GPU MINING: THE SECOND GENERATION

- Mining with Graphical Processing Units (GPU)

- GPUs designed for high-performance graphics
  - high parallelism: compute in parallel different hashes with different nounces
  - high throughput
  - possibility of overcloaking

- October 2010: first OpenCL miner
  - upset in the normal GPU market which is gaming
  - gamers becoming miners mining

- Underuse of the GPU functionalities, consuming a lot of power.

- ...and even if about one hundred GPU, it would take 300 years, on the average, to find a block
FPGA MINING: THE THIRD GENERATION

- Mining with **Field Programmable Gate Area (FPGA)**
  - can be programmed in Verilog, a hardware design language
    - performance as close as possible close to the custom hardware
    - possible to customize the card or reconfigure it “in the field”
  - higher performance than GPUs
    - excellent performance on bitwise operations
  - better cooling, but more expensive than GPUs and marginal performance/cost advance over GPUs.

- in any case, 25 years, on the average, to find a block!
ASIC MINING: THE FOURTH GENERATION

• Mining with **ASIC Application Specific Integrated Circuits**

• special purpose hardware
  • chip designed and built only for Bitcoin mining

• A few big vendors

• designed and produced very quickly, first ones not reliable
BitFury mining center, Republic of Georgia
Mining is a very risky task, even if the reward is high (6500 $ in 2015)

<table>
<thead>
<tr>
<th># blocks found in one year</th>
<th>probability (Poisson dist.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>42.4%</td>
</tr>
<tr>
<td>1</td>
<td>36.4%</td>
</tr>
<tr>
<td>2</td>
<td>15.6%</td>
</tr>
<tr>
<td>3+</td>
<td>5.6%</td>
</tr>
</tbody>
</table>

- high probability of spending a lot for mining hardware and electricity without obtaining a reward for a long time
THE LAW OF BITCOIN MINING

- Huge incertitudes:
  - law of Bitcoin mining follows the Poisson Distribution
  - large standard deviation
  - if for example in 1 month the miner expect to find 4 blocks, the standard deviation is about $\sqrt{4}=2$.
  - in one month he will find 6 is some months he will find 2, sometimes he will find 0

- Very stressful: cannot sleep at night.
  - does my miner work correctly??? wait for x years to see...
  - are other miners cheating? am I getting a fair share???
  - ...and YES, as we will see later, miners can cheat and earn more than other miners....
MINING POOLS

- an old idea: mutual insurance to lower the risk. Mine cooperatively in pools
- instead of waiting 1 year to get 25 BTC, why not get a little money every day?
- lower the variance for participating miners
MINING POOLS

- an old idea: mutual insurance to lower the risk. Money cooperatively in pools
- pool: a group of small/larger miners who work together
- pool manager collects transactions and assemble them in a block. The block is sent to all the participants of the pool
- works well if the miner trusts the pool manager
MINING POOLS

- goal: pool participants all attempt to mine a block with the same coinbase recipient:
  - money is sent to the pool manager

- distribute revenues to members based on the work they have performed:
  - minus a cut for pool manager

- problem: how to know how much work members perform?
  - idea: probabilistically prove work with “shares” that are “near valid blocks”
MINING POOLS VARIATIONS

- **pay per share**: flat reward per share
  - guarantees a reward without waiting for the pool to find a block
  - risk absorbed by the pool manager which charge a fee to the participant
  - disadvantage: workers do not have valid incentives to send valid blocks to the manager

- **proportional**: when a valid block is found, the rewards are distributed to the members in proportion to the work done
  - lower risk for pool manager: only pay when a block is found
  - workers incentivised to send valid blocks
MINING POOLS HISTORY

- first pools appear in late-2010, back in the GPU era!
  - by 2014: around 90% of mining pool-based
  - miners tend to flock to the largest pools.

- June 2014: one pool, Ghash.io, has in 2014 reached 55% 
  - possibility of 50% attack = total control of bitcoin by one single entity.
  - they have publicly said: please leave, do not join!

<table>
<thead>
<tr>
<th>pool</th>
<th>percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>BTCGuild.com</td>
<td>26%</td>
</tr>
<tr>
<td>GHash.io</td>
<td>40%</td>
</tr>
<tr>
<td>Eligius.st</td>
<td>10%</td>
</tr>
<tr>
<td>Bitcoin.cz</td>
<td>6%</td>
</tr>
<tr>
<td>Bitminter.com</td>
<td>5%</td>
</tr>
</tbody>
</table>

Ukraine, moved to UK
Hash Power by mining pools, May 2017, source blockchain.info

current situation:
hash power more equally distributed
standardized protocols ease switch between pools for miners
in the history several new technologies changed the way people lived.
- recent ones: personal computers, internet, smartphones,.....

- according to experts another big change is coming brought by blockchain technology
  - based on the cryptocurrency technology

- what is the breakthrough? current world is based on the existence of “trusted parties”
  - represent an overall weaken of any system
  - are justified as necessity by governments and other "trusted authorities": systems cannot work without them...

- the distributed nature and lack of central authority of blockchains can:
  - shift the power from one to many
  - move from central concept to the distributed one in several fields
why Bitcoin is considered a disruptive technology?

- possibility of performing transactions without having to trust any central party has been unimaginable for a long time until Bitcoin emerged

- Bitcoin has showed for the first time, that it is possible that two parties:

  - who do not necessarily trust each other
  
  - to perform transactions without any central authority as intermediary
  
  - the parties only trust only the underlying architecture which guarantees the security of the system.
Only some of the numerous proposals which are around:

- **Smart Contracts**

- **Supply chain communications and proof-of-provenance**

- **Distributed Cloud Storage (Storej)**

- **Digital Identity**: authentication and authorization

- **Digital Voting**

While the Bitcoin blockchain technology remains the basic tool, several improvements/changes must be done, when developing these blockchain based applications.
• **Permissionless Blockchain**
  - anyone can participate in the verification process, i.e. no prior authorization is required to perform such a task
  - a user can contribute his/her computational power, usually in return or a reward/incentive
  - anonymity/pseudoanonymity
  - scalability problems
  - examples: *Bitcoin* and *Ethereum*

• **Permissioned Blockchain**
  - verification nodes are preselected by a central authority, or consortium
  - verifiers are named can be accounted
  - a smaller number of participants, higher scalability
  - examples: *Eris*, *Hyperledger*, and *Ripple*. 
• **Public blockchains**
  
  • anyone can read and submit transactions to the blockchain.

• **Private blockchains,**
  
  • only the users within an organisation or group of organisations can access the blockchain
 BLOCKCHAIN BEYOND BITCOIN

• a first step forward with respect to Bitcoin: *smart contracts*

• proposed by Nick Szabo in 1994

  • an abstract concept relating to the automated execution of an already agreed contract

  • can be realised in blockchains through a contract scripting language

• an existing blockchain for smart contracts: *Ethereum*
A simple example: a bank account may behave like smart contracts

- my bank account has a balance
- every month, I commission an automated payment that deducts a fixed amount and sends it to my landlady
- if there is not enough money in my bank account, the payment fails
- I get fined, and another procedure is triggered.

these are the instructions I have set up which are associated with the account

- a contract between me and the bank

- a **smart contract** is similar to a real contract, but:
  - runs on a blockchain
  - is run by many parties rather than being controlled by a single one.
• a smart contract is some code automating the “if this happens then do that” part of traditional contracts.

• better than a normal contract
  • computer code behaves in expected ways and doesn’t have the linguistic nuances of human languages.
  • code is replicated on a blockchain all the nodes execute the contract and come to an agreement on the results of the code execution.
SMART CONTRACTS: CHARACTERISTICS

Control

Centralized: my bank is the ultimate guardian of my bank account and has complete control:

• can arbitrarily add money to my account (ha...ha..), subtract it: it may happen!
• code sits on one computer and is executed by one party (the bank) which must be trusted

Distributed: smart contracts

• distributed blockchain and consensus mechanisms assure that multiple nodes are executing my contract
• results are compared by all nodes and they only change their own version of the ledger if they agree the results.
• anything that doesn’t conform to pre-agreed rules is rejected by other participants
SMART CONTRACTS: CHARACTERISTICS

Transparency

- all participants in a blockchain run the same code, each verifying the other
- the logic of the smart contract must be visible to all
- privacy may be an issue
  - solutions based on zero-knowledge proofs

Flexibility

- smart contract can be written in a “Turing complete” language
- can do anything that a normal computer can do
- but you need to pay for all nodes on the network to run the code in parallel.
  - the blockchain version will be more expensive to run than on a regular computer
A GENTLE INTRODUCTION TO ETHEREUM

- a software running on a network of computers that ensures that data and small computer programs called smart contracts are
  - replicated and processed on all the computers on the network, without a central coordinator

- extends the blockchain concepts from Bitcoin
  - runs computer code equivalently on many computers around the world.
  - what Bitcoin does for distributed data storage, Ethereum does for distributed data storage + computations (smart contracts)

- similarity with Bitcoin:
  - manages a blockchain with blocks of data and smart contracts
  - the blockchain is public and permissionless
  - has a Proof of Work mining (even Proof of Stake is proposed)
  - has a native cryptocurrency: ether (ETH)
Differences with respect to Bitcoin:

- time between blocks is around 14 seconds, compared with Bitcoin’s ~10 minutes

- has smaller blocks

- built-in Turing complete programming language

- definition of the Ethereum Virtual Machine: a planetary scale computer

- consume gas to execute contracts

- pay gas with ether
ETHEREUM AND SMART CONTRACTS

Contracts

• live on the Ethereum blockchain

• have their own Ethereum address and balance
  • set up a smart contract by creating a new account with some code in it, and uploading it to the Ethereum blockchain in a transaction.

• can send and receive transactions

• are activated when they receive a transaction, and can be deactivated

• have a fee per CPU step

• the Ethereum Virtual Machine runs a Turing complete language
ETHEREUM AND SMART CONTRACTS

- a contract like a jukebox:
  - it can be activated and run, by funding it with some ETH
  - to run create a transaction sending a payment of ETH to the contract, and possibly supplying some other input information
  - each miner runs the smart contract
  - if no one is behaving badly, each computer on the Ethereum network will produce the same output
  - the winning miner will publish the block to the rest of the network,
    - the other computers validate that they get the same result
    - if so, they add the block to their own blockchains.
  - the state of Ethereum’s blockchain so gets updated.
ETHEREUM AND SMART CONTRACTS

ETHEREUM NODE

Voting Smart Contract

function vote (....{  
voteCount

A

B

C

D

Transaction

Transaction

Transaction

Transaction

Updates the state of the blockchain
Ether buys GAS to fuel the EVM

Every opcode instruction executed by the EVM uses up Gas.
the scenario:

• Alpha corporation designs and oversees the manufacture of complex multi-part equipment, for example for heavy industry.

• the equipments components are produced and assembled in one of Alpha’s factories.
the scenario

- the equipment is shipped to different remote locations and used heavily. Regular servicing and maintenance is required to comply with local safety rules and legislation.

- Alpha contracts out equipment maintenance to authorized third parties, who use certified service engineers and approved replacement parts
COMPONENT SERVICING USE CASE

The scenario

- equipment may be sold from one corporation to another, and a record of the service history and provenance is vital
- at the end of the life-cycle the equipment is decommissioned
COMPONENT SERVICING: TRADITIONAL SET UP

- A1 runs a database tracking components, equipments, locations, service histories and life-cycle.

- A denial of service attack on a database (or even database server farm) can put the operation serious risk.

- Records in any databases can be altered or deleted at a later date.

- Processes such as replacement part orders are manual, requiring human memory and intervention.
**COMPONENT SERVICING: TRADITIONAL SET UP**

- sharing access to the databases may be limited or need to be blocked by one or more of the participants at any time without notice.

- difficult to obtain:
  - compliance and third-party auditing
  - access permission granting and revocation
  - interoperability between different systems

- disputes can be difficult to resolve at a later date
  - costly and protracted legal action.
THE BLOCKCHAIN SOLUTION

Component tracking and servicing on a blockchain has several advantages

- **blockchains are distributed**: different nodes in different locations run the software powering the blockchain and keep a copy of the data. The more nodes there are:
  - the more backup copies of the data there are
  - the more backup “servers” there are
  - denial of service attacks become impossible

- **blockchains are tamper-proof**: once a component record has been on the blockchain for a short time it is locked down, and cannot be changed.
  - hacking the data records becomes impossible
  - “delete and deny” defenses become impossible
  - auditing is built into the system by running a simple scan over the blockchain
THE BLOCKCHAIN SOLUTION

Component tracking and servicing on a blockchain has several advantages:

- **Blockchains can support “smart contracts”**

  - Components can have their required service and replacement history pre-loaded onto the blockchain using a “smart contract”
  
  - When the scheduled service date or wear-and-tear usage limit is reached, the system automatically triggers a service request.

  - On completion of the service or replacement, a record is generated on the blockchain.
INITIALIZING THE BLOCK CHAIN

- node Alpha:
  - starts a “permissioned blockchain” with a “genesis block”
  - runs the first block on the chain on a computer in its head office

- the genesis block contains the announcement message of Alpha’s first public key
  - this key will identify Alpha head office on the blockchain in future.

- the blockchain runs, and at regular intervals new blocks are added by Alpha’s blockchain node.
**ADDING OTHER PARTICIPANTS**

- Alpha's factory A2 and the contracted service company B
  - generate public/private key pairs
  - send the public key to A1, who announces it on the blockchain

- now A2 and B can
  - run their own blockchain nodes
  - add blocks onto the blockchain (they are in the blockchain peer-to-peer network)

- A1 does not know A2 and B's associated private keys and they don't A1's private key
  - now one of the participants can impersonate each other
  - a message on the blockchain signed with B's private key and verified with its public key has to have come from B.
KEYS FOR COMPONENTS

- A2 which manufactures components
  - creates a unique private/public key pair for each component
  - announces the public key on the blockchain, along with the location of the part, which is its factory warehouse.
- the components are assembled to produce the final machine
  - another key pair is generated for the finished product and is also reported
Cheap components: marked with a QR code of the public key for scanning,
- do not participate to the blockchain
- other RFID scanning devices read the key value and submit reports to the blockchain on their behalf.
TAGGING METHODS

more expensive components:

• marked with an active RFID tag with Bluetooth connectivity. T
• can contact nearby network connected devices and submit reports to the blockchain,
A finished machinery may have:

- A fully connected IoT device, with good connectivity to the network.
- Onboard GPS for positioning.
- Possibly a lightweight blockchain node for participating in the generation of new blocks.

An integrated RFID tag reader:

- Scans the complete machine for all its components.
- Detects changes made to those components.
SMART CONTRACTS FOR COMPONENTS

- when a component is logged on the blockchain by its public key announcement, a “smart contract” can be attached to the announcement
- this script will be run by nodes maintaining the blockchain if certain conditions are met.
- such contracts can trigger a service request, decommissioning, or part replacement order to automatically be sent to a service engineering company.
- the complexity of the automation provided depends on the design of the blockchain system.
A condition of the smart contract is met, a node generating the current block sends out a service request to B.

B sends a service engineer to the machinery in order to replace the wornout part.

The service engineer is certified on the blockchain by their own private/public key, stored in their smartphone or their service tablet.

The engineer replaces the part, the tablet and the machine’s central IoT device may send reports back to the blockchain to record the event. The contract is met.

The service tablet may also be a node participating in the blockchain.
ADDITION AND REMOVING PARTICIPANTS

- B decides to revoke a service engineer’s certification: he/she lefts the company...

- B posts
  - a key revocation message on the block chain
  - signed with his/her private key, revoking key b555.

- B can do this because he/she issued and signed the key in the first place.

- all records made with key b555 prior to block 412 remain valid, however, and the records cannot be removed.
company X is the owner of the machine, and sells it to company Y.

- the record of ownership and the location change of the machine is recorded to the blockchain.
- it provides providing a future immutable record of its provenance.

at the end of the life-cycle the equipment is decommissioned,

- this event is recorded on the blockchain, with a signature from the original manufacturer, A.
- this allows future trusted auditing of the application of environmental legislation to be conducted.
CONCLUSIONS

At the end of the process a complete record exists on the blockchain of all:

- participants
- components
- their locations and journeys, who replaced what parts and where and when
- transfer of ownership
- final decommissioning

The records cannot be altered or deleted afterwards

- individual participants can track progress during transit, and review the data after delivery, alteration or transfer
- only those directly concerned with or involved in the design, manufacture, repair or order fulfillment can create relevant records
- there is no central server or central point of failure
CONCLUSIONS

Blockchain 1.0: currency, Bitcoin and altcoins

Blockchain 2.0: smart contracts, Ethereum

Blockchain 3.0: areas in government, health, science, IoT

• work in progress

• several new ideas
Temporary block-withholding

- default mining behaviour: immediately announce a block to the network
- selfish mining: the attacker will mine his blocks privately and release them at the right time
- honest miners waste their computational power and the attacker increases its mining rewards
- Never observed in the Bitcoin network
- Easy to detect: look at the rate of near-simultaneous block announcements
State 0: only a single public chain
State 1: Adversary manages to mine a block. The block is kept *private*. 
State 2: Adversary manages to mine a block. The block is kept private.
State 2: Honest miners find a block

In this situation the private chain is published and the honest miners loose their block.
**State 0:** After releasing the private chain, back to state 0.

New head of the public chain.
State 1: Adversary manages to mine a block. The block is kept private.
State 0': Honest miners and adversary's chain are competing

In this situation release the private block and hope the honest miners will mine on top of it.