DEPARTMENT OF
COMPUTER SCIENCE

RICARDO FILIPE MENDES LOUREIRO
Masters in Computer Science and Engineering

REASONING ABOUT CONSENSUS PROTOCOLS:

SIMULATION AND VALIDATION ENVIRONMENT

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
Jully, 2023

DEPARTMENT OF
COMPUTER SCIENCE

REASONING ABOUT CONSENSUS PROTOCOLS:

SIMULATION AND VALIDATION ENVIRONMENT

RICARDO FILIPE MENDES LOUREIRO
Masters in Computer Science and Engineering

Adviser: António Ravara
Assistant Professor, NOVA University Lisbon

Co-adviser: Simão Melo de Sousa
Associate Professor, University of Beira Interior

Dissertation Plan
MASTER IN COMPUTER SCIENCE AND ENGINEERING

NOVA University Lisbon
Jully, 2023

Abstract

The number of services and applications that require and rely on transactional, replicated and verifiable data to function is increasing with each passing day, from banking and financial applications to online voting. With these requirements also come challenges, like availability, consistency, and security mechanisms that allow for integrity, non-repudiation and encryption of messages.

A common solution that these applications use to satisfy these requirements is Distributed Ledger Technologies or DLT. These systems are characterized as having a decentralized database, consensus mechanisms to validate transactions and immutable data once verified. But with the wide range of different requirements come a wide range protocols. These require testing, validation, and inevitably come vulnerabilities or errors that need to be corrected. Since the use of DLTs is recently new there is a lack of tools for the testing and validation of these protocols, which means that problems with the logic of the algorithm or vulnerabilities are often discovered in live applications.

One of these tools is MOBS(meter referência), which stands for Modular Blockchain Simulator, this simulator was built with the extensibility and modularity in mind, allowing users to simulate any family of protocols as well as parametrize multiple scenarios to study these protocols. These parameterizations include bandwidth limits, byzantine behavior of the participant nodes and adversarial behavior. After the execution the desired statistics and information needed to validate the execution of the protocols can also be parametrized.

In this document we propose an extension to this tool to better allow the simulator to provide different sets of execution environments by allowing parameterization of the network layer of the simulator. This will allow for the definition of membership protocols in which this layer will be built, either with structured or unstructured overlays or with the help network optimization algorithms, allowing the study of the performance and correctness of these protocols in dynamic and different network layers that will independently respond to byzantine behaviors or network changes.

Keywords: Distributed Ledger Technology

Resumo

O número de serviços e aplicações que necessitam e utilizam dados replicados, verificáveis e transacionais para funcionarem estão a aumentar com cada dia que passa, desde aplicações bancárias e financeiros a sistemas de voto online. Com estes requisitos também vêm desafios, como disponibilidade, consistência dos dados e mecanismos de segurança que permitam a integridade, a não repudiação e a encriptação de mensagens.

Uma solução comum que estas aplicações usam para satisfazer estes requisitos são Tecnologias de Registos Distribuídos ou TRD. Os sistemas de TRD são caracterizados por terem uma base de dados descentralizada, mecanismos de consensos para validar transações e dados imutáveis após verificados. Mas com a vasta diversidade de requisitos vem uma quantidade diversa de protocolos. Estes necessitam de ser testados, validados e inevitavelmente corrigir os erros de lógica e as vulnerabilidades descobertas ’a posteriori’. Como o uso de TRDs é relativamente recente, há uma falta de ferramentas para testar e validar estes protocolos, o que implica que problemas de lógica ou vulnerabilidades são muitas vezes descobertos depois das aplicações que os utilizam serem lançadas.

Uma destas ferramentas é o MOBS(referencia), Modular Blockchain Simulator, este simulador foi construído com a extensibilidade e modularidade em mente, permitindo os utilizadores simular qualquer família de protocolos tal como parametrizar múltiplos cenários para o estudo destes. Estas parametrizações incluem limites de bandwidth, comportamentos bizantinos dos nós participantes e comportamento adversarial. Após a execução as estatísticas escolhidas e informações necessárias para a validação da execução destes protocolos também pode ser parametrizada e estendida.

Neste documento propomos uma extensão desta ferramenta para melhor simular e fornecer diferentes conjuntos de ambientes de execução permitindo a parametrização da camada de rede do simulador. Isto vai permitir a definição de protocolos onde esta camada vai ser construida, tanto com uma rede estruturada ou não estruturada, ou permitindo à rede a execução de algoritmos de otimização para a mesma, permitindo assim o estudo do desempenho, a correção destes protocolos num ambiente dinâmico, em diferentes camadas de rede que vão independentemente responder a comportamentos bizantinos ou mudanças na camada de rede.

Palavras-chave: Tecnologia de Registo Distribuído

Contents

Appendices

Annexes

List of Figures

List of Tables

1

Introduction

This thesis aims to address the practical challenges in designing, implementing and maintaining Distributed Ledger Technologies protocols by providing a simulation environment so that results can be extracted and the behavior of the protocols can be verified before going into a live environment. This thesis involves taking MOBs and expanding the previously done work to help better test and validate these protocols by providing a wider range of networks and a more dynamic and independent network layer.

1.1 Context

With each passing day the amount of distributed application is increasing, and a subset of these are applications that deal with data that requires validation, support transactional operations by validated users and simultaneous access for updating and consulting the records. For this specific subset there are a group of protocols that have been created to meet and ensure these requirements. Distributed Ledger Technologies allow for simultaneous access, validation and update of records across a distributed database, each node has its own copy of the ledger that it uses to validate information and reach a consensus about its accuracy.

Around 2008 began the rise of a new set of protocols that takes DLTs as their basis. Blockchain protocols appeared with the motivation to serve as a distributed ledger for cryptocurrency transactions. The main difference from Blockchains to DLTs is that the log of records is created in blocks, each blocked is closed by a hash and the next beginning with the closing hash of the previous.

These protocols are not static, being because vulnerabilities or flaws need to be correct or due to the very nature of the protocol itself. One of these dynamic protocols is Tezos(reference miguels thesis), which relies on the stake-holders that participate in the system to propose and agree on changes and upgrades to the protocol. Another example is Ethereum(reference see miguels thesis) that uses a blockchain protocol based on proof of work and their current goal is to migrate towards an implementation based solely on proof of stake for Ethereum2 (reference see miguels thesis). This opens a necessity for tools that aid in support these evolutions in a faster, more seamless, secure fashion.

Blockchain protocols operate on top of membership layers that dictate how the topology of the network is configured. Different membership environments come with different properties and trade-offs. Structured membership overlays allow for faster lookups for specific nodes and a pre-defined and predictable structure to the network. Non-structured membership offer a more resilient overlay when new nodes are introduced or existing ones leave, albeit by choice or failure. And overlays that operate by building a partially structured overlays allow for the benefits of a non-structured overlay at the cost of slower restructuration since optimization procedures are regularly executed to improve routing and lookup operations. The trade-offs and some of these protocols will be further explained in Chapter 2.

This motivated the development of MOBs, a modular and extensible simulator that provides the ability to simulate different families of protocols, parameterizable, a well-defined structure and a qualitative evaluation for the study of implemented protocols. But this simulator can be further improved by giving conditions closer to real life, like a dynamic and parameterizable network layer, allowing for the study and validation of the behavior of these protocols in different scenarios and abstracted from the intricacies of the arrangement of the participants.

1.2 Problem

Consensus protocols are not trivial to define or understand correctly and in blockchain systems, where the behavior is dynamic and mostly financial transactions are dealt with, their correctness is crucial and errors can be costly. To help with this MOBs was developed giving the ability to test and validate these protocols under different conditions and settings by changing the execution parameters.

Right now the parameters to the network are set before the execution of the simulation, and regarding the network behavior we can set the network topology, how many nodes will fail and when. In the real world a network's topology is dynamic, new nodes can enter as new participants, existing ones can leave, either by choice or by failure, and even when the participants are static the network can suffer changes to its topology as a result of optimizations performed by this layer.

The simulation of how this layer behaves and the parametrization of the different protocols that can be used as its basis are the problems that we aim to address with this thesis.

1.3 Goal

With the different properties that different network overlays provide besides the parametrizations that are already provided by MOBs, we can further provide an even more modular simulator that allows for the study of the optimal conditions of executions of the protocols. We can also leverage the modularity of this layer to better simulate the behavior of new participants coming into the system, existing ones leaving and how changes to the topology of the network affects their execution.

The goal of this thesis is to improve the networking layer of MOBs by making it more modular and therefore giving the following advantages:

- Different environments for more diverse testing scenarios.

- Stronger parametrization regarding the behavior of the network.

- Better qualitative data by providing scenarios that better mimic real world execution.

1.4 Document Organization

The remainder of this document is organized in the following manner:

Chapter 2 studies related work: in particular we will cover several membership protocols and some implementations; MOBs and the work that has already been conducted on this simulator;

Chapter 3 describes in more detail the work that has already been conducted and the proposed work, including an evaluation and work plan

2 Related work

In this chapter we will discuss relevant work considering the goals of the work to be conducted in this thesis. In particular, we will focus on the following topics:

In section 2.1 we discuss the differences between the different kinds of membership systems as well as some implementations.

In section 2.2 we will discuss Babel(insert reference), a framework to develop distributed protocols.

In section 2.3 we will present and compare some protocol analysis tools and compare them to MOBs.

2.1 Membership Protocols

This section introduces some membership protocols that have been considered to part of the work that will be developed.

2.1.1 Membership

A protocol where each node knows every other node in the network might work well for small networks, but it is not a scalable solution since each node would need to have *n - 1* communications channels open at all time, being n the amount of nodes in the system, and each node needs to be following any and all changes to the system. This is not feasible since the number of links between nodes would rise quadratically and networks can easily reach thousands of participants. In order to overcome the challenges that arise from large networks, we use partial view membership protocols. In these protocols each node only knows and maintains information about a small selection of nodes in the systems, making it a more scalable strategy than a total view protocol, since the number of connections grow at a logarithmic rate instead.

When maintaining a partial view, protocols usually follow one of two strategies when managing their memberships:

- **Reactive:** With this strategy the partial view only changes when the membership suffers changes, usually by a node leaving or joining the membership

- **Cyclic:** With this approach, the partial view changes periodically. Every *t* seconds nodes exchange information that may lead to changes to the partial view.

This membership protocols may be represented by a graph where the vertices represent the participants of the networks and the edges represent the communication links. Depending on how the graph ends up we can classify the membership as structured, unstructured or partially structured. In the next sections we will go over these types of memberships and some implementations.

2.1.2 Structured Overlays

Structured overlay memberships follow a pre-defined structure by having the nodes routed to a specific logical position in the network. This known structure allows improvement on search primitives, enabling efficient discovery of data and process. This position is usually determined by giving a unique identifier for each node of the network.

However, having a predefined topology come at the cost of more costly re-structuring usually by a node leaving the network or a new participant joining. This process is usually slow and costly since a change of one node in the membership may affect the network at a global level. This drawback becomes more relevant in high churn rate scenarios.

2.1.2.1 Chord

Consistent Hashing and Random Trees, or Chord(referencia) is a Distributed Hash Table based algorithm where each node is assigned a unique identifier based on the output of a hash function. Usually this has is based on the IP address of the node since this is unique for every node of the network, and making it so that once a node joins the network their identifier is set in stone. These identifiers are also so that each node knows which keys of the distributed hash table are assigned to them, since the range of keys that belong to a certain node is represented by the range defined by their identifier up to the identifier of the next node.

The graph generated based on the membership overlay will be a cyclic graph with a ring shape, with the nodes ordered by their generated identifier. Each node also maintains a routing table with around O(log n) distinct entries called a finger table. This routing table is used in order to navigate the overlay in a more efficient manner.

The tables maintained by these nodes are automatically updated when a new node joins or leaves the system making it always possible to find a node responsible to a given key. However, simultaneous failures may break the overlay since the correctness of the membership depends on each node knowing their correct neighbors. In order to increase fault tolerance, a periodical stabilization protocol is run in background, making sure the neighbors are still active and restructuring the overlay accordingly as well as updating the entries on the finger table.

2.1.2.2 Pastry

Pastry(insert reference), similarly to Chord is a structured overlay protocol that defines a distributed hash table. A Pastry system is a self-organizing overlay network of nodes where every node has a 128-bit identifier. This identifier is assigned randomly when a node joins the system and is used to indicate a node's position in a circular space that ranges from 0 to $2^{128}-1$. It is assumed that the hash function that gives the node's identifier generated a uniform distribution of identifiers around the 128-bit space.

2.1.3 Unstructured Overlays

2.1.3.1 SWIM

2.1.3.2 HyParView

2.1.4 Partially Structured Overlays

2.1.4.1 T-MAN

2.1.4.2 X-BOT

2.2 Babel

2.3 Protocol Analysis Tools

3

The simulator

3.1 Overview

3.2 Critical Analysis

3.3 Initial Exercise

4

Work to be developed

4.1 Proposed work

4.2 Evaluation

4.3 Work Plan