Week 3 EPFsg Consensus Layer Notes

Guest speaker - Alex Stokes

- Alex Stokes - Researcher at Ethereum Foundation, Worked on multiple topics/ projects incl. MEV, PBS, Reth etc.

Summary notes

- Edited by Chloe Zhu
- Online version: https://ab9jvcjkej.feishu.cn/docx/X7Ard9UlPoj2lmxKMvucfNubnTb

Overview

Blockchain enables a way to create digital scarcity

- Why we care about blockchain in the first place?
  - Blockchain creates a way to manufacture digital scarcity, which is difficult to achieve previously.
  - Thus, this property of digital scarcity can be used to emulate different sorts of physical assets in the digital realm, eg. money, tokens, property rights etc.
- The way to make digital scarcity: An example to create a digital money with scarcity
  - Target: Create a digital money with scarcity
  - Unit: Coins
  - Scarcity: There are only ever N coins at a time. And a user can't spend more coins than they have.

	Single trusted operator case	Distributed nodes case
Operator	A single operator runs a webserver that implements this money protocol	A distributed nodes network that implements the protocol N nodes would compute some output over the same inputs
Implication	Users need to trust this webserver operator to ensure no double-spends	Consensus via "state machine replication" among different nodes, witihout a trusted 3rd party

Consequence	• **Hard to have a trusted operator due to different scenarios** eg: • Bug in money protocol • Active attack in protocol server • Dishonest operator • Incentive to attack & abuse the protocol due to scarcity	• Nodes in the system would duplicate the same input log to get the same output • Every node should agree on the output and honest nodes must end up with the same output • **As the number of nodes increase, the system become harder to attack**
Result	Need to remove the single trusted operator and minimize trust	At any point of time, even if there are some nodes with faulty output, as long as there is a majority of nodes that have the same view on the output state, the protocol can reach consensus and continue its operation

Distributed networks deal with Byzantine fault tolerance (BFT)

Why do we need Byzantine fault tolerance?

- If more nodes lead to higher security, then we would want to have a greater number of nodes. However, in an open & distributed system, nodes might have issues (eg. Hardware failure, missed messages, bugs, attacks etc.) which leads to faulty output different from the consensus.
- Thus, we need to have a certain fault tolerance to make the system continue operating

What's Byzantine fault tolerance (BFT)?

- Byzantine fault tolerance (BFT) is the property of a system that is able to resist the class of failures derived from the Byzantine Generals' Problem. This means that **a BFT system is able to continue operating even if some of the nodes fail or act maliciously**.

Two-phase commit (2PC)

- 1st Prepare Phase: One node will ask other nodes whether they can commit the proposed tx.
  - In the case of Ethereun, a node with updated state incorporating new txs will broadcast the updated state to all the other nodes. The other nodes will acknowledge the receipt. And when it reaches the BFT (2/3 super majority), then the preparation phase is done.
- 2nd Commit Phase: The node will command other nodes to either commit or abort the proposed tx.
  - In the case of Ethereun, when 2/3 super majority is reached, nodes within the system will update to the new state.

Practical Byzantine fault tolerance (PBFT)

- PBFT consensus algorithm **allows a distributed system to reach a consensus even when a small amount of nodes demonstrate malicious behavior**.
- Issue of PBFT
  - **It only works well with small consensus group size** due to the cumbersome amount of communication that is required between the nodes (eg. If there are 10 nodes in the system, it needs 10^10 times of message passing to exchange the updated state and reach consensus.)
  - **It also susceptible to sybil attack** where a single party can create or manipulate a large of nodes in the network, thus compromising the network

Q&A

- Do PBFT protocols have an upper bound of nodes? If yes, what is that, practically speaking?
  - The number of nodes doesn't have a hard cap, but there is a trade-off between high throughput and low latency.

Bitcoin solves the BFT with PoW approach

Bitcoin is considered the 1st solution to solve the Byzantine General problem

- The system can scale to unlimited node count
- Open & permissionless participation
- Use PoW mechanism to reach consensus

Bitcoin consensus mechanism

- Bitcoin's state machine replication
  - Input: Tx (organized in blocks) to spend bitcoin
  - Output: Current state of bitcoin ledger
- Use cryptography to reduce possible state space
  - Digital signature: Use cryptography to verify the authenticity of a tx
  - Parent hash: Every new block must include the hash of the previous block
- Use PoW to implement consensus
  - Mining difficulty adjustment:
    - Bitcoin has the concept of **mining difficulty,** which refers to how hard it is for miners to solve the math equation and find the hash for the next block.

        - **The mining difficulty is determined by the network’ s total computational power.** If more miners join the network and the hash rate increases, the mining difficulty will increase and vice versa.
    - Implications of mining difficult adjustment
        - Sybil protection: A new block must perform a certain amount of work to be considered valid
        - Consensus algorithm:: The way for nodes to find the head of the chain is to sum up the mining difficulty done by each block and pick the chain with the largest total difficulty
- Issue its native currency BTC for incentive
    - Provide rewards to incentive work to the current single chain with the most work

Ethereum moves from PoW to PoS

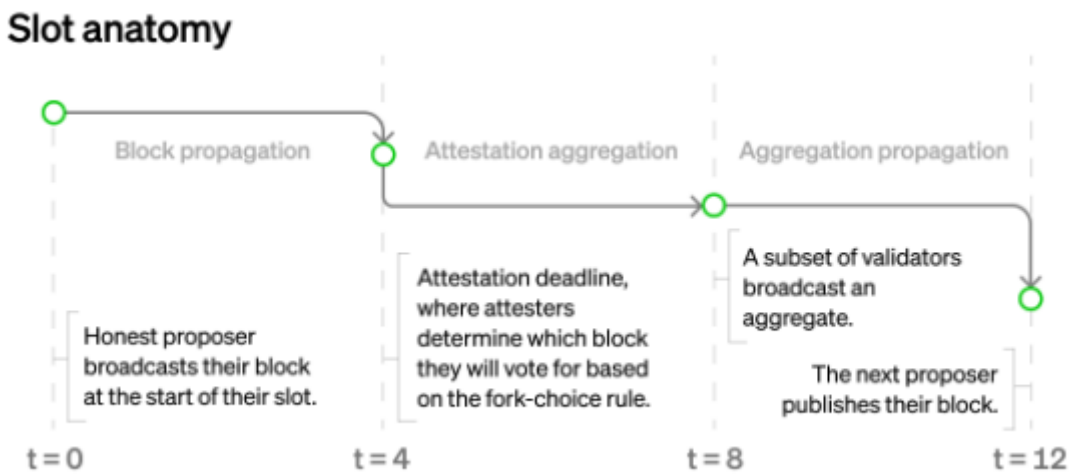
Essence of PoW -> PoS

- Switch from exogenous signal for Sybil protection (work) to endogenous signal (stake) in the system
- Consideration behind
    - Energy usage concerns with PoW
    - Incentive concerns with PoW: Compared to PoW, PoS's in-protocol signal allows for both penalties and rewards

Ethereum consensus mechanism

- Valiators: In-protocol consensus actors
    - Become a consensus validator
        - User need to lock up 32 ETH and send them to the deposit contract in the EVM, which would be seen in the CL level.
    - Responsbility
        - Make attestation: i.e. Validator makes cryptographic signature over the state of the chain
        - Different types of attestations
            - LMD GHOST vote: Validator attests to the beacon chain head
            - Casper FFG vote: Validator attests to the checkpoint in its current epoch
- Key concepts and terminology
    - **Slot**
        - Every 12 seconds there will be a new slot, and every slot will have a block

- Within the slot, it is divided into 3 phases, each consuming 4 seconds. And the most critical moment in a slot is the attestation deadline at t=4. (Paradigm blog)



- **Epoch**
  - Each epoch has 32 slots. The reason behind creating the epoch is to reduce the frequency of consensus processing, so that it doesn't need to happen in every slot
  - Heavier processing is usually done at the epoch boundary, incl. slashing, rewards info etc.
    - Epoch boundary blocks (EBB) can also be considered synonymous with checkpoints. (The Beacon Chain Ethereum 2.0 explainer)
- **Committee**
  - Validators within the network will be randomly shuffled under different committees.
  - Each validator will make one attestation per epoch. The exact slot the validator is assigned is determined by the protocol through RANDAO.
- **Finality**
  - Finality means that a tx is part of a block that can't change.
  - Justification: When an epoch ends, if its checkpoint has gathered a 2/3 supermajority, the checkpoint gets justified.
  - Finality: When a checkpoint is justified, the previous checkpoint that is already justified becomes finalized.

Q&A

- Is there any significance of choosing 32 ETH?
  - It's a choice under tradeoff. i.e. If the threshold is too low, there will be too many validators, thus it may take too long to reach consensus. And if the threshold is too high, there will be too few then the system becomes less secure.

        - Originally, the core dev was thinking of 1000 ETH as the threshold, then Justin Drake suggested the usage of BLS signature technology, which lowered the minimum capital required to 32 ETH.
        - Justin Drake's research on pragmatic signature aggregation with BLS: https://ethresear.ch/t/pragmatic-signature-aggregation-with-bls/2105
    - Why there is 12s in a slot?
        - The 12s is kind of arbitrary, which is inspired by the PoW time (14s on average).
    - Question about randomess of RANDAO and how validators shuffled randomly?
        - The randomness is achieve using the algorithm RANDAO that mixes a hash from the block proposer with a seed that gets updated every block. This value is used to select a specific validator from the total validator set. **The validator selection is fixed two epochs in advance** as a way to protect against certain kinds of seed manipulation.
        - Although validators add to RANDAO in each slot, the global RANDAO value is only updated once per epoch. (Block proposal)
        - Github link: https://github.com/randao/randao
    - Gasper in the context of finality and finding the canonical chain?
        - Gasper is the combination of Casper-FFG and LMD-GHOST fork choice algorithm (Gasper)
            - Casper is the mechanism that upgrades certain to finalized, so that new entrants can be confident that they are syncing the canonical chain.
            - LMD-GHOST is the fork choice algorithm that uses accumulated votes to ensure that nodes can easily select the correct one when forks arise in the blockchain.
    - Brief explanation of PBS (proposer-builder separation)
        - PBS (proposer-builder separation):
            - MEV issue: MEV refers to validators max their profit by favourably ordering txs. Maximizing MEV requires sophisticated know-how and hardware & software, which could potentially lead to centralization as institutional operators usually outperform individual validators.
            - PBS: Allow block proposer to outsource block construction, so that validators can continue running on consumer-grade hardware without missing out MEV exposed
            - Research link: https://ethresear.ch/t/why-enshrine-proposer-builder-separation-a-viable-path-to-epbs/15710
            - Roadmap blog: https://ethereum.org/en/roadmap/pbs/
        - Some of the important things on the roadmap of Ethereum

- SSF (single slot finality): Aim to get finality in a single slot
  - Vitalik post on SSF: https://notes.ethereum.org/@vbuterin/single_slot_finality
  - Roadmap blog: https://ethereum.org/en/roadmap/single-slot-finality/
- SSLE (single secret leader election): Aim to have proposer selection in secret
  - Research link: https://ethresear.ch/t/simplified-ssle/12315
  - Roadmap blog: https://ethereum.org/en/roadmap/secret-leader-election/
- Max EB (max effective balance): Aim to increase the effective balance of Ethereum validators at 32 ETH
  - Research link: https://ethresear.ch/t/increase-the-max-effective-balance-a-modest-proposal/15801