Vrije Universiteit Amsterdam

Universiteit van Amsterdam

Master Thesis

[WIP]Optimistic and Wait-free Concurrent Execution of Blockchain Transactions

Author: Kian Paimani (2609260)

1st supervisor:	dr. ir. A.L. Varbanescu	
daily supervisor:	supervisor name	(company, if applicable)
2nd reader:	supervisor name	

A thesis submitted in fulfillment of the requirements for the Master of Science degree in Parallel and Distributed Computer Systems

May 9, 2020

"In the future, trusting an opaque institution, a middleman, merchant or intermediary with our interest, would be as archaic a concept, as reckoning on abacuses today"

– Dr. Gaving Wood

Prelude

This will be prelude. Some wise words about Web3, and how it will evolve from web2 and how web2 was designed in a peer to peer fashion but ended up in this mess that it is now.

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac orci a nulla finibus ornare. Proin ultricies tellus a metus facilisis tristique. Vivamus vel lectus magna. Donec nibh sapien, pulvinar ut hendrerit nec, porta nec est. Nam rutrum aliquet egestas. Etiam nibh ex, ultrices vel arcu eu, vulputate venenatis enim. Vestibulum vel nisi quis libero finibus sollicitudin. Aenean ornare nibh id tincidunt tempus. Duis imperdiet, dui sed finibus tempor, lacus enim tempor sem, eget fermentum enim magna sit amet nunc. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. In malesuada ligula risus, sit amet efficitur quam accumsan sed. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Mauris non felis sed felis euismod efficitur interdum eget massa. Nullam eget accumsan arcu. Nunc suscipit volutpat ligula sed blandit.

Vivamus lobortis leo est, gravida consequat risus volutpat eget. Morbi quam mi, porta ut vulputate in, gravida ut ipsum. Ut ultricies ullamcorper molestie. Suspendisse vestibulum eros eget elit finibus volutpat non vitae risus. Pellentesque gravida, lacus sit amet egestas fringilla, diam erat laoreet augue, id feugiat orci lorem vitae augue. Sed augue augue, volutpat et fringilla quis, dictum vel velit. Nam et congue diam.

Integer rhoncus ipsum ante, non sollicitudin nisi sagittis vel. Duis rhoncus consequat ante at semper. Vivamus ex turpis, blandit in nisi eget, hendrerit scelerisque libero. Donec posuere dui libero, vitae ornare orci tempus sollicitudin. Mauris ut tortor vel orci malesuada blandit in a felis. Phasellus at congue mauris. Sed efficitur tellus at ligula euismod rutrum. Aliquam mattis scelerisque ultricies. Integer rutrum erat a tellus ultricies cursus. Integer gravida suscipit nulla, interdum lacinia elit dictum sed. Donec luctus nibh ac tortor rutrum rutrum. Aliquam vitae fermentum ante, ac maximus turpis. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aliquam vel neque id felis viverra pretium.

Nullam scelerisque, orci quis porttitor rutrum, orci magna mattis neque, ac vestibulum dolor erat eget mi. Curabitur porta, lectus ac faucibus pellentesque, ante diam pretium nisi, ut maximus mauris lorem tincidunt eros. Mauris scelerisque elit ut ante auctor, in lacinia mauris dictum. Nam nec mattis nulla, eget tincidunt leo. Ut bibendum at mi nec commodo. Suspendisse augue massa, pharetra id pellentesque at, finibus at odio. Vestibulum ac enim ut tellus fermentum feugiat nec id eros. Duis hendrerit sem et ornare hendrerit. Suspendisse at sem vel tellus aliquet tempor. Fusce accumsan elementum est, sed aliquet nibh facilisis at. Sed venenatis condimentum magna. Curabitur iaculis augue in arcu ullamcorper placerat. Suspendisse ac placerat justo.

Maecenas facilisis, enim quis fermentum rhoncus, nunc magna pretium risus, nec lacinia lectus metus sed tortor. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Curabitur hendrerit quam sed dolor vehicula pharetra. Morbi pretium ligula vitae nisi ultrices, in mattis nisi mattis. Fusce facilisis ex nec purus malesuada gravida. Quisque facilisis tincidunt sapien, ac dignissim libero porta sit amet. Curabitur in sem et libero imperdiet sollicitudin eget ut turpis. Morbi vulputate risus a augue lacinia, vehicula vestibulum odio aliquet. Donec placerat diam et lorem viverra aliquam. Praesent nibh ipsum, hendrerit eget ultricies mollis, convallis id est. Nullam imperdiet, arcu non faucibus malesuada, metus velit tempor diam, at semper erat urna sed orci.

Acknowledgements

Here goes the acknowledgements.

Contents

List of Figures

List of Tables

1

Introduction

> *"If Bitcoin was the calculator, Ethereum was the ENIAC*[1]*. It is expensive, slow and hard to work with. The challenge of today is to build the* ***commodity, accessible and performant*** *computer."*
>
> – Unknown.

Meeting notes:
Why is this problem interesting??
mostly short. Start with requirements (correctness).
Chapter 3 should be roughly design and impl details. and it should promise some goals that you meet in chapter 4.
Design in Notion is final. Write a set of requirements and explain **why** the design respects it.
related work: say where you fit in the literature overlay. No need to brag and**compare with others**.
experiment setup should prove the goal, not only speedup etc.

Blockchains are indeed an interesting topic in 2020. Some people skeptically see them as controversial, or merely a "hyped hoax", and doubt that they will ever deliver much real value to the world. On the other hand, many believe that it is a revolutionary technology that will shape our future societies, much like the internet and how it has impacted many aspect of how we live in the last few decades (1).

In a very broad term, a blockchain is a tamper-proof, append-only ledger that is being maintained in a decentralized fashion, and can only be updated once everyone agrees upon that change as a bundle of transactions. This bundle of transactions is called a **block**. Once this block is agreed upon, it is appended (aka. *chained*) to the ledger, hence the term block-*chain*. Moreover, the ledger itself is public and openly accessible to everyone. This means that everyone can verify and check the ledger and the transaction and block history, should they need to do it. At the same time asymmetric cryptography is the core identity indicator that one needs to interact with the chain, meaning that your personal identity can stay private in principle, if that is desired based on the circumstance.

In some sense, blockchains are revolutionary because they remove the need for *trust*, and release it from the control of one entity (i.e. a single bank or institute), by encoding it

[1] the first generation computer developed in 1944. It fills a 20-foot by 40-foot room and has 18,000 vacuum tubes.

as a self-sovereign decentralized software. Our institutions are built upon the idea that they manage people's assets, matters and belongings, and they ensure veracity, because we trust in them. In short, they have **authority**. Of course, this model could work well in principle, but it suffers from the same problem as some software do: it is a **single point of failure**. A corrupt authority is just as destructive as a flaky single point of failure in a software is. Blockchain attempts to resolve this by deploying software (i.e. itself) in a transparent and decentralized manner, in which everyone's privacy is respected, whilst at the same time everyone can still check the veracity of the ledger, and the software itself. In other words, no single entity should be able to have any control over the system.

Now, all of these great properties don't come for cheap. Blockchains are extremely complicated pieces of software and they require a great deal of expertise to be written correctly. Moreover, many of the machinery used to create this *decentralized* and *public* append-only ledger, requires synchronization, serialization, or generally other procedures that are slow and reduce the throughput at which the chain can process transactions. This, to some extent, contributes to the skepticism mentioned in at the beginning of this chapter. For example, Bitcoin, one of the famous deployed blockchains to date, consumes a lot of resources to operate, and cannot exceed a transaction throughput of more than half a dozen transactions per second (2).

Hence, we see it as in important goal to investigate the possibilities through which a blockchain system can be enhanced to perform *faster*, and more *efficiently*.

1.1 Research Question

Based on the mentioned scenario, we can settle on the basic goal of improving the *performance* and *efficiency* of a blockchain system. There are numerous ways to achieve this goal, ranging from zooming into the details such as concurrency within some components of the software, all the way back to tuning the network parameters and implementations. In this work, we precisely focus on the former, enabling concurrency within transactions that are processed and then appended to the ledger. This approach is better compared with the other alternatives at the end of chapter 2. We also mention in the same chapter why each of these approaches could have their own merit and value, and how they differ with one another.

All in all, we formulate the following as our research questions:

1. What approaches exist to deploy concurrency methods in the realm of blockchains, when seen as a generic decentralized transaction processing systems?

2. How deficient and reasonable are each these approaches, given that different blockchains can have radically different transaction types and concurrency requirements.

1.2 Rest of this work

The first half of this work leads the way toward our proposed approach to solving the aforementioned research question. . In chapter 2, we dive deep into the two pillars of background knowledge that we need to answer the above questions: blockchains and concurrency. First, we will clarify what exactly a blockchain is in section. Then, having known this, we can have a 100-feet view into a blockchain system and see what are the broad terms ways in which its speed can be improved in section 2.2 . Then, in chapter 3 we will define the requirements of the system and consequent design goals. This is followed by our proposed design in section 3.2. Finally, we provide some implementation details in 3.

should I say in this work or this thesis or our work or our thesi

In the second part of this work, we put our solution to action and observe the outcome. Chapter 4 will be devoted to the benchmarks and experiments through which we evaluate whether our research goals have been met. An important detail of this chapter is the data set, explained in section 4.2. In chapter 5 we compare our work with other similar research in the domain and see how it fits in the scientific literature. Finally, we conclude and propose future work in chapter 6.

1. INTRODUCTION

2

Background

"The use of credit cards today is an act of faith on the p a t of all concerned. Each party is vulnerable to fraud by the others, and the cardholder in particular has no protection against surveillance."

– David Chum et. al. - 1990

In this chapter, we will dive into the background knowledge needed for the rest of this work. Two pillars of knowledge need to be covered, blockchains in section 2.1 and concurrency, upon which our solution will be articulated, in section 2.3.

2.1 Blockchains And Decentralized Applications

In this section, we will provide an overview about the basics of distributed system, blockchains, and their underlying technologies. By the end of this chapter, it is expected that an average reader will know enough about blockchains system to be able to follow the rest of our work in chapter 3 an onwards.

2.1.1 Centralized, Decentralized and Distributed Systems

We cannot begin to describe blockchains before defining a distributed system. Blockchains, at the end of the day, or just another form of distributed systems. A distributed system is a system in which a group of nodes (each of which having an individual processor and memory) cooperate and coordinate for a common outcome. From the perspective of an outside user, most often this is transparent and all the nodes can be seen and interacted with, as if it was one cohesive system (3).

Blockchains differ in many ways from other distributed system, yet the underlying concepts resonate in many ways (4). Like distributed system, a blockchain system is also consisted of many nodes, operated either bur organizations, or by normal people with their commodity computers, and this trait is transparent to the end user, when they want ot interact with the blockchain.

Blockchains are also decentralized. This term was first introduced in a revolutionary paper in 1964 as a middle ground between purely centralized system that have a single point of failure, and a 100distributed system which is like a mesh, all nodes have links to many other nodes (5) [1]. A decentralized system falls somewhere in between, where no single node's failure can have a unrecoverable damage to the system, and communication is somewhat distributed, some nodes might act as hops between different sub-networks.

Blockchains, depending on the implementation, can resonate with either the above. Most often, from a networking perspective, they are much closer to the ideals of a distributed system. From an operational and economical perspective, they can be seen more as decentralized, where the operational power, i.e. the authority falls into the hands of no single entity.

maybe add a picture here from the old paper or similar to it?

2.1.2 From Ideas to Bitcoin: History of Blockchain

While most people associate the rise of blockchains with bitcoin, it is indeed incorrect and the basic ideas of blockchains was mentioned decades earlier. The first relevant research paper was already mentioned in the previous section. (5), along the definition of decentralized system, the paper also describes many other metrics regarding how survivable a network is, under certain attacks.

Next, (7) famously introduced what is know as Diffie-Hellman Key Exchange, which is the backbone of public key encryption. (7) is heavily inspired by the work of (8), which all together form the digital signature scheme which is heavily used in all blockchain systems [2].

Moreover, even the idea of blockchain itself heavily predates bitcoin. The idea of chaining data together, whilst placing the some digest of the previous piece (i.e. a *hash* thereof) in the header of the next one was first introduced in (9). This, in fact, is exactly the underlying reason that a blockchain, as a data structure, can be seen as a append-only, tamper proof ledger. Any change to previous blocks will break the hash chain and cause

[1] The design of Paul Baran, author of (5) originally was proposed, like many other internet-related technologies, in a military context. His paper was a solution to the US Authorities concerns about communication links in the after math of a nuclear attack in the midst of the cold war (6).

[2] Many of these works were deemed military applications at the time, hence the release dates are what is referred to as the "public dates", not the original, potentially concealed dates.

the hash of the latest block to become different, making any changes to the history of the data structure identifiable, hence *tamper-proof*.

Finally, (10) introduced the idea of using the digital computers as a means of currency in 1990, as an alternative to the rise of credit carts at the time. There were a number of problems with this approach, including the famous double spend problem, in which an entity can spend one unit of cash currency numerous times. Finally, an unknown scientist, who used the name Satoshi Nakatomo as an alternative, released the first draft of bitcoin whitepaper, in which he proposed proof of work as a means of solving the double spend problem, among other details and improvements (11). Soon after, the first implementation of bitcoin followed.

2.1.3 Basic Concepts

Having known where the blockchain's idea originates from, and which fields of previous knowledge in the last half a decade it aggregates, we can now have a closer look at these technologies and eventually build up a clear and detailed understanding of what a blockchain is and how it works.

2.1.3.1 Elliptic Curve Cryptography

- secp - edcsa vs schnorr

2.1.3.2 Hash Functions

- what a has function is - Blake

2.1.3.3 Peer to Peer Network

- Distributed in the network layer. Noes typically have the same role, although this is not absolute. - gossip messages/transaction/blocks.

2.1.3.4 Key-Value Database

- just a database - transaction is the way to update it.

2.1.3.5 Transactions and Signatures

- it needs to be accountable.

2.1.3.6 Blocks

- bundle of transaction + header + chained by a hash to the previous one.

2.1.3.7 Trie and Storage Root

2.1.3.8 Consensus and Block Authoring

2.1.3.9 Transaction Queue

2.1.3.10 Binary Encoding - Parity Scale Codec

2.1.4 Putting it All Together: Decentralized State Transition Logic

2.1.4.1 Disclaimer: The Context of Technology

Mention that some of this technology might vary from chain to chain, and that we do our best to stay neutral. When having to decide, we adhere to Substrate's standards, since we will be using the same underlying libraries as it does, and since it is the best blockchain framework of the time that allows us to experiment outside the scope.

2.2 Speeding up a Blockchain: A Broad Perspective

Blockchains are arguably among the most sophisticated peer-to-peer software deployed to date. They provide a trust-less environment in which different types of applications can be deployed. The early chains mostly adopted the application of being a digital currency, also known as *cryptocurrency*. Bitcoin (11) was the pioneering cryptocurrency, announced in 2009. Nonetheless, soon thereafter, other chains were designed and released that could function in a more generic way. Ethereum (12), was the first of such chains that was programmable via the notion of smart-contracts, small scripts that were stored on-chain and executed upon receiving particular transactions. In the broad term, regardless of the application, the blockchain can be seen as a distributed application which can be executed by the means of submitting a *transaction* to any of the nodes in its peer-to-peer network.

The blockchain industry brought a great deal of hype with it. This, to some extent caused many of the underlying technologies that power blockchains to grow at a fast pace. Most notably, many peer to peer technologies have observed a significant advent rate[1]. Nonetheless, one area is still lacking behind, which is their relatively poor *performance*. Some blockchain networks that are active today cannot exceed an overall throughput of more than a few dozens of transactions per second on average. This concern is the basis of this thesis.

In the next section, we will briefly survey some of the ways through which the throughput of a blockchain can be improved, and delineate which approach we will be focusing on for

[1]A simple query in google trends for terms such as "bitcoin", "blockchain" and "peer-to-peer" can show a direct correlation between the rise of bitcoin and the rest of the keywords

the rest of this thesis. Moreover, we will extract our exact research question from this brief survey. Note that we will not explain some blockchain concepts in-depth at this point and leave that for chapter 2.

As mentioned, blockchains can be seen, in a very broad way, and from a transaction processing point of view, as a *decentralized transaction processing network*. The throughput of a blockchain network, in transaction's per second, is a function of numerous components and can be analysed from different points of view. While in this work we focus mainly on one aspect, it is helpful to enumerate all viewpoints and see how they each affect the overall performance.

2.2.1 Consensus and Block Authoring

The consensus algorithm is the means by which the nodes in the network align their viewpoints on the state of the world, and come to agreement about it. Similarly, the nodes in the network must also decide when and who will have permission to alter the state, i.e. take the role of *author*. Two common consensus protocols are Proof-Of-**Work** (henceforth denoted as POW) and Proof-Of-**Stake** (henceforth denoted as POS). They use the computation power (*work*) and a number of bonded tokens (*stake*) as their guarantees that the author was indeed eligible for authoring a block. Without getting into further details about each protocols, what we care about is the fact that each of these consensus protocols has an *inherently* different performance (13). POW, as the names suggests, requires the author to prove their legitimacy by providing a proof that they have solved a particular hashing puzzle. This is slow by nature, and wastes a lot of computation power on each node that wants to produce blocks, which in turn can have a negative impact on the transaction throughput. Making this process faster requires the network to agree on an easier POW puzzle that can in turn make the system less secure (2). More precisely, the difficulty of the puzzle dictates the average time any node needs to spend to be able to produce a block, which dictates the final throughput.

To the contrary, POS does not need this this fruitless puzzle solving, which is beneficial in terms of computation resources. Moreover, since the chance of any node being the author is determined by their stake. Thus, a smaller block-time is not insecure by itself.

All in all, one general approach towards increasing the throughput of a blockchain is to *re-think the consensus and block authoring mechanisms* that dictate when blocks are added to the chain, and by whom, with what frequency. It is crucially important to note that any approach in this domain falls somewhere in the spectrum of centralized-decentralized, where most often approaches that are more centralized will be more capable of delivering better performance, yet they do not have any of the security and immutability guarantees of a blockchain.

In this work transcend from this point of view and will look at a different component of a blockchain system which is completely independent of the underlying consensus. The main reason for this is that this is entirely different domain of research compared to our proposed approach, concurrency.

2.2.2 Chain Topology

Another approach is changing the nature of the chain topology. A classical blockchain is theoretically limited due to the fact that only one entity can append to the block at each time. This property will bring extra security and make the chain state easier to reason about (i.e. there is only one cannon chain). A radical approach is to question this property and allow different blocks to be mined at the same time. Consequently, this turn a blockchain from a literal *chain of blocks* into a *graph of nodes*. Hence, most often such technologies are referred to Directed Acyclic Graphs, **DAG** in short, solutions.

Such approaches will bring even more radical changes to the original

2.2.3 Sharding

2.2.4 Networking

Further factors can exist, but not for a general purpose blockchain, hence we

2.2.5 Summary: Any push forward toward better performance is added value

2.3 Concurrency

2.3.1 Locking, RW-Locks and more.

2.3.2 Software Transactional Memory

2.3.3 Static Analysis

2.3.4 Transposition Driven Scheduling

3

System Design and Implementation

...

This chapter will basically explain the main idea of the thesis.

3.1 Requirements and Goals

3.2 System Design

3.3 Implementation

4

Benchmark and Analysis

...

This chapter will basically explain our benchmark, dataset and environment (like implementation details).

4.1 Simulation Environment

4.2 Data set (generation and collection)

4.3 Benchmark results

5

Related Work and Discussion

...

In this chapter we will survey the related works in the filed and discuss how our approach differs from them.

5.1 Related Work

5.2 Discussion

6

Conclusion

...

Final words and conclusions.

6.1 Conclusion

6.2 Further Work

Appendix

References

[1] Craig Pirrong. **Will Blockchain Be a Big Deal? Reasons for Caution**. *Journal of Applied Corporate Finance*, **31**(4):98–104, 2019. 1

[2] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. **On the Security and Performance of Proof of Work Blockchains**. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 3–16. Association for Computing Machinery, Oct 2016. 2, 9

[3] IMRAN BASHIR. *MASTERING BLOCKCHAIN: distributed ledger technology, decentralization, and smart contracts explained, 2nd edition;distributed ledger.* PACKT Publishing, 2018. 5

[4] Maurice Herlihy. **Blockchains from a distributed computing perspective**. *Communications of the ACM*, **62**(2):78–85, Jan 2019. 5

[5] P. Baran. **On Distributed Communications Networks**. *IEEE Transactions on Communications Systems*, **12**(1):1–9, Mar 1964. 6

[6] 1776 Main Street Santa Monica and California 90401-3208. **Paul Baran and the Origins of the Internet**. 6

[7] W. Diffie and M. Hellman. **New directions in cryptography**. *IEEE Transactions on Information Theory*, **22**(6):644–654, Nov 1976. 6

[8] Ralph C. Merkle. **Secure communications over insecure channels**. *Communications of the ACM*, **21**(4):294–299, Apr 1978. 6

[9] Stuart Haber and W. Scott Stornetta. **How to time-stamp a digital document**. *Journal of Cryptology*, **3**(2):99–111, Jan 1991. 6

[10] David Chaum, Amos Fiat, and Moni Naor. **Untraceable Electronic Cash**. In Shafi Goldwasser, editor, *Advances in Cryptology — CRYPTO' 88*, Lecture Notes in Computer Science, page 319–327. Springer, 1990. 7

[11] Satoshi Nakamoto. **Bitcoin: A Peer-to-Peer Electronic Cash System**. page 9. 7, 8

[12] Vitalik Buterin. **Ethereum: A next-generation smart contract and decentralized application platform**, 2014. Accessed: 2020-05-03. 8

[13] Alessio Meneghetti, Tommaso Parise, Massimiliano Sala, and Daniele Taufer. **A survey on efficient parallelization of blockchain-based smart contracts**. *arXiv:1904.00731 [cs]*, Feb 2019. arXiv: 1904.00731. 9