A Blockchain Computer [DRAFT/NOTES]

Rasmus Erik Voel Jensen

2017

Abstract

We propose a design for a new kind of decentralised trustless computer. The shared state is stored in a blockchain. This allows computations to be distributed safely across without trusting indivudual nodes. It also allows proofs of work, and thus crediting nodes for their commputations. Computations are securely distributed and verified, through a blockchain containing the state By storing shared state in the blockchain, it is possible to By storing the machine state in the blockchain, it is possible to securely run distributed computations, without trusting the individual nodes. Individual nodes only need to know/store the small subset of the blockchain that they need for their computation.

Outline:

- Introduction
  - Motivation
  - Related work
- Architecture
  - State
  - Computation
  - Scheduling of computation
  - General tasks
- Future work
  - Actual implementation (in progress)
  - Stakes in addition to proof of work for better security

Background

Use case

Concrete use case: Web Applications without backend.

Related Work

TODO: explore these in more details, and document differences to our approach

Ethereum ...

Golem ...

Computes.io https://blog.computes.io/distributed-computed-centralized-vs-decentralized-c1d2120

TrueBit https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf

iEx.co ??

Architecture

State

stored in blockchain, for proofs and scheduling

Computational tasks

provable results

- task definition (and max amount of work)
- scheduling and computation
- proof of work done, without revealing value
- reveal result (and amount of work)
- update of ledger

Distributed ledger

The currency is bound to the value of computational work, and not based on artificial scarcity. Upper bound on value: solving a computational task gives the node currency corresponding to amount of computing power used. Lower bound on value: the currency is used to

Conclusion

Future work

Finishing the actual implementation.

Generalise computational tasks to storage, bandwidth, etc.

Adding stake, in addition to proof of work for better security.