ON DATABASIFYING BLOCKCHAINS

by

RUAN PINGCHENG
(B.S., Nanyang Technological University)

A THESIS SUBMITTED FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

in the

GRADUATE DIVISION

of the

NATIONAL UNIVERSITY OF SINGAPORE

2021

Supervisor:
Professor OOI Beng Chin

Examiners:
Associate Professor CHAR Kway Teow
Assistant Professor Yummy Bee Hoon CRAB
Professor BAK Kwa, Dessert University

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

阮平成

RUAN PINGCHENG

1 September 2020

To my teachers, parents, peers and friends...

Acknowledgments

I would like to place my foremost and deepest gratitude to my supervisor, Distinguished Professor Beng Chin Ooi. Thank you for your guidance throughout my PhD years. In the early years when I feel lost from time to time, it is Professor Ooi who tirelessly guides me through the research process.

Two heads are better than one. This thesis is inseparable from the efforts of my collaborators. Chapter 3 is the joint work with Gang Chen, Tien Tuan Anh Dinh, Qian Lin, Dumitrel Loghin, and Meihui Zhang. Chapter 4 is collaborated with Gang Chen, Tien Tuan Anh Dinh, and Qian Lin. Chapter 5 is a result of the contribution from Dumitrel Loghin, Quang-Trung Ta, Meihui Zhang, and Gang Chen. I would like to take this opportunity to thank the above co-authors, which offer me abundant tips to conduct first-class research.

It is also my privilege to work with my seniors, Luo Zhaojing, Zheng Kaiping, Xie Zhongle, Wang Ji, Ji Xin, and my fortune to grow with my peers, Cai Shaofeng, Feng Piaopiao.

I would also like to say thanks to my thesis committee members, Chee Yong Chan and Yong Meng Teo, for their valuable feedback on this thesis.

Doing PhD is hard. I still remember the tough time when I started my journey. Every day buried under hundreds of papers, it is easy to become frustrated at my research direction and then the desperation looms. Things turn around in my third year when my first publication wins the VLDB 2019 Best Paper Award. This award boosts my confidence and convinces me that my previous efforts pay off. And immediately next year, I, as the first author, published on the top-tiered database conference SIGMOD 2020, a moment hard to imagine at the beginning. Beyond the technical knowledge and research skill, this is the most valuable lesson from my PhD experience: no achievements accomplish at one stroke but they will definitely spur with long accumulation. And I would like to share this wisdom of life with all my dear readers of this thesis, especially to those junior PhD candidates.

Last but not the least, I never forget the continuous support from my family. For my parents, sorry for being even grumpier than normal, especially during my early PhD days. For my wife, Junna, I love you more than you can imagine. This thesis is dedicated to you and our bright future.

Contents

Abstract

On Databasifying Blockchains

by

RUAN PINGCHENG

Doctor of Philosophy in Computer Science

National University of Singapore

The success of Bitcoin brings enormous interest to its underneath technology, the blockchain. A blockchain is a decentralized system capable to settle disputes between mutually distrusted parties. Throughout the years, blockchain applications are mostly restricted to cryptocurrencies, without fully unleashing their potential. It is until the emergence of smart contracts then blockchains start their transformation from simple cryptocurrency platforms into general data processing systems. Unfortunately, most blockchains researches are still carried out in the security community. Only a few database researchers are aware of this trend. In this thesis, we focus on the enhancement and optimization of blockchains from the perspective of a data system. As an attempt to databasify blockchains, we not only demonstrate the vast opportunities in this area but also appeal to more system researchers.

First, we treat a blockchain also as a generic distributed system, and as such it shares some similarities with distributed database systems. Existing works that compare blockchains and distributed database systems focus mainly on high-level properties, such as security and throughput. They stop short of showing how the underlying design choices contribute to the overall differences. Our paper is to fill this important gap. To be particular, We perform a twin study of blockchains and distributed database systems as two types of transactional systems. We propose a taxonomy that helps illustrate their similarities and differences. The taxonomy is along four dimensions: replication, concurrency, storage, and sharding. We discuss how the design choices have been driven by the system's goals: the blockchain's goal is security, whereas the distributed database's goal is performance. We then conduct an extensive performance study on two blockchains, namely Quorum and Hyperledger Fabric, and three distributed databases, namely CockroachDB, TiDB

and etcd. We demonstrate how the different design choices in the four dimensions lead to different performances. And the experimental insight sheds light on our database-styled optimization on blockchains.

Secondly, with a tamper-evident ledger for recording transactions that modify some global states, a blockchain system captures the entire evolution history of the states. The management of that history, also known as data provenance or lineage, has been studied extensively in database systems. However, querying data history in existing blockchains can only be done by replaying all transactions. This approach is applicable to large-scale, offline analysis, but is not suitable for online transaction processing. We hence present *FabricSharp*, a fine-grained, secure and efficient provenance system for blockchains. *FabricSharp* exposes provenance information to smart contracts via simple and elegant interfaces, thereby enabling a new class of blockchain applications whose execution logics depend on provenance information at runtime. *FabricSharp* captures provenance during contract execution, and efficiently stores it in a Merkle tree. *FabricSharp* provides a novel skip list index designed for supporting efficient provenance query processing. We have implemented *FabricSharp* on top of Hyperledger Fabric v2.2 and a blockchain-optimized storage system called ForkBase. Our extensive evaluation of *FabricSharp* demonstrates its benefits to the new class of blockchain applications, its efficient query, and its small storage overhead.

Thirdly, catering for emerging business requirements, a new architecture called execute-order-validate has been proposed in Hyperledger Fabric to support parallel transactions and improve the blockchain's throughput. However, this new architecture might render many invalid transactions when serializing them. This problem is further exaggerated as the block formation rate is inherently limited due to other factors besides data processing, such as cryptography and consensus. In this work, we propose a novel method to enhance the execute-order-validate architecture, by reducing invalid transactions to improve the throughput of blockchains. Our method is inspired by state-of-the-art optimistic concurrency control techniques in modern database systems. In contrast to existing blockchains that adopt database's preventive approaches which might abort serializable transactions, our method is theoretically more fine-grained. Specifically, unserializable transactions are aborted before ordering and the remaining transactions are guaranteed to be serializable.

For evaluation, we implement our method on top of our *FabricSharp*, and *FastFabricSharp* on top of *FastFabric*. We compare the performance of *FabricSharp* with carefully-chosen baselines. The results demonstrate that *FabricSharp* achieves 25% higher throughput compared to the other systems in nearly all experimental scenarios. Moreover, the *FastFabricSharp*'s improvement over *FastFabric* is up to 66%.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Blockchain Overview

Blockchains shake the industry, academia, and the entire world with storms. The swing of the butterfly that initiates the storm is an unidentified hacker named Satoshi Nakamoto, who authored the Bitcoin whitepaper in 2008 [72]. His proposal makes the breakthrough by employing Proof-of-work (PoW) mechanism, which allows mutually distrusting parties to reach an agreement on the ledger. The ledger records the forever-appending monetary transactions, with the immutability guarantee. Along with other cryptographic techniques, such as the asymmetric encryption, the Merkle index, and the hashed chain structure, Bitcoin is the first-ever practical cryptocurrency that operates under a pure peer-to-peer network, without any central authority. And it immediately follows a series of alt-coins variants [94]. Bitcoin is ground-breaking, as no early design can reach such scalable Byzantine consensus while defying Sybil Attacks. Nakamoto overcomes it by relying on the built-in cryptocurrency to regulate the participant behavior via the economic incentive.

To further unleash the power of blockchains beyond the cryptocurrency, there are two distinct directions. On the one hand, researchers preserve the incentive-based consensus to resolve the anonymity in the open setting. But it extends the system functionality from simple monetary flow into arbitrary data transformation, powered by smart contracts. A typical example is Ethereum, which allows to encode Turing-complete logic and execute it on an embedded virtual machine. We refer to this class of blockchains, featured with incentive-based consensus, built-in cryptocurrencies and the unauthenticated setup as *permissionless blockchains.*

On the other hand, to cater for applications where authenticity and auditability are already mandated, blockchain designers take advantage of their close membership, and turn for more efficient and established state-machine replication [86] for the consensus. In addition, without the built-in cryptocurrencies, the smart contracts of these blockchains are more oriented towards their specific domains, such as Corda [44] for the financial sector and Hyperledger Fabric [8] for the enterprise. We refer to the above class of blockchains *permissioned*. Permissioned blockchains shows more potential to disrupt the industry and attract more interest from entrepreneurs.

Despite the above differences, both classes of blockchains share the identical high-level architecture, as proposed in BLOCKBENCH [27] and illustrated in Figure **??**. The architecture is layered into four. Enumerating from the top, they are the application, consensus, execution, and data model layer. A typical processing pipeline for a generic blockchain constitutes of the following procedures. The consensus layer continuously drives participants to reach an agreement on the block at the ledger tip. Each participant then invoke the contract to mutate the state, based on the context in each transaction in the block. This step is conducted at the execution layer. If a transaction conforms to the blockchain protocol, the participant then persists its effect in the data model layer. And the top application layer hides all the underneath processing details but leaves interfaces to accept the request and query the ledger.

1.2 Vision, Motivation and Principle

Our system-wide optimization primarily focuses on permissioned blockchains. When compared with permissionless blockchains, permissioned blockchains resembles more to distributed databases and hence are more applicable to the database techniques. Their similarities and implication are summarized as follows:

Generic Workload Support. Smart contracts of permissioned blockchains support arbitrary data transformation, like stored procedures in databases. And both of their invocation result into a transaction in their respective context. This is in contrast with permissionless blockchains, which mostly restrict their attention to the ownership transfer. Inevitably, permissioned blockchains raises for more challenges due to their generality. Our optimization can no longer exploit the strong notion of asset ownership like in permissionless blockchains.

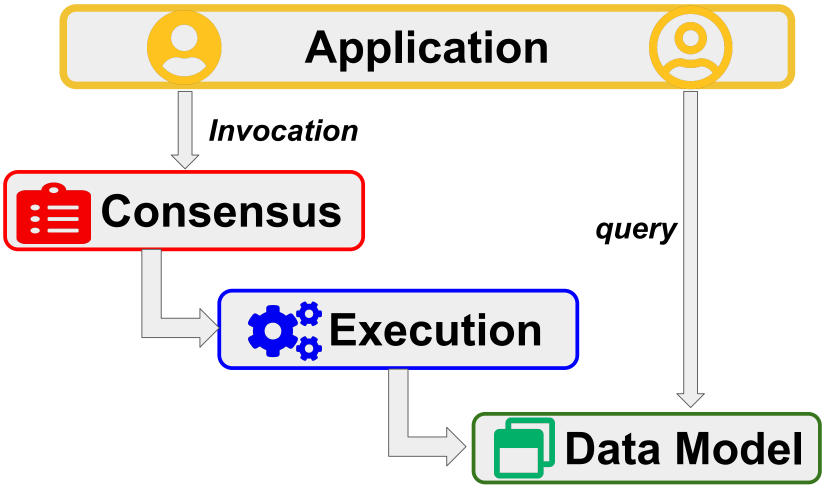

Figure 1.1: Blockchain high-level architecture.

State-mutating transactions must undergo the consensus and the execution components before persisting their effects at the data model layer, whereas ledger and state queries are directly answered by the storage component.

Authenticated Identity. Analogous to distributed databases, permissioned blockchains operate under the authenticated setup. Hence both categories of systems allow for more efficient state-machine replication consensus to withstand the byzantine failure. In comparison to permissionless blockchains, their PoW-like consensus (which is a must to mitigate Sybil Attacks) dominates the entire system performance. From this angle, the enhancement on other components of permissioned blockchains is necessary and worthwhile, as to side with their faster state-machine replication approach. Furthermore, the known membership relieves us from the identity problem. So that we will never get plagued by any Denial-of-the-service Attacks or Sybil Attacks throughout.

The above commonalities explain why a number of entrepreneurs are actively exploring permissioned blockchains to replace their enterprise-ready databases. They seek to harness their common data processing capability while enjoying the additional decentralization and security that permissioned blockchains uniquely provide. However, their attempts are primarily hindered by the following limitations of the permissioned blockchains:

Utility. Even though smart contracts open us opportunities for arbitrary transaction logic for blockchains, their provided utility is far from comparable to that of databases. For example, mainstream blockchains only provide procedural languages to encode the data transformation, such as Ethereum with Solidity and Hyperledger Fabric with Golang. But current relational databases already adopt the more expressive declarative SQL language, not to mention the enriched query features that databases develop over decades. In the face of growing demand, permissioned blockchains call for more data processing functionalities, like databases.

Performance. Another challenge is the low processing volume of permissioned blockchain to accommodate the business load. Researchers in BLOCKBENCH evaluated three blockchains with database workloads on the same testbed. Their results show that blockchains lag far behind databases in around two magnitudes. We believe the extra security properties of blockchains shall not solely account for such a huge performance gap. There must exist abundant optimization room available for the speedup.

In the above, we lay out the optimization vision by showing vast similarities between permissioned blockchain and distributed databases. And we have also motivated such necessity by pinpointing the pain points for the adoption of blockchains in the industry. We now explain the three principles that our enhancement follows:

- We break no security properties. We believe security lies at the core of blockchains. Although relaxing security assumptions is a standard engineering approach for the performance speedup, we do not find it scientific. A typical approach is to improve the system throughput by simply switching from byzantine tolerant consensus to crash failure tolerant. In our thesis, this principle can be manifested in the following two ways. Firstly, for the utility enhancement, we must preserve security on the added features. For instance, in Chapter 4, we take special care to extend the tamper-evidence guarantee to the data provenance. On the other hand, any proposed procedures must be accompanied by their security analysis. This is why we dedicate a subsection in Chapter 5 on the security implication of the transaction reordering.

- We adopt the modularized approach. We decouple a complex system into individual layers for the separate optimization. Modularization allows for the separation of concerns for the ease of reasoning. It also facilitates interoperability with independent optimization. We follow the proposed architecture in BLOCKBENCH for our blockchain optimization in Chapter 4 and 5. In particular, we pinpoint our instrumentation with respect to each of four layers, as classified in Figure 1.1.

- Rather than building from the scratch, we ground our optimizations on Hyperledger Fabric v2.2, the most popular permissioned blockchain, for the demonstration. Building on an existing system not only reuses its well-examined components, but this action by itself proves our practicality. Moreover, the results from a full-fledged system, instead of a prototype, are more convincing. And the evaluation is more meaningful when directly comparing with the vanilla baseline.

1.3 Optimization Basis

We incrementally apply our optimization on Hyperledger Fabric 2.2.0 into *FabricSharp* and open source it [34]. We fork its codebase with the commit hash *2821cf*, the last commit on the branch *release-2.2* when we start preparing this thesis. In later paragraphs, Fabric, without any version specification, all refers to this codebase snapshot.

We now evaluate its throughput under its off-the-shelf configuration, which serves as the baseline. Unless otherwise mentioned, all our following experiments in this thesis are conducted in the same testbed as this experiment. The testbed consists of a local cluster of 16 nodes. Each node is equipped with E5-1650 3.5GHz CPU, 32GB RAM, and 2TB hard disk. The nodes are interconnected via 1Gbps Ethernet.

We present the primary results, after averaging over 3 times on the four-peer setup, in Figure 1.3. The first workload consists of no-op transactions that access no records. Fabric keeps around only 2300 tps throughput when the number of transactions per block is 2000 and beyond. In the second workload with 2000 transactions per block, we gradually increase the request skewness in the modification workload, in which each transaction reads and updates a single record out of 10k. The request distribution follows Zipfian distribution, where the larger coefficient θ implies for more skewness. When θ reaches 1.0, we observe its throughput reduces to 549 tps, 65% of that when $\theta = 0$. And more than half of transactions in the ledger do

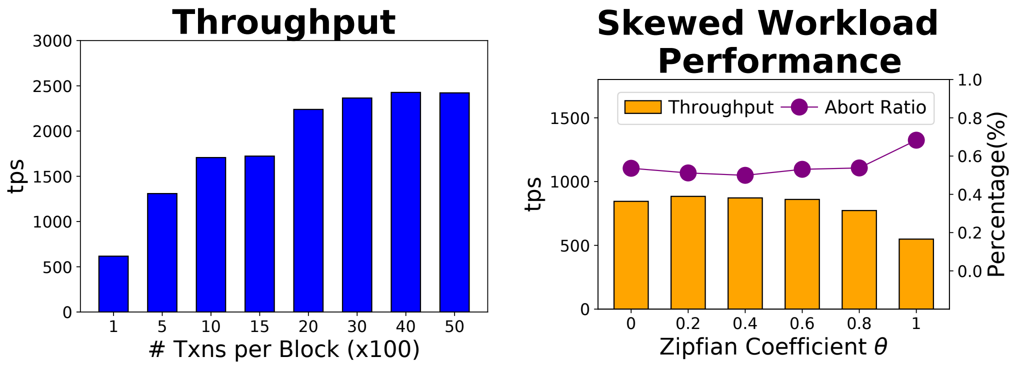

Figure 1.3: Primary evaluation results for Fabric.

not take effect due to the transactional conflicts. This primary experiment further motivates for the performance speedup of permissioned blockchains.

1.4 Thesis Synopsis

We structure the rest of this thesis as follows. Chapter [**ch:literature**] overviews the recent year progress on both permissioned and permissionless blockchains. The covered literature spans from the database, distributed computing, and security communities. For the modularized fashion, we organize their reviews according to the classified layers in Figure 1.1. This chapter ends with our critical analysis in this area.

Chapter 3 proposes a taxonomy that unifies blockchains and distributed databases. The study considers both systems as the same type of distributed transactional systems, for the joint analysis on their respective focus. According to each dimension in the taxonomy, we devise corresponding workloads for the evaluation. Our results reveal the implication of their design choices. This comprehensive study sheds light on the optimization opportunities on permissioned blockchain with database techniques.

Chapter 4 demonstrates our optimization for the utility. We first explain the added business value when data provenance is exposed to smart contracts. Then we introduce how lineage information in blockchains are captured, stored, and queried. The greatest contribution of our proposal is to extend the integrity property for the entire data evolution history. After implementing it into *FabricSharp*, the empirical evaluation demonstrates the negligible performance and storage overhead with this additional feature.

Chapter 5 demonstrates another optimization on the performance. The work originates from our following subtle observation: the execute-order-validate architecture in permissioned blockchains may over-abort transactions under the Serializable isolation level. Borrowed from the well-established transactional analysis from databases, we reason about the potential of transaction reordering to streamline the execution schedule. Based on the developed insight, then we adapt *FabricSharp* to attain the theoretical limits. And the improvement is empirically demonstrated with the remarkable speedup, compared with the vanilla Fabric and the state-of-the-art.

At last, we wrap up the thesis with the conclusion and future directions in Chapter 6.

Chapter 2

Literature Review

In this chapter, we lay out the foundation of blockchains and explore a systematic exposition on their recent progress. We organize the review based on the abstract layers as classified in Figure 1.1, before identifying the research gap.

2.1 Data Model Layer

The data model in blochchains concerns on how to organize data that reflect the latest states, and model the ledger that record the historical transactions.

2.1.1 State Organization

There are two state organizations in the blockchains, the unspent transaction output-based (UTXO) and the account-based. Their key difference lies whether systems explicitly maintain the states. we illustrate both schemes in Figure 2.1.

UTXO Model. UTXO purely operates on the transaction basis. In their structure, a transaction consists of multiple inputs and outputs. Each output is associated with an amount of cryptocurrency and an unanswered cryptographic puzzle. Any future transaction can reclaim this amount in its input, by providing the puzzle answer and referencing the previously unspent output. An canonical puzzle and its solution can be an address in the format of a public key hash, and a digital signature from the corresponding private key. Notably, the UTXO model does not bookkeep the balance to addresses. All transactions in the ledger form a Direct Acyclic Graph that records the cryptocurrency flow, where identity hides under the anonymous addresses.

Bitcoins, due to its decentralization and anonymity, provide a terrain for financial crimes, such as drug dealing and the money laundry. There are a number of attempts

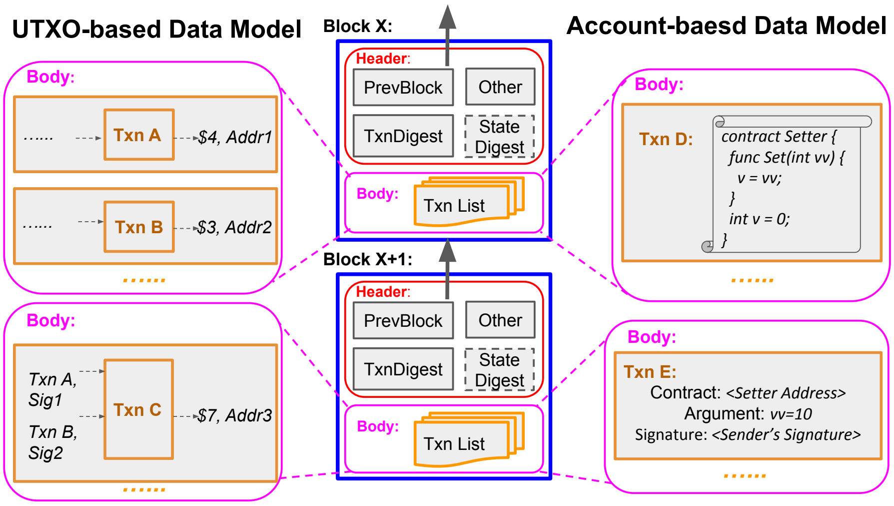

Figure 2.1: Unspent Transaction Output (UTXO) and the Account-based data model.

to exploit the transactional graph, and identify the pattern for the detection [36, 81, 93]. Some analysis relies on the graph linkage to discover the identity [75, 37, 71] or other information to predict the bitcoin price [42].

Account-based Model. Despite its simplicity, the UTXO model is solely applicable to cryptocurrency-based platforms. To support more general workloads, Ethereum introduces the smart contract to encode Turing-complete logic. In their design, a transaction either takes in the form of a contract deployment with the executable code. Or a transaction provides the execution context to invoke a contract. In all cases, each transaction is tagged with the digital signature of the sender. In addition, each blockchain peer must explicitly compute for contract states, including the cryptocurrency balance, in each account. Such requirement is enforced by the blockchain protocol that all peers shall reach consensus on a post-execution state digest in the block header. In contrast, the UTXO-based blockchain only requires for a digest on the transaction integrity.

The account-based data model with smart contracts transition a blockchain into a more general processing platform. But it inevitably incurs more vulnerability from the additional complexity. For example, some malicious users might run an infinite loop in a transaction to waste system resources. To defer such Denial-of-service Attack in the permissionless setting, all blockchains are designed with an incentive-compatible mechanism to prevent the abusive usage. For example, Ethereum charges transaction senders with the transaction fee, in a n amount proportional to the number and the complexity of contract operations [95]. Despite this, computational-heavy transactions may still render blockchains securely-flawed: some researchers reveal that rational block validators tend to skip their execution to gain an edge for the next block mining [65]. It is because all the transaction fee is credited to the block miner.

2.1.2 Ledger Abstraction

The term *blockchain* originates from the fact that the ledger takes in the form of a hash chain of blocks. Initially, the rationale for Bitcoin to batch transactions into blocks is to amortize the cryptographic overhead. However, this single-chain structure prohibits the concurrency, as all participants must sync up on the single ledger tip. To address this problem, numerous studies have transformed a chain into a direct-acyclic graph (DAG), and then derive a total order of transactions [90, 51, 59, 91]. The exact ledger format is strongly tied to their consensus.

Meanwhile, batching trades off the latency for the throughput, as PoW bounds the block interval by the network delay. But from the perspective of permissioned blockchains. such tradeoff is not worthwhile: the state-machine replication consensus places no such restriction on the block interval. This observation accounts for why some researchers abandon the canonical block-based design but directly work upon transactions [48]. For a similar reason, we have observed a number of proposals known for their transaction-based DAG-typed ledger [58, 20, 28].

2.2 Execution Layer

The execution layer concerns on how to process transactions. Different blockchain platforms adopt their distinct execution platforms, such as the Docker environment for Hyperledger Fabric and Ethereum Virtual Machine (EVM) for Quorum. Despite their implementation differences, the execution architecture of any blockchains falls into either of the following categories. Figure 2.2 presents their distinction.

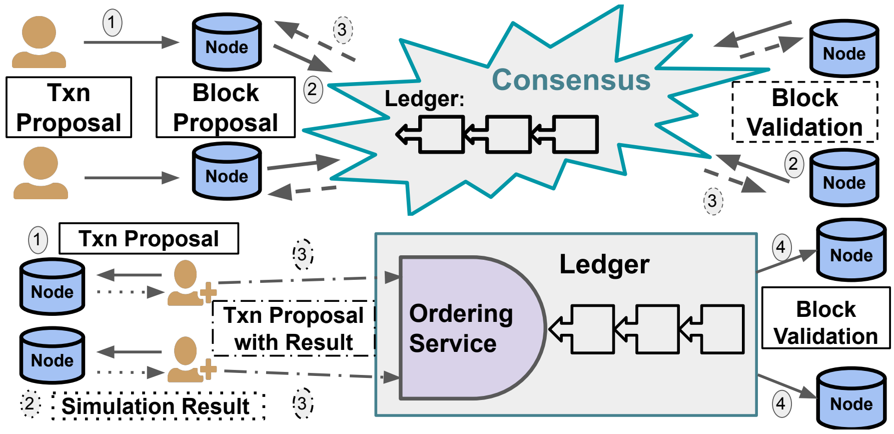

Figure 2.2: Comparison of Order-execute and Execute-order-validate architecture.

2.2.1 Order-execute Architecture

In the Order-execute architecture, each peer serially executes transactions, based on the established order according to the ledger by the consensus. Bitcoin as well as all cryptocurrency-based blockchain adopts this concurrency-free design, simply because the consensus, rather than the execution layer, decides on the system performance. Moreover, sequentiality makes it easy to reason about the system behavior.

Despite this, things still get convoluted when transactions deal with Turing-complete logic. For example, many security researchers have demonstrated that Solidity contracts in Ethereum are far more tricky than expected [62, 76, 10]. Due to some subtle misunderstanding on operation semantics on EVM, flawed contracts can be exploited by adversaries to gain profits. And the problem is further exaggerated given the transparency and the irreversibility of blockchains. The DAO hack shows that such an attack is not only a possibility in the theory but a true threat in reality [83]. In the meantime, there come along a series of empirical guidelines and practical tools to aid the contract development [30, 49, 12, 92].

Through the Order-execute architecture takes on the sequential approach, it is never insulated from the concurrency topic. For example, after remarking the reminiscence between contract bugs in blockchains and data races in shared-memory programming, researchers propose a novel viewpoint on the security issues of contracts, from the concurrency perspective in distributed computing [46, 87]. In [26], researchers explore to tentatively execute transactions in parallel and then fall into serial if encountering a conflict. Instead of speeding up the entire system, the goal of the concurrency is to facilitate the block validation, so that validators can gain a competitive edge on the next block mining. Quantitative analysis of the transaction graph shows abundant concurrent chances in mainstream blockchains [77, 85].

2.2.2 Execute-order-validate Architecture

Execute-order-validate architecture is proposed in Hyperledger Fabric v1.0 [8]. Rather than taking a monolithic approach, the system is designed with two types of blockchain nodes: peers which execute smart contracts and validate blocks, and orderers which order transactions. A transaction pipeline is divided into three phases. In the Execute phase, a client requests a subset of peers to execute the transaction speculatively. The client collects the results and signatures from peers and sends them to the orderers. In the Order phase, orderers order the transactions and batch them into blocks. For modularity, orderers do not inspect the transaction details. In the Validate phase, each peer pulls blocks from the orderers and independently validates each block before persisting the results. The block validation process firstly verifies whether transactions satisfy the endorsement policy, i.e., enough number of peers show the endorsement by their signature. Then validation procedure checks for conflicts in the read/write sets for each transaction. The invalid transactions will not persist their effects, even though they are part of the ledger. Read-only queries only involve the Execute phase.

This architecture brings additional benefits compared to Order-execute architecture. Firstly, the endorsement policy decouples the trust condition of a contract from the consensus. For example, a valid transaction may carry the execution results from only one of three peers. In contrast, the Order-execute architecture mandates the majority of peers to agree on the contract result. Secondly, it preserves confidentiality by restricting the execution to specify peers. Clients, aware of the results before transactions are effected, can also minimize the uncertainty. Lastly, speculative execution at the start fits well for the concurrency. It greatly facilitates computation-heavy transactions, which would queue up in Order-execute architecture.

However, such concurrency comes with the cost, which manifests as the aborted transactions for the serializability. We have empirically demonstrated that in Figure 1.3 and will elaborate this issue in Chapter 5. In light of this, Fabric++ reorders transactions during the Order phase to minimize the abort [88]. OXII architecture is featured for an additional dependency resolution phase at the start [6]. So that it enables for a concurrency-friendly transaction schedule. OXII relies on the core assumption that the dependency can be extracted by inspecting the contract codebase. In the same spirit, XOX architecture runs a patch-up code to streamline a contended transaction [41]. The dependency captured during the transaction execution determines this snippet of code.

2.3 Consensus Layer

Byzantine-tolerant consensus differentiates blockchains from other distributed systems. Proof-of-work (PoW), initially proposed in Bitcoin, opens up a new horizon on the study of incentive-based protocols. Meanwhile, the permissioned blockchains revitalize the interest of state machine replication approach to address the arbitrary failure.

2.3.1 PoW and its Analysis

The essence of PoW is a hard-to-solve and easy-to-verify puzzle. The protocol prescribes that the first solver has the privilege to broadcast a new block on the tip of the ledger. By adjusting the puzzle difficulty such that the solving duration exceeds the maximum network delay, all blockchain participants can then sync up on the unique chain. In Bitcoin, the puzzle solution is a nonce to make the block hash value prefixed with enough zeros. The block proposer gets compensated for the hashing power with the newly-minted cryptocurrencies and the transaction fee.

In the context of Bitcoin, the solution-finding process is called *mining*. Due to the unpredictability of the hash function, mining is fair to all participants, including the adversaries. Ruling out the possibility that adversaries control the majority of the hash power, it leads to the following two implications. If the adversaries pour all their resources to mine on a shorter chain fork, that fork can catch up the longest chain with negligible probability, where the rest of honest power concentrates. Nakamoto has bounded this possibility to be exponentially small in the original whitepaper [72]. More in-depth mathematical frameworks have been established to reason the mining behavior [39, 50, 78, 69, 38]. On the other hand, honest block proposers, aware that the longest chain is hardly revertible, are incentivized to extend on it. It is because their proposed block contains a coinbase transaction that credits proposers to the minted cryptocurrencies. Only the block is in the longest chain then this transaction can take into effect.

Inevitably, the longest-chain-rule may lead to two divergent forks during the asynchronous network period. When the network delay exceeds the block interval, two honest participants may generate two different blocks but on the same height. The longest-chain-rule will eventually resolve to a unique chain when the network partition heals, Hence, transactions in the ledger may not be secure given that they might reside on a shorter to-be-pruned fork. Considering this, clients are advised to wait for their transactions until deep enough, before considering them committed. Intuitively, this depth, quantified by the number of blocks behind, balances between latency and security. In the meantime, the block interval and the block size controls the tradeoff between throughput and security. The shorter interval and the larger block imply greater system capacity. But it compromises the security, i.e., the network may not propagate a block to each participant before the next block is generated.

Moreover, [33] suggests that adversaries may not be incentivized to follow the default mining policy, i.e., mining on the longest chain and broadcast the block immediately. Instead, they may temporally withhold mined blocks and selectively publish them to gain extra profits. Since then, more mining policies are discovered which allows adversaries to gain a competitive edge [73, 21], even infiltrating other mining pools to waste their resources [64, 31]. Researchers employ a Markov Decision Process to find the optimal mining policy and explore the performance-security tradeoff [39, 84].

2.3.2 The Enhancements on PoW

PoW has long been plagued for its energy assumption. In 2019, Bitcoin's electricity consumption is reported to reach 45.8 TWh. Naturally, several researchers propose to transform PoW to be more eco-friendly, or at least dedicate resources to useful tasks. We refer to these PoW-based enhancements as Proof-of-X (PoX).

The major challenge of PoX is to replace the original computation-intensive puzzle and preserve its long-to-solve and easy-to-verify nature. For example, Proof-of-Stake (PoS) requires miners to stake a certain amount of cryptocurrencies [55]. The stake will be confiscated if the proposed blocks are found to be invalid. One can find the adoption of PoS in the following blockchains [52, 24, 15] and their security analysis in [74, 60, 16]. Proof-of-Elapsed-Time (PoET) replies on the Trusted Execution Environment to attest block validators that miners have waited enough time before the next block proposal [18]. To fully exploit the consumed resources to useful works, Proof-of-Retrievability is repurposed for the data archival [68], Proof-of-Prime-Number for the prime number searching [54], and others for matrix product problems [89].

Another enhancement direction is to optimize the primitive PoW. Researchers adapt the puzzle-computation to resist ASIC-equipped mining and deter power centralization [101, 19]. On the other hand, they refine the PoW mechanism to increase the system capacity. For example, the original PoW groups the leader election and transaction proposal together. Bitcoin-NG decouples these two tasks, by allowing a puzzle solver to continuously propose transactions until the next solver emerges [32]. GHOST incentives miners to extend on the heaviest sub-tree instead of the longest chain [90]. This is to better recycle orphaned blocks. Conflux furthers replies on GHOST to determine the pivot chain and use the pivot chain to decide on a direct-acyclic graph, whose topological order determines the overall transaction sequence [59]. The OHIE is renowned for its multiple-chain structure to harness the distributed network setup [98]. Its protocol drives each chain to catch up at the same height.

2.3.3 Byzantine Fault-tolerant Consensus

The state-machine replication method to address byzantine consensus resurges with the popularity of blockchains. The Byzantine-fault tolerant consensus problem involves a number of nodes with initial divergent proposals. The problem requires honest nodes to reach agreement on one of the proposals, while reserving the safety (no possibility of disagreement) and liveness (guarantee of termination). And this must hold true under the premise that a fraction of malicious nodes may perform arbitrary action. Compared to Bitcoin, this problem modeling is more general, i.e., it is not targeting only on the ledger and no incentives can be hinged on.

Long before Bitcoin, PBFT has provided the first-ever practical approach. It achieves the consensus in $O(n^2)$ message complexity where n is the number of nodes. Quorum adopts a PBFT-variant, IBFT as one of its consensus options but still requires for $O(n^2)$ messages [82]. Tendermint reduces this bound to $O(n)$ but forgoes the network responsiveness [17]. In a word, unlike PBFT, the progress of Tendermint is dependent on the maximum network delay, rather than the actual message transmission delay [7]. Hotstuff augments on the two-phase approach of PBFT and Tendermint into three phases of the message exchange [97]. But it achieves both network linearity ($O(n)$ message complexity) and the responsiveness.

While the above series of researches are rooted from PBFT, researchers in VMWare address the problem from a new angle. They remark the possibility to reach consensus in a single phase, given more number of nodes have acknowledged the proposal than normal. Based on this insight, they starts from Fast Byzantine Paxos [67] and Zyzzyva [43], and leads into a series of re-analysis [2] and re-design [3, 43]. Their novelty comes from a fast path if more-than-normal acknowlegements have been collected before a timeout. Otherwise, the consensus falls backs to the standard two-phase pipeline.

FLP Impossibility Result imposes no deterministic, safe and live protocols under the asynchronous network [35]. The above protocols trade the liveness for the safety during the network partition. In other words, disagreement never occurs even the network loses the connection. Correspondingly, there also exist a number of approaches that rely on the network synchrony for the safety [4, 1]. In this manner, these protocols behave like PoW in Bitcoin, i.e., a chain will fork under the asynchronous period. In their design, the strict network assumption comes into play when to reliably broadcast the latest, majority-acknowledged block proposal. Notably, Flexible BFT mixes two types of deterministic protocols together, so that clients can draw their own commit decisions at their discretion on the network condition [66].

Randomness is another alternative route to break the Impossibilty, in order to reach agreement on the ledger. For example, HonestBadgerBFT combines a variety of randomized agreement protocols [70]. So that it achieves linear message complexity, as PoW in Bitcoin. As the carry-on work, BEAT provides a diversity of randomized protocols that tailors for various application demands [29]. Such randomization takes in the form of the unpredictable network connectivity in [80]. Each node repetitively queries for the proposals the neighbors that it knows of. It then determines his own proposal from the majority of query results and uses it to respond for other's queries. Eventually, all honest nodes are probabilistically steered towards a uniform outcome, even though a limited fraction may cheat.

2.3.4 Committee-based Consensus

Blockchains with the committee-based consensus select a single or multiple representative committees, and then to establish the ledger with the above state-machine replication consensus. This idea has been dated back to PeerCensus [25]. But their application may not be that straightforward. It is because protocols require for strong assumptions, including identity management, member constitution, and so on. To cater for these conditions, most blockchains must couple wth some additional procedures, especially on the fair committee formation. For example, Algorand determines a single committee based on the Verifiable Random Function (VRF) [40]. In particular, nodes must compute a value with VRF from its private key and a common seed. Only those with specific values are allowed to join the consensus with their public keys. VRF remains unpredictable without knowing the common seed beforehand. By starting from an unbiased random seed, Algorand guarantees that no adversaries may dominate a committee. Omniledger depends on the Randhound to achieve the decentralized and unbiased randomness, and power the multi-committee establishment [57]. Byzcoin repurposes PoW to determine the membership for a single committee [56]. PoW not only guarantees the randomness but also resists the Sybil attacks under the permissionless setting. RapidChain also uses PoW to determine the participants, from which it organizes into multiple committees with a theoretical-involved method [100]. Elastico directly uses PoW nonce to assign puzzle solvers into one of the committees [63]. Such randomness for the committee formation comes from Trusted Execution Environment in AHL [23]. Not that all the above systems are subject to the periodic committee re-configuration between epochs. This prevents adaptive adversaries to exploit the setup and infiltrate honest nodes. This reconfiguration can be a full swap like in Elastico [63], Algorand [40], and Omniledger [57], or in a rotational manner like Byzcoin [56], RapidChain [100], and AHL [23].

We also observe several blockchains adopt a non-technical solution on membership management. For instance, a central authority decides on the validators' role in RSCoin [22]. The participants and their weight in Libra [14] and EoS [96] are pre-determined according to their initial investment in the project. Chainspace completely sidesteps the committee formation problem [13]. It leaves the behavior of malicious committee audible in the hope that they can be enforced by an external entity.

2.4 Application

2.4.1 Industrial Adoption

Casper

2.4.2 Benchmarks and Surveys

2.5 Summary

Chapter 3

Twin Study

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi

blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Chapter 4

Provenance

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi

blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Chapter 5

Txn

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi

blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Chapter 6

Conclusion and Future Directions

6.1 Conclusion

This thesis focuses on the design and optimization on permissioned blockchains with database techniques. In light of the growing demand on blockchains as emerging transaction platforms, our mission is to identify their bottlenecks and pain points for the improvement while preserving the security. This leads to a series of three works, the first as a survey-like investigation, the second as a utility enhancement, and the last as the performance speedup. Throughout, we adopt the modularized methodology and ground all our implementation into *FabricSharp*.

Firstly, we presented a comprehensive dichotomy between blockchains and distributed databases, viewing them as two different types of transactional distributed systems. We proposed a taxonomy consisting of four design dimensions: replication, concurrency, storage, and sharding. Using this taxonomy, we discussed how both system types make different design choices driven by their high-level goals (e.g., security for blockchains, and performance for databases). We then performed a quantitative performance comparison using five different systems covering a large area of the design space. Our results illustrated the effects of different design choices to the overall performance. Our work provides the first framework to explore future database-blockchain design fusions.

In second work, we showed how to build a fine-grained, secure and efficient provenance system on top of blockchains. We implemented our techniques into *FabricSharp*. The system efficiently captures provenance information during runtime and stores it in secure storage. It exposes simple APIs to smart contracts, which enables a new class of provenance-dependent blockchain applications. Provenance queries are efficient in *FabricSharp*, thanks to a novel skip list index. We benchmarked it against several baselines. The results show the benefits of *FabricSharp* in supporting rich, provenance-dependent applications. They demonstrate that provenance queries are efficient and that the system incurs small storage and performance overhead.

Last but not the least, we proposed a novel solution to efficiently reduce the transaction abort rate in execute-order-validate blockchains by applying transactional analysis from optimistic-concurrency-control databases. We first draw theoretical parallelism between both blockchains and databases. Then, we introduced a fine-grained concurrency control method and implemented it in *FabricSharp* and *FastFabricSharp* based on Fabric and FastFabric, respectively. Our experimental analysis shows that both *FabricSharp* and *FastFabricSharp* outperform other blockchain systems, including the vanilla Fabric, Fabric++, and FastFabric. Unlike databases that achieve high throughput, the blockchains' limited throughput due to factors related to security opens up opportunities for precise transaction management.

6.2 Future Directions

6.2.1 Blockchain Interoperability

While a growing number of blockchains proliferate, most of them operate in silos, with poor synchronization and coordination. Such fragmentation of the landscape not only results into a waste of resources and data isolation, but it also runs counter to the very essence of the Internet, openness and freedom. Even though we observe a number of research works with the special emphasis on the across-ledger token swap, their scope is mostly restricted to the cryptocurrency domain [45, 79, 99]. For wider applications, the community of blockchains should look forward to some more generic standards, just like TCP/IP to the Internet. Promisingly, we notice that Interledger Protocol has taken on the initial attempt [47]. And we expect more will follow.

6.2.2 Declarative Language for Smart Contracts

Even though we demonstrate the provenance support on blockchains in Chapter 4, it still follows an imperative approach. To be specific, users are required to explicitly program *how-to-do*, instead of implicitly declaring *what-to-do* in the smart contract. The high-level declarative language can not only allow users to work on a high abstraction level, saving efforts. It also opens up a vast room for low-level tailored optimization. We haven't observed any progress along with this direction. But according to the development roadmap of the database over the decades, we believe that such an easy-to-use and intuitive usage scheme is essential for the mass adoption of blockchains.

6.2.3 Blockchain-like Verifiable Databases

The impact of blockchains comes from its revolutionary decentralization. But in reality, their byzantine tolerant consensus proves to be an overkill for most applications. In light of this, there are a growing number of secure databases, which, unlike blockchains, completely eliminate distributed setup [9, 102]. Despite the single point, these databases still support verifiability on the state storage. Some vendors simulate a ledger-structure to expose the data provenance with the integrity guarantee [5]. We believe such blockchain-like verifiable databases are capable enough to satisfy most business applications, in which blockchains would prove redundant.

6.2.4 Federated Learning on Blockchains

Considering their common decentralized nature of the federated learning and blockchains, it is not hard to image a number of literatures that pair both hot topics together [61, 53, 11]. The researchers have attempted to rely on blockchains to consolidate data from mutual distrusted users and collectively train for a shared model. In their design, a blockchain serves a trust-building platform to regulate on data ownership and the model copyright. But the challenge still comes in the way, as data privacy may be at odds with the blockchain transparency. And it remains a open topic how to properly allocate the model ownership according to the heterogeneous data sources. In the AI-driven future with the immense adoption of Internet-of-Things, we expect more such interdisciplinary proposals between blockchains and machine learning.

Bibliography

[1] I. Abraham, S. Devadas, D. Dolev, K. Nayak, and L. Ren, "Efficient synchronous byzantine consensus", *arXiv preprint arXiv:1704.02397*, 2017.

[2] I. Abraham, G. Gueta, D. Malkhi, L. Alvisi, R. Kotla, and J.-P. Martin, "Revisiting fast practical byzantine fault tolerance", *arXiv preprint arXiv:1712.01367*, 2017.

[3] I. Abraham, G. Gueta, D. Malkhi, and J.-P. Martin, "Revisiting fast practical byzantine fault tolerance: Thelma, velma, and zelma", *arXiv preprint arXiv:1801.10022*, 2018.

[4] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and M. Yin, "Sync hotstuff: Simple and practical synchronous state machine replication", in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, pp. 106–118.

[5] "Amazon quantum ledger database", `https://aws.amazon.com/qldb/`, Accessed: 2020-09-2.

[6] M. J. Amiri, D. Agrawal, and A. El Abbadi, "Parblockchain: Leveraging transaction parallelism in permissioned blockchain systems", in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2019, pp. 1337–1347.

[7] Y. Amoussou-Guenou, A. Del Pozzo, M. Potop-Butucaru, and S. Tucci-Piergiovanni, "Dissecting tendermint", in *International Conference on Networked Systems*, Springer, 2019, pp. 166–182.

[8] E. Androulaki, A. Barger, V. Bortnikov, C. Cachin, K. Christidis, A. De Caro, D. Enyeart, C. Ferris, G. Laventman, Y. Manevich, *et al.*, "Hyperledger fabric: A distributed operating system for permissioned blockchains", in *Proceedings of the thirteenth EuroSys conference*, 2018, pp. 1–15.

[9] A. Arasu, K. Eguro, R. Kaushik, D. Kossmann, P. Meng, V. Pandey, and R. Ramamurthy, “Concerto: A high concurrency key-value store with integrity”, in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 251–266.

[10] N. Atzei, M. Bartoletti, and T. Cimoli, “A survey of attacks on ethereum smart contracts (sok)”, in *International conference on principles of security and trust*, Springer, 2017, pp. 164–186.

[11] S. Awan, F. Li, B. Luo, and M. Liu, “Poster: A reliable and accountable privacy-preserving federated learning framework using the blockchain”, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2561–2563.

[12] X. Bai, Z. Cheng, Z. Duan, and K. Hu, “Formal modeling and verification of smart contracts”, in *Proceedings of the 2018 7th International Conference on Software and Computer Applications*, 2018, pp. 322–326.

[13] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, “Chainspace: A sharded smart contracts platform”, *arXiv preprint arXiv:1708.03778*, 2017.

[14] M. Baudet, A. Ching, A. Chursin, G. Danezis, F. Garillot, Z. Li, D. Malkhi, O. Naor, D. Perelman, and A. Sonnino, “State machine replication in the libra blockchain”, *The Libra Assn., Tech. Rep*, 2019.

[15] I. Bentov, R. Pass, and E. Shi, “Snow white: Provably secure proofs of stake.” *IACR Cryptol. ePrint Arch.*, vol. 2016, p. 919, 2016.

[16] J. Brown-Cohen, A. Narayanan, A. Psomas, and S. M. Weinberg, “Formal barriers to longest-chain proof-of-stake protocols”, in *Proceedings of the 2019 ACM Conference on Economics and Computation*, 2019, pp. 459–473.

[17] E. Buchman, “Tendermint: Byzantine fault tolerance in the age of blockchains”, PhD thesis, 2016.

[18] L. Chen, L. Xu, N. Shah, Z. Gao, Y. Lu, and W. Shi, “On security analysis of proof-of-elapsed-time (poet)”, in *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, Springer, 2017, pp. 282–297.

[19] H. Cho, “Asic-resistance of multi-hash proof-of-work mechanisms for blockchain consensus protocols”, *IEEE Access*, vol. 6, pp. 66 210–66 222, 2018.

[20] A. Churyumov, “Byteball: A decentralized system for storage and transfer of value”, *URL https://byteball. org/Byteball. pdf*, 2016.

[21] N. T. Courtois and L. Bahack, “On subversive miner strategies and block withholding attack in bitcoin digital currency”, *arXiv preprint arXiv:1402.1718*, 2014.

[22] G. Danezis and S. Meiklejohn, “Centrally banked cryptocurrencies”, *arXiv preprint arXiv:1505.06895*, 2015.

[23] H. Dang, T. T. A. Dinh, D. Loghin, E.-C. Chang, Q. Lin, and B. C. Ooi, “Towards scaling blockchain systems via sharding”, in *Proceedings of the 2019 international conference on management of data*, 2019, pp. 123–140.

[24] B. David, P. Gaži, A. Kiayias, and A. Russell, “Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake blockchain”, in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2018, pp. 66–98.

[25] C. Decker, J. Seidel, and R. Wattenhofer, “Bitcoin meets strong consistency”, in *Proceedings of the 17th International Conference on Distributed Computing and Networking*, 2016, pp. 1–10.

[26] T. Dickerson, P. Gazzillo, M. Herlihy, and E. Koskinen, “Adding concurrency to smart contracts”, *Distributed Computing*, pp. 1–17, 2019.

[27] T. T. A. Dinh, J. Wang, G. Chen, R. Liu, B. C. Ooi, and K.-L. Tan, “Blockbench: A framework for analyzing private blockchains”, in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1085–1100.

[28] M. Divya and N. B. Biradar, “Iota-next generation block chain”, *International journal of engineering and computer science*, vol. 7, no. 04, pp. 23 823–23 826, 2018.

[29] S. Duan, M. K. Reiter, and H. Zhang, “Beat: Asynchronous bft made practical”, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 2028–2041.

[30] S. Ducasse, H. Rocha, S. Bragagnolo, M. Denker, and C. Francomme, "Open-source tool suite for smart contract analysis", *Blockchain and Web 3.0: Social, Economic, and Technological Challenges*, 2019.

[31] I. Eyal, "The miner's dilemma", in *2015 IEEE Symposium on Security and Privacy*, IEEE, 2015, pp. 89–103.

[32] I. Eyal, A. E. Gencer, E. G. Sirer, and R. Van Renesse, "Bitcoin-ng: A scalable blockchain protocol", in *13th {USENIX} symposium on networked systems design and implementation ({NSDI} 16)*, 2016, pp. 45–59.

[33] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable", in *International conference on financial cryptography and data security*, Springer, 2014, pp. 436–454.

[34] "Fabricsharp", `https://github.com/ooibc88/FabricSharp`, Accessed: 2020-09-2.

[35] M. J. Fischer, N. A. Lynch, and M. S. Paterson, "Impossibility of distributed consensus with one faulty process", *Journal of the ACM (JACM)*, vol. 32, no. 2, pp. 374–382, 1985.

[36] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis", *arXiv preprint arXiv:1502.01657*, 2015.

[37] A. Gaihre, Y. Luo, and H. Liu, "Do bitcoin users really care about anonymity? an analysis of the bitcoin transaction graph", in *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, 2018, pp. 1198–1207.

[38] J. Garay, A. Kiayias, and N. Leonardos, "The bitcoin backbone protocol: Analysis and applications", in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, 2015, pp. 281–310.

[39] A. Gervais, G. O. Karame, K. Wüst, V. Glykantzis, H. Ritzdorf, and S. Capkun, "On the security and performance of proof of work blockchains", in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 3–16.

[40] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, "Algorand: Scaling byzantine agreements for cryptocurrencies", in *Proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 51–68.

[41] C. Gorenflo, L. Golab, and S. Keshav, "Xox fabric: A hybrid approach to blockchain transaction execution", in *2020 IEEE International Conference on Blockchain and Cryptocurrency (ICBC)*, IEEE, 2020, pp. 1–9.

[42] A. Greaves and B. Au, "Using the bitcoin transaction graph to predict the price of bitcoin",, 2015.

[43] G. G. Gueta, I. Abraham, S. Grossman, D. Malkhi, B. Pinkas, M. Reiter, D.-A. Seredinschi, O. Tamir, and A. Tomescu, "Sbft: A scalable and decentralized trust infrastructure", in *2019 49th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, IEEE, 2019, pp. 568–580.

[44] M. Hearn, "Corda: A distributed ledger", *Corda Technical White Paper*, vol. 2016, 2016.

[45] M. Herlihy, "Atomic cross-chain swaps", in *Proceedings of the 2018 ACM symposium on principles of distributed computing*, 2018, pp. 245–254.

[46] M. Herlihy, "Blockchains from a distributed computing perspective", *Communications of the ACM*, vol. 62, no. 2, pp. 78–85, 2019.

[47] "Interledger", `https://interledger.org/`, Accessed: 2020-09-02.

[48] Z. István, A. Sorniotti, and M. Vukolić, "Streamchain: Do blockchains need blocks?" In *Proceedings of the 2nd Workshop on Scalable and Resilient Infrastructures for Distributed Ledgers*, 2018, pp. 1–6.

[49] Y.-w. Jeng, Y.-c. Hsieh, and J.-L. Wu, "Step-by-step guidelines for making smart contract smarter", in *2019 IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA)*, IEEE, 2019, pp. 25–32.

[50] A. Kiayias and G. Panagiotakos, "Speed-security tradeoffs in blockchain protocols." *IACR Cryptol. ePrint Arch.*, vol. 2015, p. 1019, 2015.

[51] A. Kiayias and G. Panagiotakos, "On trees, chains and fast transactions in the blockchain", in *International Conference on Cryptology and Information Security in Latin America*, Springer, 2017, pp. 327–351.

[52] A. Kiayias, A. Russell, B. David, and R. Oliynykov, “Ouroboros: A provably secure proof-of-stake blockchain protocol”, in *Annual International Cryptology Conference*, Springer, 2017, pp. 357–388.

[53] H. Kim, J. Park, M. Bennis, and S.-L. Kim, “Blockchained on-device federated learning”, *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2019.

[54] S. King, “Primecoin: Cryptocurrency with prime number proof-of-work”, *July 7th*, vol. 1, no. 6, 2013.

[55] S. King and S. Nadal, “Ppcoin: Peer-to-peer crypto-currency with proof-of-stake”, *self-published paper, August*, vol. 19, p. 1, 2012.

[56] E. K. Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, “Enhancing bitcoin security and performance with strong consistency via collective signing”, in *25th usenix security symposium (usenix security 16)*, 2016, pp. 279–296.

[57] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, “Omniledger: A secure, scale-out, decentralized ledger via sharding”, in *2018 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2018, pp. 583–598.

[58] C. LeMahieu, “Nano: A feeless distributed cryptocurrency network”, *Nano [Online resource]. URL: https://nano. org/en/whitepaper (date of access: 24.03. 2018)*, 2018.

[59] C. Li, P. Li, D. Zhou, W. Xu, F. Long, and A. Yao, “Scaling nakamoto consensus to thousands of transactions per second”, *arXiv preprint arXiv:1805.03870*, 2018.

[60] W. Li, S. Andreina, J.-M. Bohli, and G. Karame, “Securing proof-of-stake blockchain protocols”, in *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, Springer, 2017, pp. 297–315.

[61] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, “Blockchain and federated learning for privacy-preserved data sharing in industrial iot”, *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4177–4186, 2019.

[62] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, “Making smart contracts smarter”, in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016, pp. 254–269.

[63] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, “A secure sharding protocol for open blockchains”, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 17–30.

[64] L. Luu, R. Saha, I. Parameshwaran, P. Saxena, and A. Hobor, “On power splitting games in distributed computation: The case of bitcoin pooled mining”, in *2015 IEEE 28th Computer Security Foundations Symposium*, IEEE, 2015, pp. 397–411.

[65] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, “Demystifying incentives in the consensus computer”, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 706–719.

[66] D. Malkhi, K. Nayak, and L. Ren, “Flexible byzantine fault tolerance”, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1041–1053.

[67] J.-P. Martin and L. Alvisi, “Fast byzantine consensus”, *IEEE Transactions on Dependable and Secure Computing*, vol. 3, no. 3, pp. 202–215, 2006.

[68] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, “Permacoin: Repurposing bitcoin work for data preservation”, in *2014 IEEE Symposium on Security and Privacy*, IEEE, 2014, pp. 475–490.

[69] A. Miller and J. J. LaViola Jr, “Anonymous byzantine consensus from moderately-hard puzzles: A model for bitcoin”, *University of Central Florida Tech. Report CS-TR-14-01 (accessed 5 June 2019) https://socrates1024. s3. amazonaws. com/consensus. pdf*, 2014.

[70] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, “The honey badger of bft protocols”, in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 31–42.

[71] M. Moser, “Anonymity of bitcoin transactions”,, 2013.

[72] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system”, Manubot, Tech. Rep., 2019.

[73] K. Nayak, S. Kumar, A. Miller, and E. Shi, “Stubborn mining: Generalizing selfish mining and combining with an eclipse attack”, in *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, 2016, pp. 305–320.

[74] C. T. Nguyen, D. T. Hoang, D. N. Nguyen, D. Niyato, H. T. Nguyen, and E. Dutkiewicz, “Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications and opportunities”, *IEEE Access*, vol. 7, pp. 85 727–85 745, 2019.

[75] M. Ober, S. Katzenbeisser, and K. Hamacher, “Structure and anonymity of the bitcoin transaction graph”, *Future internet*, vol. 5, no. 2, pp. 237–250, 2013.

[76] R. M. Parizi, A. Dehghantanha, *et al.*, “Smart contract programming languages on blockchains: An empirical evaluation of usability and security”, in *International Conference on Blockchain*, Springer, 2018, pp. 75–91.

[77] D. Reijsbergen and T. T. A. Dinh, “On exploiting transaction concurrency to speed up blockchains”, 2020. arXiv: `2003.06128 [cs.DC]`.

[78] L. Ren, “Analysis of nakamoto consensus.” *IACR Cryptol. ePrint Arch.*, vol. 2019, p. 943, 2019.

[79] P. Robinson, D. Hyland-Wood, R. Saltini, S. Johnson, and J. Brainard, “Atomic crosschain transactions for ethereum private sidechains”, *arXiv preprint arXiv:1904.12079*, 2019.

[80] T. Rocket, M. Yin, K. Sekniqi, R. van Renesse, and E. G. Sirer, “Scalable and probabilistic leaderless bft consensus through metastability”, *arXiv preprint arXiv:1906.08936*, 2019.

[81] D. Ron and A. Shamir, “Quantitative analysis of the full bitcoin transaction graph”, in *International Conference on Financial Cryptography and Data Security*, Springer, 2013, pp. 6–24.

[82] R. Saltini and D. Hyland-Wood, “Correctness analysis of ibft”, *arXiv preprint arXiv:1901.07160*, 2019.

[83] F. Santos and V. Kostakis, “The dao: A million dollar lesson in blockchain governance”, *School of Business and Governance, Ragnar Nurkse Department of Innovation and Governance*, 2018.

[84] A. Sapirshtein, Y. Sompolinsky, and A. Zohar, “Optimal selfish mining strategies in bitcoin”, in *International Conference on Financial Cryptography and Data Security*, Springer, 2016, pp. 515–532.

[85] V. Saraph and M. Herlihy, “An empirical study of speculative concurrency in ethereum smart contracts”, *arXiv preprint arXiv:1901.01376*, 2019.

[86] F. B. Schneider, “Implementing fault-tolerant services using the state machine approach: A tutorial”, *ACM Computing Surveys (CSUR)*, vol. 22, no. 4, pp. 299–319, 1990.

[87] I. Sergey and A. Hobor, “A concurrent perspective on smart contracts”, in *International Conference on Financial Cryptography and Data Security*, Springer, 2017, pp. 478–493.

[88] A. Sharma, F. M. Schuhknecht, D. Agrawal, and J. Dittrich, “Blurring the lines between blockchains and database systems: The case of hyperledger fabric”, in *Proceedings of the 2019 International Conference on Management of Data*, 2019, pp. 105–122.

[89] A. Shoker, “Sustainable blockchain through proof of exercise”, in *2017 IEEE 16th International Symposium on Network Computing and Applications (NCA)*, IEEE, 2017, pp. 1–9.

[90] Y. Sompolinsky and A. Zohar, “Secure high-rate transaction processing in bitcoin”, in *International Conference on Financial Cryptography and Data Security*, Springer, 2015, pp. 507–527.

[91] G. Srivastava, A. D. Dwivedi, and R. Singh, “Phantom protocol as the new crypto-democracy”, in *IFIP International Conference on Computer Information Systems and Industrial Management*, Springer, 2018, pp. 499–509.

[92] S. Tikhomirov, E. Voskresenskaya, I. Ivanitskiy, R. Takhaviev, E. Marchenko, and Y. Alexandrov, “Smartcheck: Static analysis of ethereum smart contracts”, in *Proceedings of the 1st International Workshop on Emerging Trends in Software Engineering for Blockchain*, 2018, pp. 9–16.

[93] M. Weber, G. Domeniconi, J. Chen, D. K. I. Weidele, C. Bellei, T. Robinson, and C. E. Leiserson, “Anti-money laundering in bitcoin: Experimenting with graph convolutional networks for financial forensics”, *arXiv preprint arXiv:1908.02591*, 2019.

[94] Wikipedia, “List of cryptocurrencies — Wikipedia, the free encyclopedia”, `http://en.wikipedia.org/w/index.php?title=List%20of%20cryptocurrencies&oldid=975908929`, [Online; accessed 01-September-2020], 2020.

[95] G. Wood *et al.*, “Ethereum: A secure decentralised generalised transaction ledger”, *Ethereum project yellow paper*, vol. 151, no. 2014, pp. 1–32, 2014.

[96] B. Xu, D. Luthra, Z. Cole, and N. Blakely, “Eos: An architectural, performance, and economic analysis”, *Retrieved June*, vol. 11, p. 2019, 2018.

[97] M. Yin, D. Malkhi, M. K. Reiter, G. G. Gueta, and I. Abraham, “Hotstuff: Bft consensus with linearity and responsiveness”, in *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, 2019, pp. 347–356.

[98] H. Yu, I. Nikolić, R. Hou, and P. Saxena, “Ohie: Blockchain scaling made simple”, in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, 2020, pp. 90–105.

[99] V. Zakhary, D. Agrawal, and A. E. Abbadi, “Atomic commitment across blockchains”, *arXiv preprint arXiv:1905.02847*, 2019.

[100] M. Zamani, M. Movahedi, and M. Raykova, “Rapidchain: Scaling blockchain via full sharding”, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 931–948.

[101] A. R. Zamanov, V. A. Erokhin, and P. S. Fedotov, “Asic-resistant hash functions”, in *2018 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, IEEE, 2018, pp. 394–396.

[102] M. Zhang, Z. Xie, C. Yue, and Z. Zhong, “Spitz: A verifiable database system”, *arXiv preprint arXiv:2008.09268*, 2020.