Blockchain, Consensus, Nothing at Stake

Why we need blockchains

Replicated State Machine

A certain (possibly changing) number of nodes realize Replicated State Machine

$$S + \Delta = S'$$

Nodes can be faulty: Fail-stops, Byzantine failures.

In general RSM should have 2F+1 nodes to tolerate F faulty nodes, 3F+1 (2F+1 with authenticated communication) in case of malicious Byzantine attacks.

Impossibility results.

CAP theorem

CAP - Consistency, Availability, Partition tolerance. Can't have all three!

If a network partition takes place you can't have availability and consistency.

FLP impossibility (Fisher, Lynch, Patterson)

No *deterministic* algorithm can guarantee consensus in asynchronous setting.

Dwork, Lynch, Stockmeyer 1984 Bounds on fault tolerance

Partially synchronous networks can tolerate up to ⅓ byzantine failures.

Synchronous model can tolerate up to 100% failures (although more than 50% faulty nodes makes things more complicated)

Pre-blockchain algos, Paxos, Leader election

Consensus research was very much alive before Bitcoin.

Paxos protocol family is based on voting (many rounds of it). Leader is elected, and he proposes the consensus value to vote on. The algorithm may stall, but it guarantees consistency (if nodes agree, the agree on the same value).

Paxos and PBFT is extensively used in practice. What is the problem with it and why we need blockchains?

1. Paxos scales as ~N^2, and becomes impractical when we have many validators.
2. Sybil attacks are cheap, attacker can spawn as many Byzantine nodes as he wishes cheaply.

In a closed, permssioned system we can do without.

Enter the Blockchain

Leader election in Proof of Work blockchains (Bitcoin, Ethereum) takes place through solving a cryptographic puzzle H(y|x) < Target, node that solves it obtains a right to generate a block (serialize transactions and change state in RSM). y depends on past tx history, Target can be tweaked to obtain constant processing speed.

Block contains its ID, ID of the previous block, solution to the cryptographic puzzle (based on the hash of the previous block), “hash” of the current block. Miner obtains a reward for generating a block.

Blocks are stored in append-only log called Blockchain. The longest blockchain becomes the “official” view of the system history. Consistency guarantees are strong, “eventual consistency” claims are more of a misunderstanding.

Blockchain-based consensus scales as N, not N^2. Besides, they make Sybil attacks costly.

Bitcoin can be put into synchronous network model, and should have ~ 50% fault tolerance (a little less due to latency). When we talk about Bitcoin we talk about Byzantine computing power, not Byzantine nodes per se.

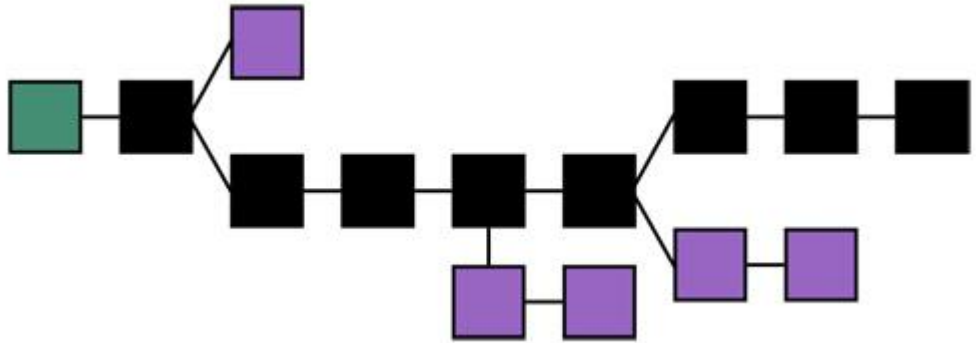
Selfish Mining Discovery (Ittay Eyal, Gun Sirer) - economical incentives for miners to join Byzantine hash power. 25% limit on total Byzantine hashes. Is compatible with results from Byzantine Consensus Theory.

Blockchain throughput, latency and forks.

Bitcoin: 1 MB blocks, 10 min between blocks, ~ 1 tx per second.

Why not decrease the time between blocks or increase block size?

LATENCY kills: too many forks, it's hard to select one, nodes waste resources validating blocks on forks and mining on "wrong" branches



Ghost protocol, POW in Ethereum

Average time for new block propagation in Bitcoin network is 12.6 sec. Average time between block is 600 seconds.

Stale rate, the number of *orphaned,* discarded blocks is 1.97% in Bitcoin network. What if we try to reduce block time to 12.6 sec? Stale rate is 50%, half of computing power is wasted, network becomes weaker against attacks. Besides, mining pools have substantial advantage in this set-up, since they find about the new blocks faster.

To account for stale blocks we have to include them into calculation of the weight of the main chain. Zohar, Sompolinsky (2013) GHOST protocol.

You have also pay a part of the mining reward to the miners of stale blocks (“uncles” in Ethereum), otherwise we still have the mining pools issue.

Ethereum: for every uncle U included in B the miner of B gets 3.125% of the reward, the miner of U gets 93.75% of the standard reward.

ASIC's, Pools, Centralization, Waste of Resource

Total cost of Bitcoin mining in terms of electricity expenditures is around 0.5 billion dollars per year. Is it worth it?

Bitcoin (and Ethereum) can't be mined on home CPU's. Specialized hardware (ASIC's) dominates the Bitcoin market. Chinese mining pools effectively control Bitcoin network.

Ethereum uses ASIC resistant, memory-hard mining algorithms, requiring lots of RAM.

Makes sense to looks for cheaper alternatives, not depending on specialized hardware and expensive computations.

Proof of stake - cheap alternative.

Proof of stake, mining with your stake.

We use coins instead of hardware to mine. Based on the previous block and some randomness a miner of the next block is selected, with the probability proportional to her stake in the system (balance of coins). First POS system Peercoin, 2012. First POS systems using balances instead of UTX (outputs) - NXT. The biggest existing system is Waves. “ Big” Ethereum is planning to move to Casper POS.

Leader election (typical):

HASH(prevhash + address + timestamp) <= balance / diff

POS is more of a classical Byzantine fault tolerance problem than POW. It might be more formalizable in the long run then POW.

Also some of the algos don’t use validator selection and fall back to consensus algo’s similar to Paxos (Tendermint)

What's at stake? Attacks on POS, solutions.

Nothing at stake problem: POS is cheap, so economically it makes sense to mine on top of all the forks a node has access to. In POW that would be too costly.

How current system solve this: 1. Default implementation storing one chain only. 2 Checkpointing, node can't roll back to another chain after a certain amount of blocks (also fends off history attack)

Weak subjectivity problem is Nothing at Stake problem too: a NEW node can't tell a fake chain from a true one, needs a trusted source. POS blockchain becomes subjective, contrasted to "objective" POW blockchain.

Grinding attack: there's no external random in POS systems, new miner is elected quasi-deterministically. It is possible to predict the miner succession and obtain the control of the network.

Some randomization is needed to counter this. Provably secure Proof-of-stake: Kiayias et al. (2016). Also hintinging that Nothing at Stake problem is not "practical"

Another way to counter Nothing-at-stake is Slasher type algos.

Slasher and Casper: punishing the bad player

Make POS great again: let miners bet their funds, bad miners lose funds.

Iddo Bentov (2015): If miner is caught mining on two chains she loses her security deposit

Vlad Zamfir, Vitalik Buterin (2014) Casper POS proposal: Consensus by Bet

Validators bet on each block. If their bet is different from the consensus they lose their deposit. Something at stake! POS modelled on POW (and classical byzantine fault tolerance research)

Economic finality: attempts to justify the result from Byzantine research about 100% fault tolerance of synchronous Byzantine tolerant protocols.

Conclusion

- Blockchains are interesting for closed, permissioned networks also, although not so much as for open systems
- Future of blockchain tech is most probably POS-like consensus algo’s
- Next step in development of open blockchains is scalability enhancements (throughput, blockchain pruning, system state storage)

sasha@wavesplatform.com facebook.com/sasha35625 twitter.com/sasha35625

WAVESPLATFORM.COM