Technische Universität München

Fakultät für Information Systems

Master's Thesis in Informatik

Bitcoin-like Blockchain Simulation System

Fabian Schüssler

Technische Universität München

Fakultät für Informatik

Master's Thesis in Information Systems

Bitcoin-like Blockchain Simulation System

Bitcoin-ähnliches Blockchain Simulationssystem

Author:	Fabian Schüssler
Supervisor:	Prof. Dr. Hans-Arno Jacobsen
Advisor:	Pezhman Nasirifard
Submission:	xx.xx.20xx

I confirm that this master's thesis is my own work and I have documented all sources and material used.

München, xx.xx.20xx

(Fabian Schüssler)

Abstract

English Abstract

Inhaltsangabe

German Abstract

Acknowledgment

Acknowledgement

Contents

List of Figures

List of Tables

Abbreviations

tps	transactions per seconds.
VIBES	Visualizations of Interactive, Blockchain, Extended Simulations.

Chapter 1

Introduction

Currently and in the last years Blockchain technologies such as Bitcoin and Ethereum are a very hot topic. According to the Gartner Hype Cycle [1] Blockchain technology is undergoing the peak of inflated expectation in 2017. Price and market capitalization changes of cryptocurrencies are covered by the media.

Blockchain technology enables decentralized consensus and can be used for record keeping. Bitcoin is the first digital currency to solve the double-spending problem without the need of a trusted authority. One of the main problems of Bitcoin or in general of Blockchains is the low maximum amount of possible processed transactions per seconds. Additionally, the Bitcoin community disagrees about how to solve this scalability problem, already split about different approaches and created Bitcoin forks.

VIBES (Visualizations of Interactive, Blockchain, Extended Simulations) is a blockchain sim-ulator, which allows fast, scalable and configurable network simulations on a single computer without any additional resources. It was developed in a master thesis by Lyubomir Stoykov [2] and which is the foundation for this master thesis proposal. The goal of my master thesis is to improve VIBES to make more realistic simulations possible. In the future VIBES could be used by developers or heavy blockchain users to simulate changes to different Blockchains. So maybe in the future it can help the Bitcoin community to agree on change proposals.

There are other Simulators like Bitcoin-Simulator, but VIBES is the first simulator designed to be extended to blockchain systems beyond bitcoin and the first of its kind to be able to simu-late transactions in the network. Bitcoin-Simulator only simulates the network at block level and therefore does not consider transactions [3].

1.1 Motivation

Motivation of Thesis.

1.2 Problem Statement

There are lots of possibilities to improve VIBES, which were also already outlined in the mas-ter thesis of Stoykov. The portability to different kinds of Blockchains and the ease of use is very important.

Here are some possibilities to extend the current version of VIBES, that I would like to work on in my master thesis: • Add maximal block size and transaction incentives • Break propagation delay down into three components: network bandwidth, block size and distance between Nodes. • Improve speed by testing mutable variables • Differentiate between miners and full nodes • Calculate resources used by network: CPU, electricity and bandwidth. • Change transaction generation from coordinator to nodes. • Other improvements: code improvements, analysis and visualizing of information that is already captured by VIBES (probability of forks), . . .

These extensions can improve the quality of the simulations and the use cases of VIBES. For example, the implications of Segwit2x could be analyzed. Maybe these extensions could also make it possible to realistically simulate the current Bitcoin blockchain.

1.3 Approach

The goal to make simulations more realistic. The approach from the original VIBES paper is taken over. Some designs and parts of the architecture have to be adjusted.

The configuration parameters (4.1.8) must be extended to break down propagation delay, to add maximal block size and transaction incentives.

Change transaction generation from coordinator to nodes (4.3.1).

Differentiate between miners and full nodes (4.3.2).

1.4 Organization

Organization of Thesis.

Chapter 2

Background

Background concepts required for understanding the thesis.

2.1 Bitcoin

2.1.1 Node/ Miner

in this work node = miner

2.1.2 Block

genesis block

2.1.3 Block size

2.2 Alternative History Attack

Chapter 3

Related Work

Related work materials

3.1 VIBES: Fast Blockchain Simulations for Large-scale Peer-to-Peer Networks

3.1.1 Prerequisites

3.1.2 The Actor Model

3.1.3 Executables and Work Requests

3.1.4 Best Guess

3.1.5 Fast-forward

3.1.6 Priority Queue

3.1.7 Votes

3.1.8 Executable Types

3.1.9 Configuration parameters

3.2 Analysis of hashrate-based double-spending

Chapter 4

Approach

In this chapter the changes to the existing VIBES framework are presented one by one. First the reasons for each improvement are described. Then the prerequisites for every change and the design and architecture are shown. Finally the implementation is explained. Part of the implementation is the implementation of the front-end. The front-end shows the results of the back-end and can therefore also be used for the evaluation.

4.1 Bitcoin-like Blockchain Simulation

changes necessary to differentiate strategies

4.2 Lazy Logging

Previously logging only occurred in the console. This made debugging for long simulations difficult. Especially for the evaluation of any implementations a log file is necessary. For this reason the Scala modules *logback* and *scala-logging* were integrated into the project. Every important event is logged into */logfile.log*.

4.3 Block size limit

One of the biggest unresolved issues of Bitcoin-like Blockchains is scalability. The main factor to measure scalability is transactions per seconds (tps). Previously VIBES had no block size limit. This means infinite transactions can be processed and changing input parameters has no effect on the scalability. To be able to investigate the effects of different input parameters on the scalability, the introduction of a block size limit is necessary. This allows a more accurate simulation of Bitcoin.

4.3.1 Prerequisites

configuration parameters

- *maxBlockSize*: the max block size in KB, the current default value is 1.000 KB

4.3.2 Design and Architecture

The main back-end change happens in the model VBlock. Depending if the simulation is a Bitcoin-like Blockchain Simulation all generated blocks obey the block size limit.

4.3.3 Implementation

```
object VBlock extends LazyLogging {
  def createWinnerBlock(node: VNode, timestamp: DateTime): VBlock = {
    var maxTransactionsPerBlock : Int = 0
    var processedTransactionsInBlock: Set[VTransaction] = Set.empty

    if (VConf.strategy == "BITCOIN_LIKE_BLOCKCHAIN") {
      // todo think about if to implement SegWit (maxBlockWeight vs
         ↪ maxBlockSize)
      maxTransactionsPerBlock = Math.floor(VConf.maxBlockSize /
         ↪ VConf.transactionSize).toInt

      // sorts the transaction pool by the transaction fee
      processedTransactionsInBlock =
         ↪ node.transactionPool.toSeq.sortWith(_.transactionFee >
         ↪ _.transactionFee).take(maxTransactionsPerBlock).toSet

      // sets confirmation status of transaction true
      processedTransactionsInBlock.foreach { _.confirmation = true }

      // sets confirmation level of transaction
      processedTransactionsInBlock.foreach { _.confirmationLevel =
         ↪ node.blockchain.size }
    } else {
      maxTransactionsPerBlock = node.transactionPool.size
      processedTransactionsInBlock = node.transactionPool
    }

    VBlock(
      id = UUID.randomUUID().toString,
      origin = node,
      transactions = processedTransactionsInBlock,
      level = node.blockchain.size,
      timestamp = timestamp,
      recipients = ListBuffer.empty,
      transactionPoolSize = node.transactionPool.size
    )
  }
}
```

4.4 Transaction fees

4.5 Block time

4.6 Alternative history attack

4.6.1 Prerequisites

To simulate an alternative history attack additional **configuration parameters** are necessary. These parameters are used for the actual simulation of the attack, the calculation of the success probability of the attack and the maximum safe transaction value.

- *isAlternativeHistoryAttack*: if an alternative history attack is simulated as a boolean
- *hashrate*: attacker's hashrate as a percentage of the total hashrate of the Bitcoin Network
- *confirmations*: the amount of confirmations the attacked merchants are waiting for to accept a transaction
- *attackDuration*: the attacker gives up after mining a certain amount of blocks and not succeeding or if it is not possible any more to surpass the level of the honest blockchain
- *discountOnStolenGoods*: discount of the stolen goods by the attacker, a value from 0 (= full discount) to 1 (= no discount)
- *amountOfAttackedMerchants*: the attack is carried out against a certain amount of merchants at the same time
- *blockReward*: current block reward in BTC

4.6.2 Design and Architecture

Simulating the attack

In the following the attacker's nodes, blockchain or blocks are interchangeably described as *evil* and the honest networks' nodes as *good.*

The solution for the simulation of an alternative history attack selects nodes as attacking nodes according to the attacker's hashrate as a percentage of the total Bitcoin Network. The good and the evil nodes both can mine the genesis block. The genesis block is then the first block in both the good and the evil blockchain. For simplicity we assume that the attacker successfully sent the transactions to the attacked merchants in the second block of the honest blockchain. Immediately after the genesis block is mined, the evil nodes start mining together on their own evil blockchain. It is necessary for all nodes to update their neighbour nodes to only have their corresponding nodes as neighbours. For example in the case of a low amount of evil nodes and a low amount of neighbours... Since the attacker of course doesn't want his...

Finally the success of the simulated attack is decided if the attacker's blockchain level can surpass the honest's blockchain level after waiting for the Merchants confirmation and before the attack duration ends.

Calculating the success probability [4]

The ... is ... Before the formula to calculate the success probability of an alternative history attack is shown, the variables need to be explained. q is *hashRate*, the attacker's percentage of the hash rate of the total network. p is $1 - q$ and the percentage of the honest network.

$$p + q = 1 \tag{4.1}$$

It is the goal to calculate the success probability r. If the attacker's hash rate q is equal or bigger than p, then the success probability of the attacker is 100%.
Due to the implementation the behaviour of the implementation can deviate from the 100%. For example the variables *attackDuration*, *confirmations* or the simulation duration can have an impact.
If $q < p$, then the upper complex formula with binomial coefficients needs to be calculated.

$$r = \begin{cases} 1 - \sum\limits_{m=0}^{n} \binom{m+n-1}{m} (p^n q^m - p^m q^n), & \text{if } q < p \\ 1, & \text{if } q \geq p \end{cases} \tag{4.2}$$

The formula for $q < p$ is transformed for the implementation. This allows the usage of factorial functions instead of binomial coefficients.

$$r = \begin{cases} 1 - \sum\limits_{m=0}^{n} \frac{(m+n-1)!}{m!\,(n-1)!}(p^n q^m - p^m q^n), & \text{if } q < p \\ \\ 1, & \text{if } q \geq p \end{cases} \tag{4.3}$$

Calculating the maximal safe transaction value [4]

(4.3)

$$\frac{(1-r)oB}{k(\alpha + r - 1)} \tag{4.4}$$

4.6.3 Implementation

Chapter 5

Evaluation

The evaluation chapter.

Chapter 6

Summary

Summary

6.1 Status

Final Status of the Thesis

6.2 Conclusions

Concluding remarks of Thesis

6.3 Future Work

Future Work

for bitcoin related stuff

node != miner selfish mining mining pools

Appendices

Appendix A

Appendix

A.1 First

First Appendix

Bibliography

[1] K. Panetta, “Top trends in the gartner hype cycle for emerging technologies, 2017.” https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/, 2017. Accessed: 2018-04-15.

[2] L. Stoykov, “Vibes: Fast blockchain simulations for large-scale peer-to-peer networks,” Master’s thesis, Technische Universität München, 2018.

[3] *On the security and performance of proof of work blockchains*, 2016.

[4] M. Rosenfeld, “Analysis of hashrate-based double spending,” 02 2014.