# Visualization of Internal Blockchain Processes

Shao-Chun Ma
shao-chun.ma@rwth-aachen.de
Matriculation number: 373895

Supervisor: Prof. Dr. Thomas Rose
Advisor: Thomas Osterland

Chair for Computer Science 5
Information Systems
RWTH Aachen

This thesis is submmited for the degree of
M.Sc. Software Systems Engineering

Aachen, Germany
May 2018

# Abstract

In a blockchain system, many nodes spread transactions and blocks simultaneously. Therefore, the behaviors of nodes and the mining processes are complex and difficult to identify. In this paper, we present a visualization tool which identifies the mining strategies of miners and the delays of networks in a blockchain system which employs proof-of-work as the consensus protocol. Our approach is based on the simulation of a blockchain system, which is built on a platform of the multi-agent system. The data sent between the nodes are monitored and recorded by the watchdog, and then they are sent to the visualizer. As a result, the visualizer can provide a fantastic visualization of internal blockchain processes in real-time. The visualization tool can help researchers to analyze the mining processes clearly and correctly.

Keywords: blockchain; visualization; mining strategies; delays of networks; proof-of-work

# Declaration

fff

# Acknowledgements

fff2

# Contents

# Chapter 1

# Introduction

The blockchain technology is first proposed in Satoshi Nakamoto's paper, Bitcoin: A Peer-to-Peer Electronic Cash System, in 2008. It created a new type of digital currency, called cryptocurrency. For example, Bitcoin and Ethereum are two of the well-known cryptocurrency. Cryptocurrency has a significant characteristic that distinguishes it from other currency: it is decentralized. This means that there is no authority in cryptocurrency. Therefore, it is important to guarantee the validity of transactions, and it is fulfilled by consensus protocols which use cryptography techniques.

The major process of the consensus protocol starts with the creation of a transaction. This transaction is distributed over the blockchain network. In this state, the transaction is not-mined and therefore not persistent. Every node in the network has their own transaction pools that contain the not-mined transactions.

In a blockchain system with the proof-of-work protocol, the nodes maintain the blockchain data structure by themselves. They compete against each other in extending the blockchain by creating new blocks that persistently store transactions from the transaction pools. The addition of new blocks provides effort, since it is necessary to solve a cryptographic puzzle with the characteristic that it is hard to solve, but easy to verify given a correct solution. The solving process is called mining. A correct solution is distributed over the network and verified by the remaining network participants. In the case that the verification succeeds, the block is added to the blockchain, and the containing transactions are removed from the transaction pools.

There are two factors that play important roles in the mining processes: the delays of networks and the mining strategies of miners. Transactions are published through the blockchain network, but each node receives them at a different time due to the unstable network. Therefore, each miner has different pending transactions in their own transaction pools. Moreover, each miner mines a block according to their own mining strategies simultaneously. The blocks generated by miners are different from each other, and they are also published through the unstable network. Consequently, it is possible that the set of nodes partitions into different groups that work on different instances of the blockchain. These blockchains are maintained

simultaneously until one blockchain becomes longer. The longest blockchain in the network is assumed to be the correct one, which is resolved by the consensus protocol.

## 1.1 Motivation

# Chapter 2

# Related Works

# Chapter 3

# Blockchain System

# Chapter 4

# Implementation

# Chapter 5

# Scenarios

# Chapter 6

# Conclusion

# Appendix A

# Source Codes