GRANDPA finality in Polkadot

Fedor Sakharov
Software developer @ Parity Technologies Ltd.

fedor@parity.io | @montekki

Agenda

- Pre-bitcoin Era (BFT consensus)
- Nakamoto Consensus
- Naive PoS systems
- First proposals on finality gadgets (Slasher, Casper)
- GRANDPA finality gadgets in Substrate and Polkadot

BFT consensus (1982)

A system of M communicating processes needs to eventually agree on some decision that would be same for everyone.

The solution provides us with the algorithm that guarantees consensus for the system as long as $> \frac{2}{3}$ of actors are honest.

BFT consensus (1982)

The solution is

- Final (it is not a subject to re-adjustments in the future)

BFT consensus (1982)

The solution is

- Final (it is not a subject to re-adjustments in the future)

But has a major drawback:

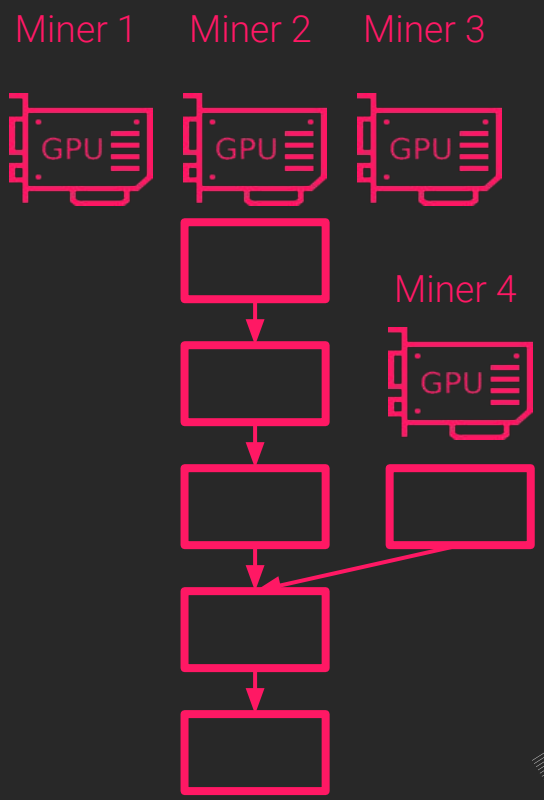
Kind of slow, to find a solution we need to send O(n^m) messages where

- n - number of actors
- m - number of malicious actors

parity

Nakamoto Consensus (2009)

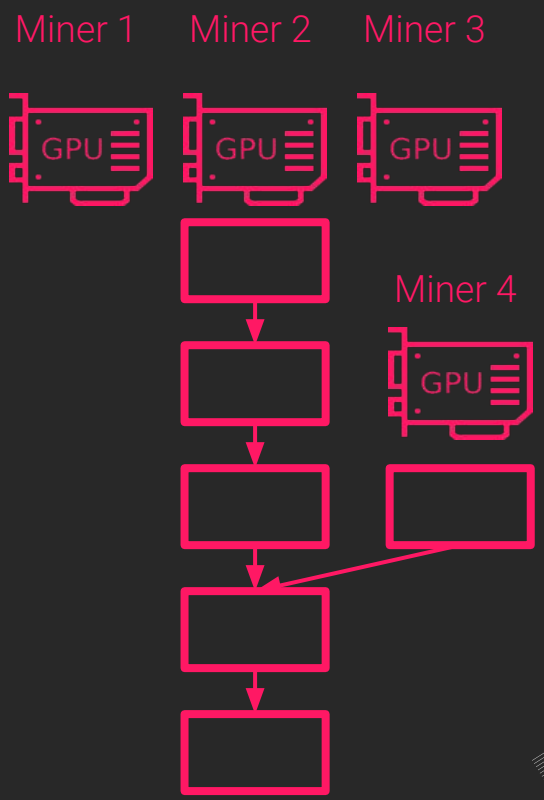
- PoW based block production
- Always follow the longest chain
- The longest chain will be the one with the biggest cumulative mining power



parity

Nakamoto Consensus (2009)

- The probability that you will mine the next block depends on your share in the network's computing power
- The more computing resources back some fork, the faster it will grow

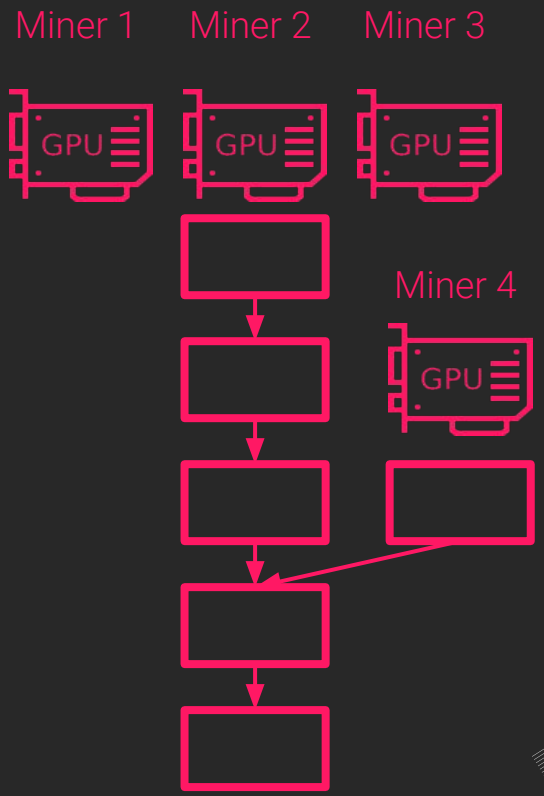


parity

Nakamoto Consensus (2009)

Nakamoto consensus is:

- Eventual
- Probabilistic (unlike BFT consensus you are never 100% sure that agreement on state is final)
- Decentralized (noone is restricted from producing blocks)

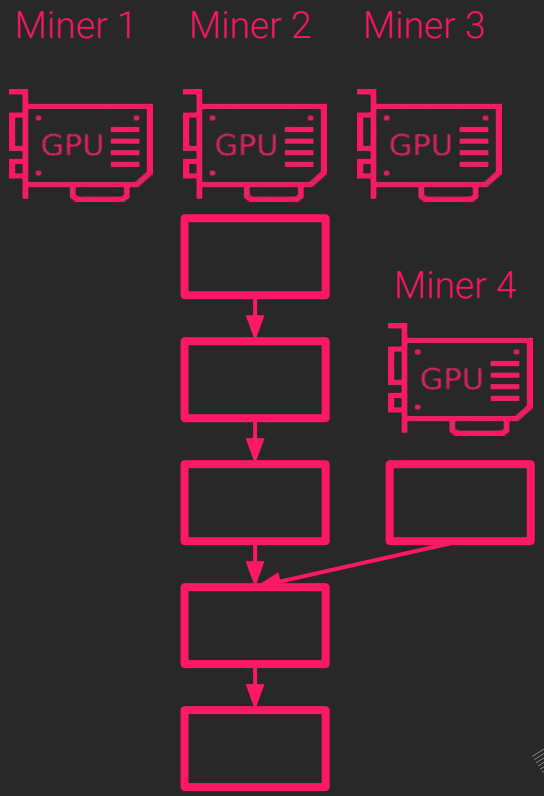


parity

Nakamoto Consensus (2009)

Has several drawbacks:

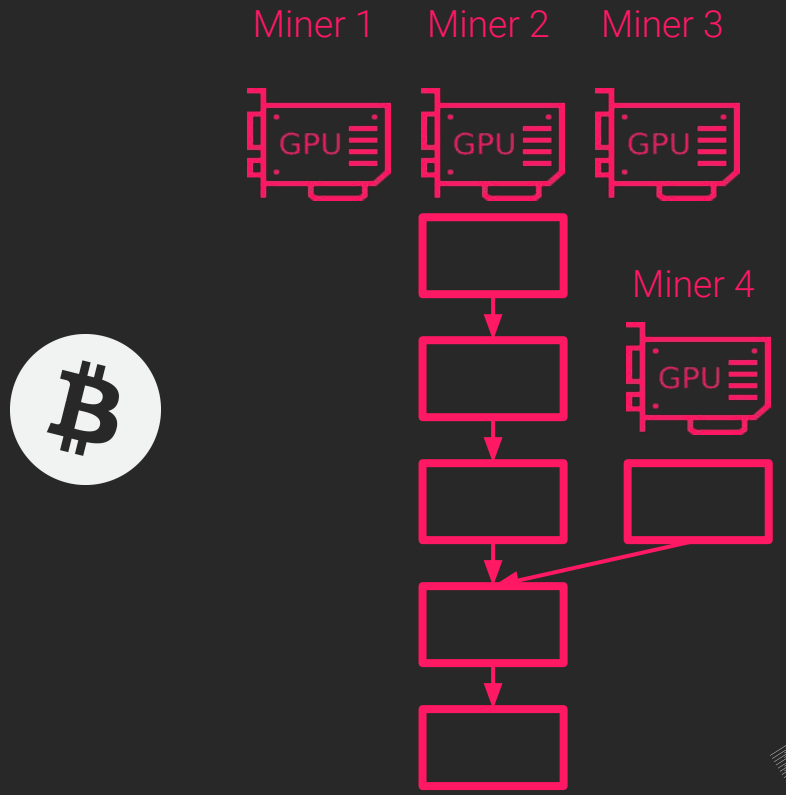
1. Environmental concerns
2. Probabilistic finality
3. Security
4. ....



parity

Nakamoto Consensus (2009)

The amount of computing power accumulated in mining pools of existing blockchains and in cloud datacenters makes it impossible to create new secure blockchains with PoW consensus*



* https://www.crypto51.app/

parity

Cost for a 51% attack

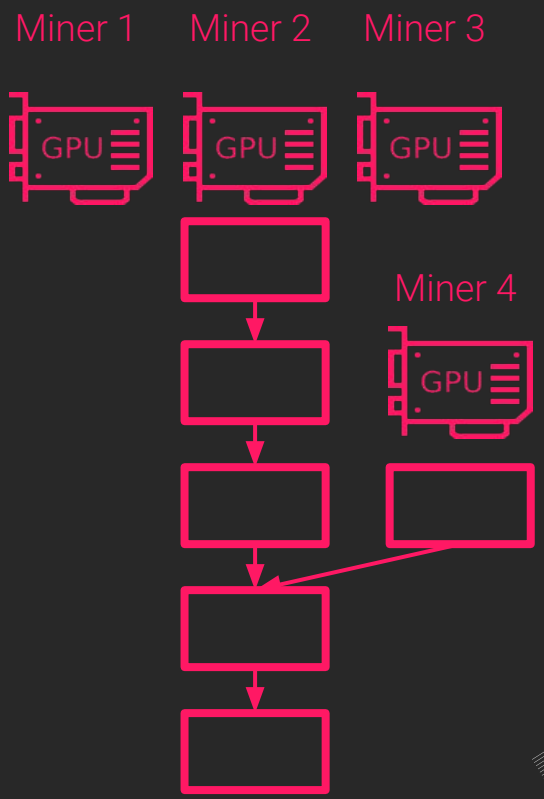
Market cap	$1.22 M
Mining algorithm	X11Gost
Network hash rate	156 GH/s
Nicehash cost	0.0001 BTC / GH / day
Nicehash cost / hr	$0.02 / GH / hour
Estimated cost of 1 hour 51% attack	*$3*
Nicehash capacity	7 GH/s
Nicehash percentage of network	4%

51% attack cost example

Chain-based PoS (~2012)

What if we try to skip the computationally heavy part and:

- Set the probability that a miner mines the next block proportional to his share in all emitted (or staked) currency
- Design some block producing function to suffice this goal

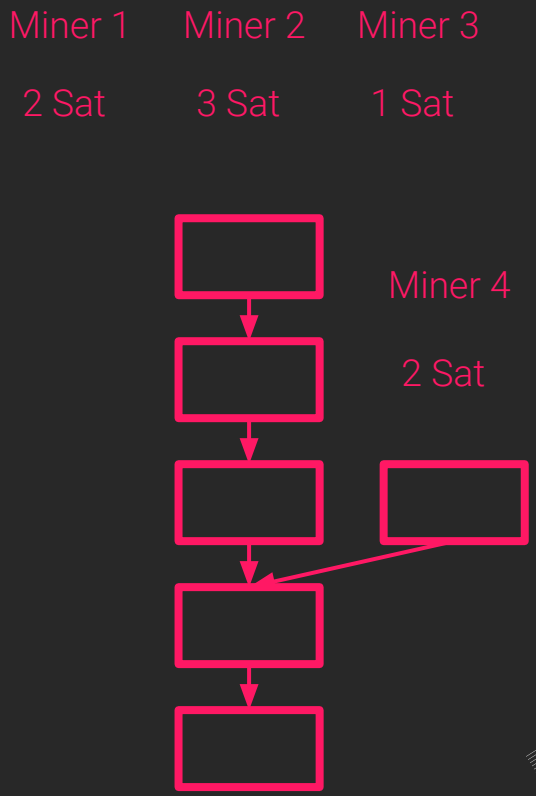


parity

Chain-based PoS (~2012)

What if we try to skip the computationally heavy part and:

- Set the probability that a miner mines the next block proportional to his share in all emitted (or staked) currency
- Design some block producing function to suffice this goal



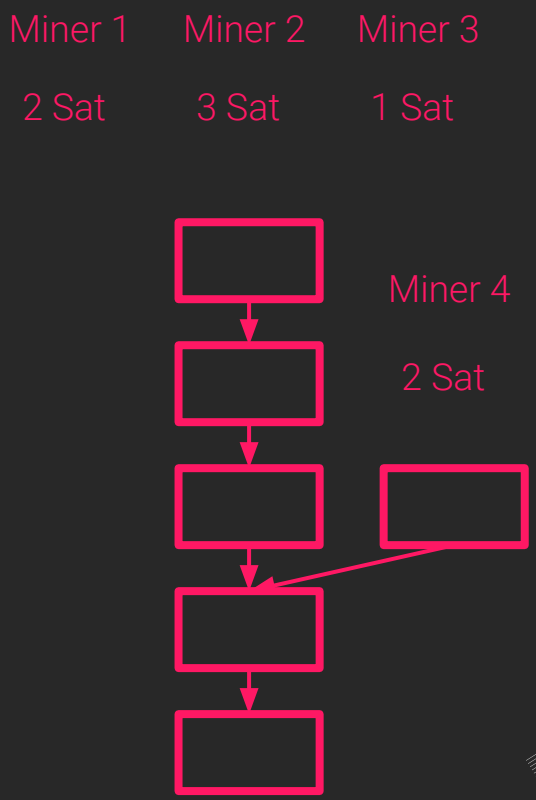
parity

Chain-based PoS (~2012)

Looks like we have a problem here:

You can not mine with the same hardware on two different forks

You can mine with the same stake on two different forks, since your stake is internal to the state of the chain



parity

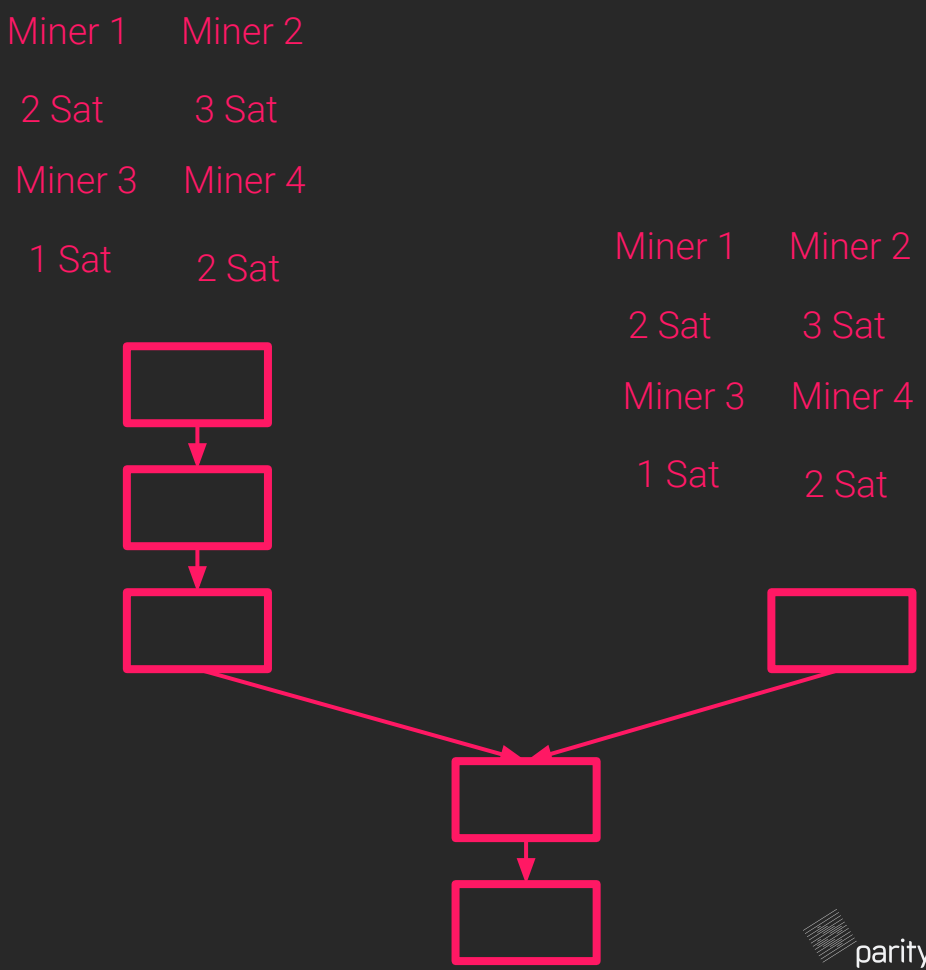
Chain-based PoS (~2012)

Looks like we have a problem here:

You can not mine with the same hardware on two different forks

You can mine with the same stake on two different forks, since your stake is internal to the state of the chain.

The Nothing-at-stake problem.



parity

Slasher: A punitive PoS (2014)

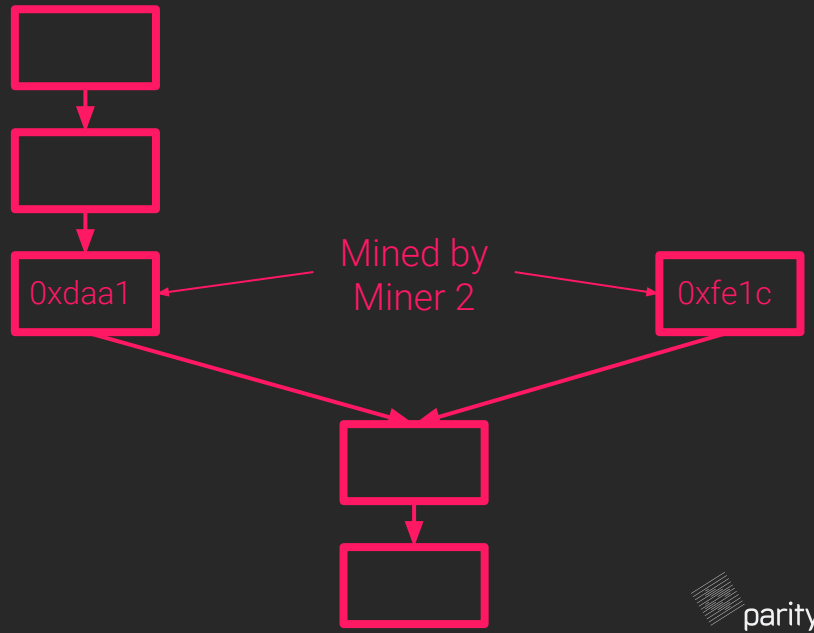
Does nothing-at-stake problem make PoS a dead-end?

If we continue treating blockchain as a linked-list and different forks have no clue about each other that is probably the case.

But what if we start treating blockchains as trees?

Slasher: A punitive PoS (2014)

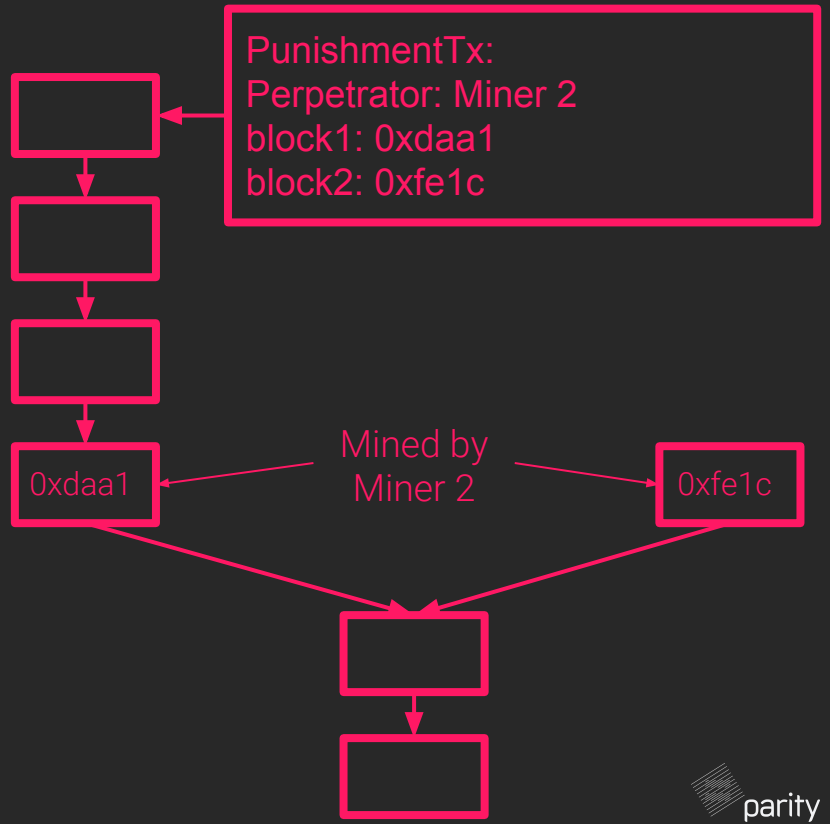
- Now we are aware of all forks in the chain
- Mine a block at height K, get the reward at height K+1000
- Mine two blocks on different forks at height K => anyone can submit the proof of this to the blockchain and you lose rewards.



parity

Slasher: A punitive PoS (2014)

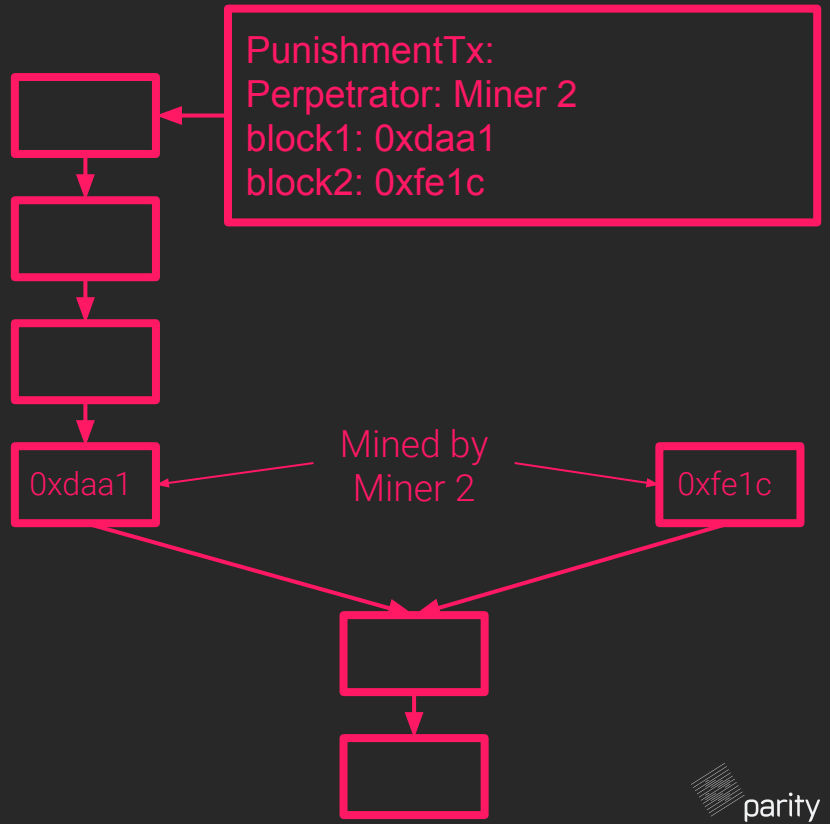
- Now we are aware of all forks in the chain
- Mine a block at height K, get the reward at height K+1000
- Mine two blocks on different forks at height K => anyone can submit the proof of this to the blockchain and you lose rewards.



parity

Slasher: A punitive PoS (2014)

- Now we are aware of all forks in the chain
- Mine a block at height K, get the reward at height K+1000
- Mine two blocks on different forks at height K => anyone can submit the proof of this to the blockchain and you lose rewards.
- Looks like this idea didn't work



parity

PoS research 2014-2017

- Tendermint (BFT-based)
- PBFT, HBBFT(BFT-based)
- Ouroboros (BFT-based)
- Casper FFG and other Casper flavours (2017) (Hybrid Consensus)

Hybrid Consensus Model

So, up to this point we have seen two schools of thought:

- chain-based PoS
- BFT PoS

Hybrid Consensus Model

So, up to this point we have seen two schools of thought:

- chain-based PoS
- BFT PoS

Why not have the best of both worlds and design a hybrid consensus model with:

- Faster probabilistic block-producing mechanism
- Slower but BFT-deterministic finality overlay

parity

Economic Finality

A block, state or any constraint on the set of admissible histories can be considered finalized if it can be shown that if any incompatible block, state or constraint is also finalized (eg. two different blocks at the same height) then there exists evidence that can be used to penalize the parties at fault by some amount $X. This value X is called the cryptoeconomic security margin of the finality mechanism.

parity

Consensus in Polkadot

A hybrid consensus model that separates block production from finality:

- Fast and probabilistically synchronously safe block production logic (traditional PoS, PoW, PoA block producing)
- Asynchronously safe finality gadget providing accountable safety (BFT-like ⅔ votes consensus)

parity

GRANDPA finality gadget (2018)

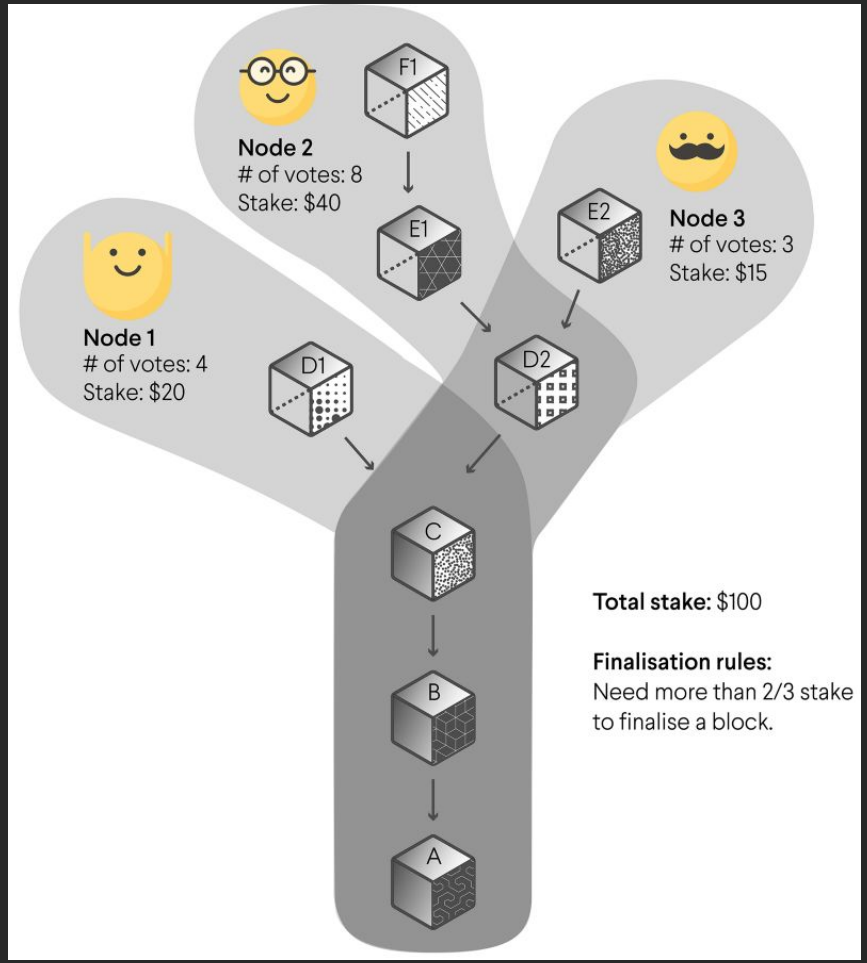
GHOST-based Recursive ANcestor Deriving Prefix Agreement

- Under good network conditions blocks are finalized nearly instantly
- In case of partitions can finalize millions of blocks when partition is resolved
- Can finalize blocks regardless of # of blocks passed after the last finalized block

GRANDPA

Rather than vote on single blocks that considered valid, vote on the highest block considered valid. (the GHOST part in the name)

As soon a block appears with > ⅔ supermajority votes, it is finalized



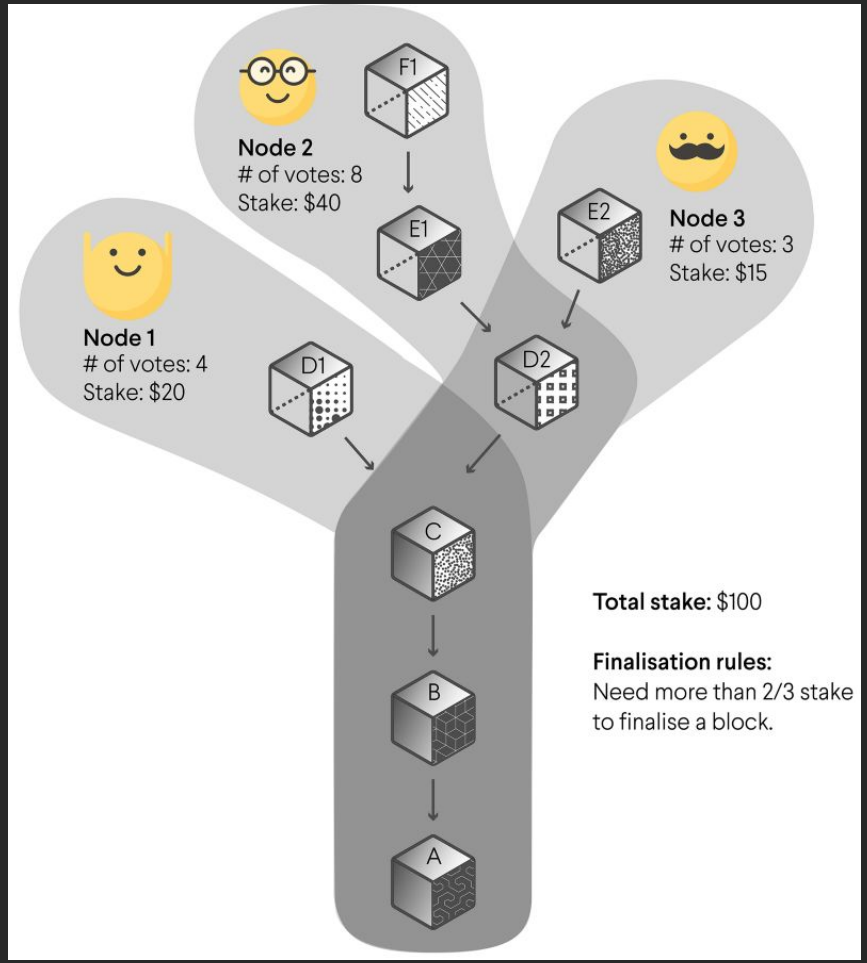
parity

GRANDPA

All the voting and finalizing logic is done off-chain

Participants are staked on-chain

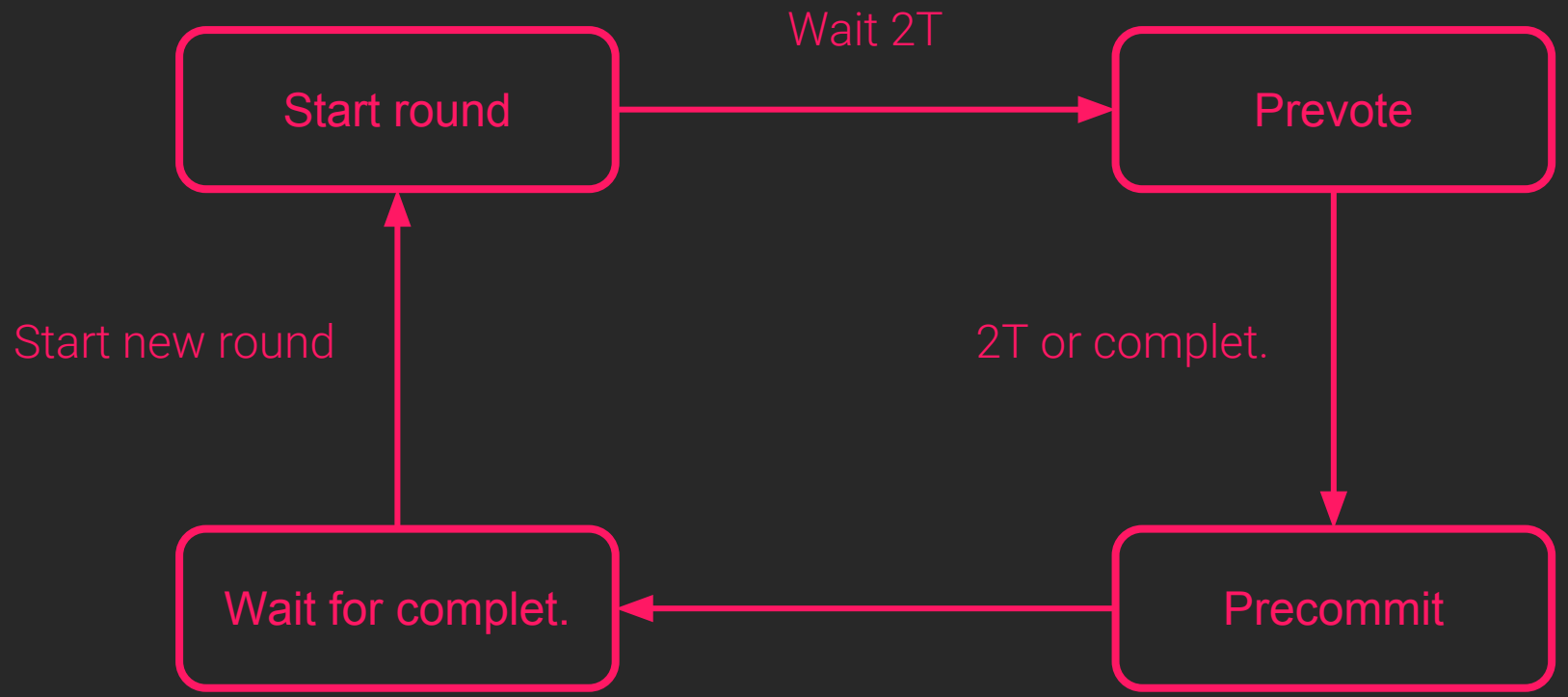
Violations are detected off-chain but penalties taken on-chain



parity

GRANDPA

Voters are FSMs:



parity

Future Work

1. Detect equivocations
2. Detect double-votes
3. Detect voters going offline for long periods offline
4. Detect any other fraudulent behaviour

…. and start punishing people.

Summing it up

- BFT consensus too expensive to get for every state transition
- PoW is not usable
- Naive approaches to PoS don’t seem to work
- We are building a hybrid model with block production (PoA, PoS) and GRANDPA finality overlay

parity

Questions?

fedor@parity.io

 montekki

We’re hiring!

parity.io/jobs

parity