CAN'T WE JUST AGREE?

ONDŘEJ CHALOUPKA, VOJTĚCH JURÁNEK

DevConf.CZ 2019
WHAT IS THIS TALK ABOUT

- WHY CONSENSUS ALGORITHM IS AN ESSENTIAL PART IN DISTRIBUTED LEDGER SYSTEMS
- A BRIEF OVERVIEW OF CONSENSUS ALGORITHMS IN DISTRIBUTED LEDGER SYSTEMS
- DIFFERENCES BETWEEN CONSENSUS ALGORITHMS AND HOW THE CHOICE IMPACTS CAPABILITIES OF DISTRIBUTED SYSTEM
Beware: distributed ledgers are not limited only to cryptocurrencies, there are lots of other applications!
CENTRAL AUTHORITY

Initial state

<table>
<thead>
<tr>
<th>ACCOUNT</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$10</td>
</tr>
<tr>
<td>Bob</td>
<td>$0</td>
</tr>
<tr>
<td>Carol</td>
<td>$0</td>
</tr>
</tbody>
</table>

1. Sending $10 to Bob
2. No funds
3. Balance?
4. You have $10
5. Coffee handover
6. Sending $10 to Carol

Centralized ledger

<table>
<thead>
<tr>
<th>ACCOUNT</th>
<th>BALANCE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>$0</td>
</tr>
<tr>
<td>Bob</td>
<td>$10</td>
</tr>
<tr>
<td>Carol</td>
<td>$0</td>
</tr>
</tbody>
</table>
In distributed systems, we need agreement between participants, which transaction is valid.
CONSENSUS

- Agreement between nodes on something, e.g. some value
- Another example: whether to commit a (distributed) transaction to a database
- Hard, network is unreliable. There can be delays or failures in communication
- Consensus has to have two properties: safety and liveness
Ok, there is a terribly wrong framework emerging around consensus protocols. People think that PoW and PoS are consensus protocols, and that they are the only two consensus protocols out there.

This is false. Let me explain.
NON-BYZANTINE

FAULT TOLERANT CONSENSUS

- can withstand failures but not a cheating participant
- trust in all involved parties
- protocols like Paxos (Google Spanner), Raft (etcd), Zab (Zookeeper)
  - leader-based vs. leader-less
RAFT ALGORITHM

Leader based

Leader election

http://thesecretlivesofdata.com/raft/
RAFT ALGORITHM

RAFT CONSENSUS PROTOCOL

RAFT node

\[ x = 8, \text{ ok?} \]

\[ x \leq 9, \text{ ok?} \]

\[ 04, \text{ ok?} \]

RAFT node

RAFT node

RAFT node

RAFT node

commit

Log of agreed values

\[ x = 7, y = 9, z = 3 \]

\[ x = 8 \]

PRIVATE BLOCKCHAINS

- all nodes are under control of a *single organization*
- number of participant is small, they know about each other and *trust each* other
  - Raft in Hyperledger (Sawtooth, on roadmap for Fabric)
- **Permissioned**: nodes needs some permission to join the network
- **Permissionless**: nodes can join the network without any permission (public blockchains)
Byzantine Failure

- Besides delays and failures in communication over network, the situation can be even worse - there can be malicious participants!
- A Byzantine failure is any fault presenting different symptoms to different observers.
- E.g. attempt to double spend money.
BFT ALGORITHMS
BYZANTINE FAULT TOLERANT

- PBFT, Tendermint, Stellar ...

- Agreement can be imaged like a three-phase commit (propose a value, pre-prepare commit, prepare, commit).

FEDERATED DISTRIBUTED LEDGERS

- Still to some extent centralized.

- Usually to some extent permissioned or have to be combined e.g. with PoS (Tendermint).

https://mc.ai/whats-new-in-deep-learning-research-understanding-federated-learning/
SIBYLS ATTACK

- Forging the identity to subverted the result
PROOF-OF-WORK

Block hash: 1
Previous hash: 0
Nonce: ax3F67
---
Transactions
---
Block #1

Block hash: 2
Previous hash: 1
Nonce: WfIY15
---
Transactions
---
Block #2

Block hash: 3
Previous hash: 2
Nonce: r3dH4T
---
Transactions
---
Block #(?)

Miner
SHA-256 from header, changes nounce value

BLOCKCHAIN
BLOCKCHAIN FORK

Block #1
- Block hash: 1
- Previous hash: 0
- Nounce: ax3F67
- Transactions
  - T1
  - T2
  - ...

Block #2
- Block hash: 2
- Previous hash: 1
- Nounce: WflY15
- Transactions
  - T4
  - T5
  - T6

Block #3 (?)
- Block hash: 3
- Previous hash: 2
- Nounce: r3dH4T
- Transactions
  - A
  - B
  - C

Block #3 (?)
- Block hash: 3
- Previous hash: 2
- Nounce: 11lbM
- Transactions
  - Y
  - Z
  - C

Alice Miner
Bob Miner

mined
mined
NAKAMOTO CONSENSUS

- Bitcoin is (almost) Byzantine fault tolerant (BFT) and also resistant to Sybil attack
- Proof-of-Work (PoW) is mechanism how to prevents Sybil attacks
- The truth (agreement) is determined by the longest chain (created by Proof-of-Work) usually called Nakamoto consensus algorithm
- **Probabilistic**: probability of consensus is less than 1
- Proof-of-Work finding a new block is stable to 10 minutes
  - Bitcoin network is essentially synchronous
WHAT ABOUT POS?

- Proof-of-Stake (PoS) is again **not a consensus** algorithm
- Proof-of-Stake **solves Sybil** attack
- **Chain-based** vs. **PBFT-style**
  - **Hybrid PoS**: Proof-of-Work with Proof-of-Stake on top
Consensus algorithms on top of DAG

Blockchain

DAG
- Hashgraph, Avalanche, Tangle...
- Usually are not resistant to Sibyl attack (needs PoS or something else)
Hashgraph

- Gossip about gossip
- Virtual voting

Avalanche

- Gossip protocol
- Metastability

https://www.youtube.com/watch?v=AXrrqtF1Gow
Chaloupka-Juranek almost-consensus algorithm with (usually) positive social side-effects:
PRACTICAL EXERCISE

Chaloupka-Juranek almost-consensus algorithm with (usually) positive social side-effects:

1. Go to [https://sched.co/Jcj3](https://sched.co/Jcj3)
2. Scroll down to Feedback form and click on this icon
3. Talk to 5 randomly chosen people on the corridor during the conference
4. Introduce yourselves, explain them the algorithm and ask them to execute points 1.-4. (and continue chatting with the person)

Expected result: if most of the DevConf attendees are not Byzantine, we should win best talk competition!
TAKE-OFFS

- Consensus protocol is a crucial part of any distributed ledger.

- Choice of consensus protocol influences heavily many characteristic of distributed ledger (including performance and security).

- There are several types of distributed ledgers, several families of consensus algorithms and not every consensus algorithm is suitable for every distributed ledger.
QUESTIONS
S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System
D. Ongaro, J. Ousterhout, In Search of an Understandable Consensus Algorithm
M. Castro, B Liskov, Practical Byzantine Fault Tolerance
D. Mazieres, The Stellar Consensus Protocol
The latest gossip on BFT consensus - Tendermint consensus algorithm
L. Baird, The Swirlds Hashgraph Consensus Algorithm
Team Rocket, Snowflake to Avalanche
Thank you for your attention!
BACKUP SLIDES
BITCOIN 51% ATTACK
POS: NOTHING AT STAKE

Vote on neither EV = 0

Vote on A EV = 0.9

Vote on B EV = 0.1

Vote on both EV = 0.1 + 0.9 = 1

https://github.com/ethereum/wiki/wiki/Proof-of-Stake-FAQs
Proof-of-X

There are many:

- Proof of Capacity
- Proof of Elapsed Time (PoET)
- Proof of Authority
- Proof of Activity
- Proof of Burn
- Proof of Weight
- ...

Beware: Proof-of-X doesn’t mean it’s similar to PoW, actually in many cases it’s quite different (e.g. centralized).
<table>
<thead>
<tr>
<th>TAXONOMY TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Proof-of-Work</strong></td>
</tr>
<tr>
<td><strong>Proof-of-Stake</strong></td>
</tr>
<tr>
<td><strong>Delegated</strong></td>
</tr>
<tr>
<td><strong>PoS</strong></td>
</tr>
<tr>
<td><strong>Raft</strong></td>
</tr>
<tr>
<td><strong>Tendermint</strong></td>
</tr>
<tr>
<td><strong>Swirlds</strong></td>
</tr>
<tr>
<td><strong>Avalanche</strong></td>
</tr>
</tbody>
</table>