A Blockchain Computer [DRAFT/NOTES]

Rasmus Erik Voel Jensen

2018

Abstract

This paper proposes a new decentralised computer architecture with the following features: A blockchain is used for safe scheduling and verification of tasks among nodes with an economic incentive. Verifiable tasks include computation, message passing, storage, etc. Computational tasks can be done with performance within an order of magnitude of running it locally. The economy is based on supply and demand of computational resources. The currency is strongly linked to the cost of computation. Nodes are designed to run within a web browsers, such that a network of web frontend applications can run its own backend functionality.

Introduction

Nodes and network

The blockchain computer consist of a distributed network of nodes. Each node has an address which is the hash of its public key. Nodes are connected in a Kademlie-like topology.

Each node has a balance, that increases when it does work for the network, and decrease when it schedules tasks.

Memory

The memory of the blockchain computer is a distributed hash table. The address of a node determines which parts of the data it stores. Data is duplicated across nodes with nearby addresses. The data for a given node, will be stored at the hash of the public key of the node, and thus on the node itself, and in nearby nodes. Each node has a constant limit of how much data it can store.

The balance of a node increases when it is online and stores nearby data, and decreases when it is offline. The cost of storing data corresponds to the density of online nodes at the point of data. If the balance reaches 0, its data is expunged from the memory.

Each block of the blockchain, contains the root hash of the merkle tree of a snapshot of the entire memory, including the list of online nodes at the time of the snapshot. This can be calculated with a distributed divide-and-conquor algorithm across the network, in $O(\log n)$ where n is the number of nodes, and the constant factor correspond to the network latency. This is also the clock of the computer.

A proof of certain data in the memory at a given time can be given by through the merkle path through the blockchain and memory in $O(\log n) + O(\log t)$ space where n is the number of nodes, and t is time passed.

Computation

The block clock

The blockchain computer has a global state, that is distributed across the network.

A block of the blockchain computer is the merkle-hash of the computers state at that point in time. The clock blockchain computer, increments each time the

Each node

Architecture

We can safely store data on an untrusted network, by duplicating it across several nodes, and keeping the cryptographic signature. We can safely do computation on an untrusted network, by running the computation on different random nodes, and verifying that the results are the same. If we keep a blockchain of

A node is computer participating in the network. The address of a node is the hash of its public key, which will be evenly distributed. Nodes are connected in a kademlia-like topology.

The blockchain computer has a global state. Each node stores a part of the state

The blockchain computer has a global memory on which computation runs in parallel.

Verifiable tasks

Computational performance

Conclusion