lab10 - Research #61

Research dfinity / BNS

05.03.2017 21:55 - didi

Status:	In Progress	
Priority:	Normal	
Assignee:		

Description

Main point: "Everything depends on randomness!"

Proposal instead of *code is law*: Blockchain Nervous System (BNS).
Dfinity (made by string lab) aims to be a *sister network* to Ethereum, focused on scalability.
It wraps the Ethereum client software in a proxy which monitors and interacts with a BNS contract.
This contract has nearly unlimited power and can do most changes a hardfork could do, e.g. revert actions (think The DAO), change gas costs (think DOS attack) etc.
Voting rights for the BNS lie with Dfinity token holders.
It's basically human controlled governance built into the Blockchain, explicitly rejecting the idea of *code is law*.
There's supposed to be a *constitution*, which can itself be updated via BNS voting.
The voting process is modeled like a brain - a neuronal network. Synapses are reflected by voters creating preference lists of leaders by topic. If a voter doesn't manually vote for/against a proposal in a given timeframe, the vote of leader from the list ist followed.
Liquid democracy style.
In essence, it's a Blockchain governed with a liquid democracy style system, with votes being weighted by economic stake.
ICO (seed funding) has raised 3,9 CHF worth of funds (mostly ETH). The first release (*Copper*) is announced for the upcoming weeks.

They coined the term crypto:3, seems inspired by Ethereum's web3.

Technical differences to Ethereum:
Every process has a *mining identity*. Created by making a deposit. When leaving, the deposit is returned after some delay.
Organized into random groups. Each process belongs to several groups. Thus groups intersect.
When processes join or leave (implemented as blockchain transaction), the protocol updates the group config.
Group gets a public key. Can sign messages via threshold signature (e.g. >= 201 out of 400 members). Verifiable secret sharing.
Threshold signature scheme [BLS](https://en.wikipedia.org/wiki/Boneh–Lynn–Shacham) has the property: no matter which set of the group contributed to the signature, the signature is always the same.
Nodes of the group broadcast their individual signature. Listening nodes just need to collect min 51% of such broadcast signature shares in order to calculate the group signature. It won't matter which set of signature shares they collect and it won't matter if some shares arrive late or never (e.g. because of bad connectivity).
The signature is used as a random number which selects the next group to forge a block (signature modulo number of groups). That results in the group selection process to be deterministic (although unpredictable), allowing instant finality.
If for some reason a group isn't able to produce a signature (e.g. 51% of nodes not reachable), the blockchain stops. They say that in such a case where obviously something is going wrong it's preferable to have it auto-pause instead of forking like other blockchains would.
This is not the same as voting. A group can either produce the predetermined signature or not, it can't produce a false one.
Per block max. 34kB need to be broadcast (400 x 86 bytes of signatures).
Probability of signature creation failing very small even with considerable numbers of bad nodes (e.g. 30%). See slide. The group size results in much higher probabilities for a *good* result then with designs where a single node is selected randomly.
The guy thinks there's a lot of fallacies involved in attempts to design randomness generating algos.

Threshold Relay Blockchain
Probabilistic Slot Protocol (PSP).
Randomness selects priority list block forgers. This is a list of all processes ordered by priority. Every block has it's own list (determined by the random signature).
Processes with high priority forge and broadcast a block. Nodes relay such blocks only if they haven't seen a block from a process with higher priority before.
The *highest scoring* chain wins.

Threshold Timestamping is how nodes of the current group sign blocks. As long as they keep getting blocks from processes with higher priority then they have seen before, they will sign and broadcast.

According to the guy that leads to instant (6s) finality (*overwhelming probability*). I couldn't yet get why that should be the case.

Light client friendly.

Comparison with Ethereum: guy says currently 50% of Ethereum blocks are empty (selfish mining).
Scale up: First (Copper) release expected to have 25-50x gas limit of Ethereum.

Scale out
Separating concerns, 3 layer architecture:

- consensus (Threshold relay chain)
- validation (scalable *validation tree*, composed *validation towers*). "Does for validation what merkle does for data"
- storage: state in shards. Passed to validation tree.

Each process has 3 identities, associating it with a consensus group, a validation tower and a storage shard.

Tags Polkadot (without directly mentioning it) as *complete rubbish*.

Is a bit similar to the proposed Ethereum sharding, but more generalized (?).
Not clear how the number and size of shards is controlled.
A guy asked if the association of shards to processes also changes like for consensus. The answer was basically: not decided, various designs possible.
This is a critical part imo. State can't as easily be switched around as the task to create a signature.
Security of the chain depends on state transitions being done correctly.
It's also not clear to me how non-leaf layers of the validation towers are supposed to check if a previous validation was correct. Will the leafs forward part of the state?

Other applications of randomness: e.g. instant and cheap decentralized search.
Lazy validation: validate after the fact. Cut deposit if cheated (for fast search results).

Statement: current crypto currency is not currency, but speculation coins.

Todo: following details on the Blockchain Nervous System.

History

#1 - 06.03.2017 11:47 - didi

- Subject changed from Research dfinity to Research dfinity / BNS

- Description updated