A CRYPTOCURRENCY INCENTIVIZED VOLUNTARY GRID COMPUTING PLATFORM FOR DNA READ ALIGNMENT

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

By
Halil İbrahim Özercan
September 2019

A CRYPTOCURRENCY INCENTIVIZED VOLUNTARY GRID COMPUTING PLATFORM FOR DNA READ ALIGNMENT
By Halil İbrahim Özercan
September 2019

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Can Alkan (Advisor)

Çiğdem Gündüz Demir

Aybar Can Acar

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

A CRYPTOCURRENCY INCENTIVIZED VOLUNTARY GRID COMPUTING PLATFORM FOR DNA READ ALIGNMENT

Halil İbrahim Özercan
M.S. in Computer Engineering
Advisor: Can Alkan
September 2019

The main computational bottleneck of High Throughput Sequencing (HTS) data analysis is to map the reads to a reference genome, for which clusters are typically used. However, building clusters large enough to handle hundreds of petabytes of data is infeasible. Additionally, the reference genome is also periodically updated to fix errors and include newly sequenced insertions, therefore in many large scale genome projects the reads are realigned to the new reference. Therefore, we need to explore volunteer grid computing technologies to help ameliorate the need for large clusters. However, since the computational demands of HTS read mapping is substantial, and the turnaround of analysis should be fast, we also need a method to motivate volunteers to dedicate their computational resources. For this purpose, we propose to merge distributed read mapping techniques with the popular blockchain technology. Cryptocurrencies such as Bitcoin calculate a value (called nonce) to ensure new block (i.e., "money") creations are limited and difficult in the system, however, this calculation serves no other practical purpose. Our solution (Coinami) introduces a new cryptocurrency called Halocoin, which rewards scientific work with alternative minting. In Coinami, read alignment problems are published and distributed in a decentralized manner while volunteers are rewarded for their work. Authorities have two main tasks in our system: 1) inject new problem sets (i.e., "alignment problems") into the system, and 2) check for the validity of the results to prevent counterfeit.

Keywords: Read Mapping, Blockchain, Cryptocurrency, Grid computing, incentivization.

ÖZET

DNA DİZİ HİZALAMASI İÇİN KRİPTOPARA TEŞVİKLİ GÖNÜLLÜ DAĞITIMLI HESAPLAMA PLATFORMU

Halil İbrahim Özercan
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Danışmanı: Can Alkan
Eylül 2019

Yüksek çıktılı dizilemeye dair veri analizlerinde en temel sayısal sıkıntı okumaların referans genoma haritalanmasıdır ve bu iş için genelde bilgisayar kümeleri kullanılmaktadır. Lakin, petabaytlara varan bu veriyi işleyecek büyüklükte bilgisayar kümeleri kurmak imkansıza yakındır. Ayrıca, hataları gidermek ve yeni dizilenmiş insersiyonları eklemek için referans genom periyodik olarak güncellenmekte olup, bu durum bir çok büyük projede okumaların tekrar haritalanmasına ihtiyaç doğurmaktadır. Bu yüzden, programlama kümelerine olan ihtiyacı azaltmak için gönüllü örgü programlama teknolojilerini araştırmamız gerekmektedir. Ancak, okuma haritalama probleminin hesaplama gereksinimleri hatrı sayılır olması, analizin geri dönüşünün hızlı olması gerekliklerinden dolayı gönüllüleri motive edecek bir yönteme ihtiyaç duymaktayız. Bu sebepten dolayı, dağıtık okuma haritalama tekniklerini popüler blokzincir teknolojisi ile buluşturmayı öneriyoruz. Bitcoin gibi kripto paralar yeni blok üretimini zorlaştırmak ve kontrolde tutmak için "nonce" adı verilen bir değeri hesaplamayı mecburi kılar ancak bu sürecin başka herhangi bir kullanışlı amacı yoktur. Bizim çözümümüz (Coinami) yeni bir kripto para birimi olarak Halocoin'i sunuyor ve bu kripto para birimi bilimsel çalışmaları ödüllendirmeyi alternatif bir para basma yöntemi olarak sunuyor. Coinami, okuma haritalama problemlerini merkezi olmayan şekilde yayınlayıp dağıtarak gönüllüleri yaptıkları işten dolayı ödüllendirmeyi esas alır. Yetkili kurumlar sistemimizde iki temel şeyden sorumludur; 1) yeni problem kümeleri yayınlamak, 2) sonuçların kontrolünü sağlayıp sahte para basılmasının önüne geçmek.

Anahtar sözcükler: Okuma Haritalama, Blok zincir, Kripto para, Örgü Programlama, teşvik.

Acknowledgement

First and foremost, I would like to extend my sincerest gratitudes to my advisor Prof. Can Alkan for his unearned trust in me and guiding me professionaly and personally for more than 4 years. Although I had my doubts and hesitation through this journey, he treated me with never-ending patience. Besides my advisor, I would like to thank the rest of my thesis committee, Çiğdem Gündüz Demir and Aybar Can Acar for reviewing this thesis. Also, I would like to thank EMBO grant (IG-2521) for sponsoring this thesis.

Next, I would like to thank my fiancée Balanur İçen for always being there for me whenever I needed emotional, mental or any kind of support. She has been my closest partner through this adventure and there are not enough words in my vocabulary to truly express my gratefulness to her.

I also would like to appreciate my friends who have challenged me to improve my methods, way of thinking and perspective on so many levels. My flatmate for two years, Alim Gökkaya, constantly provided crucial feedback during the development of my thesis project. Ezgi Ebren, Zülal Bingöl, Fatih Karaoğlanoğlu and all the other AlkanLab members have always responded to my inquiries with enthusiasm.

After spending two years in Bilkent University, I have joined Opsgenie in the last year of my Master's studies. I need to send my deepest thanks to, foremost, my teammate Taha Kırca who assisted me with utmost understanding to keep following this dream, And Anı Çalık who helped me to improve myself as a Software Engineer and Çağatay Emekçi who had always extended his support beyond his team.

Finally, none of this would have been possible without my loving family. I will never be able to articulate their significance in my life; my mother Hayriye, my father Hasan Basri, my siblings Ayfer, Ali and Hanife and their families mean the world to me. Thanks for your never ending support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Genome is regarded as serialized and compressed bits of life that shapes the concept of an organism. These encoding characters are called "nucleotides" or "bases" which belong to a 4-letter alphabet; "A" for Adenine, "G" for Guanine, "C" for Cytosine and "T" for Thymine. A human genome consists of nearly 3 billion nucleotides. DNA sequencing is the process of correctly determining the order of almost all bases in a DNA strand and cracking the code of life lies behind that. Sanger et al. published their results on sequencing bacteriophage phi X174 in 1977, which is regarded as the first successful genome sequencing [1]. Later on, the idea of decrypting the human genome, mapping all the genes to understand the fundamentals of human biology became more realistic. Human Genome Project was started in 1990 with this goal that would immensely accelerate the further genetic research. The project was completed in 2003 and ever since sequencing technologies evolved very quickly [2], and now they are among the most powerful tools available for biological research. We are now able to read the entire genome of a human individual in a few days for a fraction of costs incurred by previous technologies thanks to High-Throughput Sequencing (HTS) [3, 2]. For example, the Illumina HiSeqX platform can sequence the genomes of approximately 18,000 humans a year, at an estimated cost of $1,000 per genome [4]. This corresponds to about 2 petabytes of data per year, per sequencing center. However, data sets generated by high throughput sequencing platforms require

immense amounts of computational resources to align to reference genomes, and call and annotate genomic variants. This problem is even more pronounced if re-analysis is needed for new versions of reference genomes, which may impose high loads to existing computational infrastructures. Although there are certain distinctive characteristics of mapping reads generated by different platforms, given the popularity of Illumina as today's most popular sequencing platform, the remainder of the paper will assume that the genomes are sequenced using the Illumina technology.

Considering the fact that there are many genome centers that either already have, or will purchase this system, the amount of data generated each year will increase to hundreds of petabytes to exabytes. The computational analyses of such data involves multiple steps, but the main bottleneck is to find the potential locations of short stretches of DNA sequences in a reference genome. This step, called read mapping, usually takes 30 CPU days per human genome [5, 6, 7, 8, 9, 10] (see [11] for a review of aligners). The computational burden of read mapping is monotonically increasing not only because of the growth of data to be analyzed, but also because of updates in the reference genome assemblies. For instance, the human reference genome is updated every 3 to 4 years that fixes assembly mistakes, and adds either new sequences that are found in the genomes of newly sequenced individuals, or "alternative haplotypes" that are frequent variations from the existing reference sequence. Such drastic changes in reference genomes usually necessitate remapping of the existing data to enable more accurate characterization of genomic variants. Thanks to the embarrassingly parallel nature of this problem clusters are typically used. However, building clusters that are large enough to handle hundreds of petabytes of data is not feasible for every research center and university.

Therefore, exploring other practical methods which can leverage non-sequential characteristic of read mapping is highly important. Another way to achieve distributed computing is using heterogeneously separated computer grids. This type of computing allows us to run different tasks on individual computers which are connected by either a local network or the Internet. Although the computation is

scattered among participants in grid computing, it still has a master-slave architecture in a broader sense. This can also be replaced by peer-to-peer networking in theory but collaborative database have not existed until Bitcoin's proposal. Bitcoin, blockchain as the underlying technology, now enables us to improve traditional grid computing in multiple ways such as openness, incentivization and extensiveness. In the remainder of this chapter, we will introduce volunteer grid computing and blockchain which both have established precedents of collaborative work that stems from public contribution.

1.1 Grid Computing

Volunteer grid computing is a joined effort of individuals that resembles a large computing cluster spread across personal computers from all around the world. It was made popular by the Berkeley Open Infrastructure for Network Computing (BOINC) platform, and specifically its Search for Extraterrestrial Intelligence at Home (SETI@home) project [12]. BOINC volunteers choose a scientific problem to work on, download data from the server, solve the problem, and upload results back to the server. To reduce the burden on the volunteers, the BOINC clients can be set to run only when the computer is idle (i.e., "screen saver mode"). There are a number of bioinformatics applications ported to the BOINC platform such as Rosetta@home for protein structure prediction [13], and FiND@home [14] for docking simulations on malaria proteins. Although it is also possible to write HTS read mapping applications for BOINC, such applications must provide data security to prevent both "leak before publication" and genomic privacy, and protection against malicious users (i.e., volunteers that deliberately upload incorrect results). Furthermore, HTS read mapping "assignments" (i.e., the sequence data) continuously grow as the volume of data required to be processed grows. It is obvious that timely analysis requires substantial computational resources (CPU, RAM, disk space, and network bandwidth), which may make it difficult to motivate volunteers of BOINC network, since network bandwidth is a completely different resource than CPU or memory.

1.2 Cryptocurrencies and Blockchain

The cryptocurrencies were first introduced by Wei Dai in 1998 [15], and received considerable attention from the public after the first decentralized cryptocurrency (Bitcoin) was published in 2009 by an unknown person with pseudonym Satoshi Nakamoto [16]. The motivation behind Bitcoin was to create a distributed currency, which is not controlled by any monetary authority. On the footsteps of Bitcoin's success, a plethora of cryptocurrencies were proposed including Ethereum [17], Ripple [18] and Litecoin. Although Ethereum has its own currency called Ether, it can host other decentralized applications through smart contracts which make Ethereum a general purpose, turing-complete, decentralized state machine. Blockchain term quickly replaced "cryptocurrency" in scientific literature since this discovery proved to be practical for Ethereum. Although there are some nuances between different blockchain implementations, most of them are composed of two interconnected processes called Mining and Transaction. In a nutshell, mining refers to the creation of new "digital money" termed coin, that are recorded by a public ledger called blockchain. Mining involves a computationally difficult process called "proof-of-work", while transaction refers to an exchange of assets between blockchain wallets. As of August 2019, Bitcoin mining network's total computation power has reached approximately 70 ExaHash/s [19]. In comparison, world's most powerful scientific computation grid -BOINC- boasts a computation power of 27.45 PetaFLOP/s [20]. Although two units are not directly comparable as hashing uses integer operations while BOINC's power is measured in floating point operations, even the arbitrary yet not too radical scaling of 1 to 1000 (assuming floating point operations are 1000 times "harder" than integer operations) still posits Bitcoin network to be more powerful than the BOINC network. The computation power of the Bitcoin network is solely used to maintain the currency's integrity by ensuring that the block creation is always a difficult task which makes Bitcoin theoretically immutable unless 51% of total computing power is not controlled by a single attacker. The difficulty stems from the proof-of-work, which entails finding a number called *the nonce*. Hardness of nonce computation is recalculated according to total computation power of the network at every 2016 blocks. This arbitrary looking number is actually selected

by thinking that recalculation should execute every two weeks and a block should be generated every 10 minutes. Nonce adjustment effectively limits the amount of blocks that can be generated by the miners, preventing devaluation of the money and also ability of attackers as more blocks are mined. The proof-of-work scheme within Bitcoin and other cryptocurrencies are effective solutions for the security, reliability and robustness of the cryptocurrency systems, yet serve no other practical purpose.

1.3 Contribution of Thesis

Here we propose Coin-Application Mediator Interface (Coinami) where such computation power like Bitcoins' will be used for scientific computation as well as integrity purposes while still incentivizing miners. While high-end CPU and GPU owners will have higher chance mining new blocks, miners with higher broadband connection and storage space will have advantage executing read mapping. We chose to use DNA sequence alignment problem as the alternative proof-of-work in our initial implementation because 1) it is hard to compute and easy to verify 2) has substantially endless supply of work. Coinami is not completely decentralized due to the need for availability and generation of HTS data. Instead, Coinami has a federated structure, where one root authority tracks and validates middle level sub-authority servers that supply HTS data to the system and checks for validity of alignments, and the third level is composed of farmers. The farmers download problem sets from the middle level authorities, map the HTS reads to a reference genome, and send the results back to the middle level authority for verification (Section 3). However, the scientific proof-of-work is decoupled from the rest of the system, making Coinami easily adaptable to other scientific problems that require substantial computational resources.

The rest of this thesis includes the previous work published on read mapping, grid computing, and blockchain technology and how they correlate with computational burden problem. Chapter 3 explains in detail how Coinami functions, federated authority model in blockchain, generalization of Coinami beyond read mapping and novel incentivization method that introduces multiple proof schemes together in a single blockchain. In Chapter 4, results from real world usage of Coinami are presented. Finally, future work and potential alternative means of using Coinami are discussed in Chapter 5.

Chapter 2

Background

Coinami is a framework that aims to bring together unique problems from 3 different areas and provide a single solution. For this purpose, the history and the present condition of the technologies and problems that will be referred in this paper should be thoroughly studied. In this chapter, we are going to share the findings of our study on the mentioned problems.

2.1 Challenges facing Read Mapping

Not long after DNA's complete discovery in 1953 [21], first successful sequencing of a genomic material happened in 1977 by Frederick Sanger [1]. Although it was of a bacteriophage with a single chromosome and 3,569 bases, it was a huge achievement as it laid the foundation for further research. It took another 20 years for necessary advancements in sequencing methods to sequence a bacteria, which had 1.8 million bases [22]. Despite the fact that Human Genome Project (HGP) was completed successfully using Sanger's method, its practicality and cost-efficiency remained questionable. Thankfully, High Throughput Sequencing (HTS) had arrived by 2005 and brought a fierce competition among the companies that provided sequencing solutions. HTS offered HGP levels of sequencing

power for each sequencing center around the world [2]. Since then, time and financial costs of sequencing have dropped down significantly. These improvements transferred the bottleneck from data generation to post analysis.

Most genomic data analysis pipelines start with aligning the raw data to the reference genome which is called read mapping. Mapping is required to assemble a whole genome, discover SNPs or Structural Variants (SV) that may signal a hereditary disease. Alignment algorithms go as far back as 1950s when first protein was sequenced successfully [23]. Needleman and Wunsch proposed the first algorithm which is based on dynamic programming to calculate the ideal alignment of two genomic sequences in 1970 [24]. This algorithm has been also referred to as global alignment due to the fact that it focused on matching sequences in their entirety. Later, Smith and Waterman proposed their local alignment algorithm [25] in 1981 that is able to extract local similarities which may have gone unnoticed after loads of evolutionary mutations.

2.1.1 History

Although genome sequencing has a relatively long history as aforementioned, read mapping is a rather new problem. Of course alignment tools had existed for a while before HGP was completed [26], read mapping became a necessity only after a human reference genome was established as a reference. Researchers had already known that traditional aligment methods would not suffice for human genome that has 3 billion nucleotides but data generation was so slow that it should have not become a problem in practice. However, HTS turned the tide and big data had arrived for bioinformatics. A plethora of tools and algorithms have been published to accelerate read alignment since then. Most of these tools share a common nature called *seed-and-extend*. This method works similarly to how a person would intiutively search for only 2-3 letters in a word search puzzle. If they identify a candidate location that contains letters from the word, they can extend their search in that region. We can categorize these tools into two according to their fundamental idea: 1) Hashing, 2) Burrows-Wheeler Transform

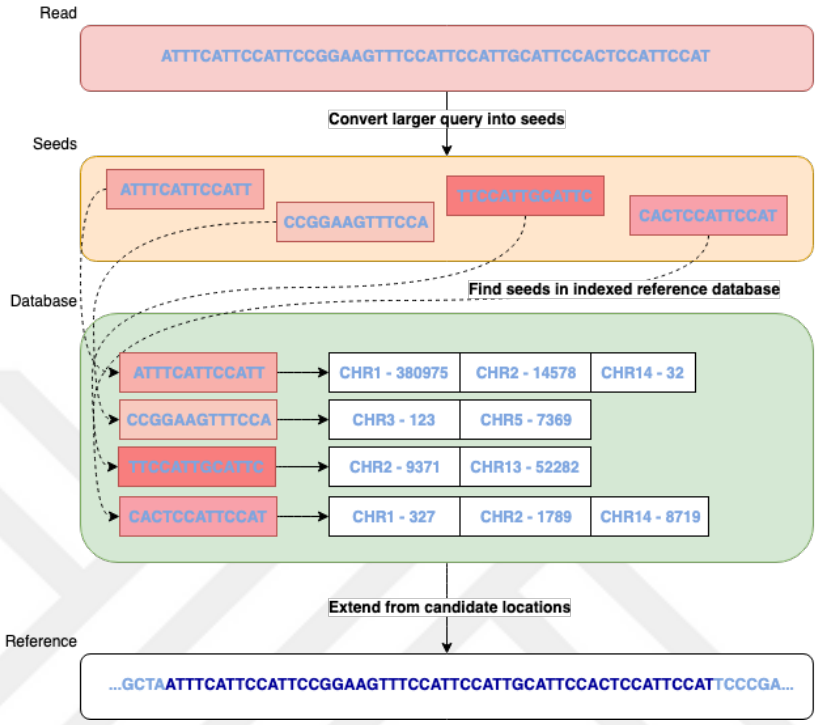

Figure 2.1: General seed-and-extend algorithm flow chart. First the larger query is broken down into seeds. These seeds can be efficiently queried in a database which has been indexed beforehand. Then, we extend from candidate locations using dynamic programming until we find an acceptable match.

(BWT).

Hashing based tools like mrFAST [27], MOSAIK [28], FastHASH [10], NovoAlign [29], Smalt [30] simply use different hash algorithms to index reference and read k-mers. These tools are comparable according to their hashing performance such as collision rate (multiple locations for a single seed) and calculation speed. Before going to extend stage, there should be as few candidate locations as possible. BWT based tools such as Bowtie and BWA use a reverse compression algorithm called Burrows-Wheeler Transform and Ferragina-Manzini (FM) index [31] for fast substring queries. These tools do not suffer from possible hash collisions since they only extract exact seed matches for extension. Although, they require rather large index structures for both FM-index and suffix array representation, it has been seen that large index is a smaller problem compared to speed.

2.1.2 "Gigantic-Massive-Colossal" Data

Big data and cloud computing are two distinct concerns in IT management. Big data refers to abundance of data that is available for statistical analysis. Cloud computing is an infrastructure type in which the computing and storage resources are easily scalable, cheaper and provided by a platform-as-a-service product like Amazon Web Services or Microsoft Azure. They are correlated since big data is a problem where data sources are unorganized, unpredictable, and unmanaged while cloud services can offer organization and management since source and processing units are bundled together.

However, data confidentiality causes a dispute for bioinformatics. Diligence becomes important while handling real genome data. Most countries already have rules concerning the analysis of genomic data [32, 33]. Read mapping on cloud infrastructure as part of a commercial practice is highly critical. Therefore, researchers published several studies on this subject to enable use of cloud for the bottleneck of biological data analysis.

2.1.3 Homomorphic Encryption

Homomorphic encryption is an encryption technique which allows preset calculations on encrypted data which does not require decryption. For example a hypothetical homomorphic encryption technique which allows algebraic addition may be used to determine the total wage of employees in a company without revealing each individual's wage. In the case of genomics, it might be possible to align a completely encrypted genome to reference genome. This operation would be completely safe from side-channel attacks since it would not allow any other operation. Researchers showed a possible way of calculating edit distance with homomorphic encryption [34].

Although homomorphic encryption theoretically offers the best mitigation, the application space is limited. Most importantly, Coinami requires to verify the

solutions. If solutions arrive to servers as encrypted, decryption process would take significant amount of time which might make Coinami worse than current approaches. Moreover, ciphertexts usually are longer than plain text versions. Bandwidth is a significant criterion in privacy trade-off since genomic data is already big enough. Therefore, homomorphic encryption might be the last resort for preserving privacy in Coinami platform.

2.1.4 Hybrid-Cloud Approach

Hybrid-Cloud is an alternative to complete cloud transition. Instead of using cloud service for all of the computation, the most heavy parts are outsourced to cloud but no identifiable information is leaked. Rest of the work that contains identifiers are handled in a private cloud, on-premise servers [35]. For this purpose, currently available read mapping tools would not be sufficient. BALAUR [36] is a hybrid-cloud enabled read mapping tool that uses kmer voting and fingerprinting to identify potential seed-and-extend locations in a secure way. However, authors disregard the potential network cost in their results which would drastically change the overall performance.

2.2 A primer for the blockchain technology and cryptocurrencies

Blockchain, in a broad sense, is a distributed, time-stamped, immutable database, shared and automatically synchronized among all participants [37]. This distributed database technology was first developed to be used as a public ledger in the popular decentralized cryptocurrency, Bitcoin [16]. Although Bitcoin, and other similar cryptocurrencies were introduced as decentralized alternatives to coinage and monetary systems that virtually remained unchanged since the time of the ancient Lydians, they are in fact mere applications of the underlying blockchain technology. Most important aspects of blockchain technology are: 1)

decentralization (i.e., a single entity cannot control the database), 2) immutability (i.e., no past record can be altered), and 3) security (i.e., accounts are protected by enhanced cryptographic methods).

2.2.1 Terminology

Being familiar with blockchain related terms is ought to be useful to understand the algorithms and processes that are discussed in this paper. Below, we provide some definitions for the keywords that are often used while discussing blockchain.

2.2.1.1 Mining

To ensure blockchain is very difficult to manipulate, a sufficient amount of work is expected to be performed to add a new block. The basic requirement of this task is to be difficult to compute but easy to verify. Although it is time and resource consuming, block generation is vital for blockchain to be functional. Therefore, any person who puts resources into this task gets rewarded as incentive. This process is similar to mining precious metals where finding the material is based partially on luck, when found it is trivial to understand if the material is non-fake, more manpower makes it easier to dig larger areas.

2.2.1.2 Proof-of-work

A proof of work is a piece of data which is difficult (costly, time-consuming) to produce but easy for others to verify and which satisfies certain additional requirements. Producing proof-of-work can be a random process with low probability so that a lot of trial and error is required on average before a valid proof of work is generated. It is a must requirement for any block to include proof-of-work to be added to blockchain.

2.2.1.3 Proof-of-stake

Proof-of-stake (PoS) is a consensus algorithm like proof-of-work which guarantees stability, immutability and finality in the blockchain. Major difference is that PoS does not require vast computational power, instead it relies on a set of rules that would severely punish any malicious user. In PoS, there are validators instead of miners. To become a validator, users deposit some amount of their coins into the network. Through a network wide deterministic random selection, a validator gets selected to forge the next block. Higher the stake someone puts in as deposit, higher their chances are to be selected. If a validator behaves malevolent, their stake gets wiped out. More information is provided by Casper Project [38] which is Ethereum's PoS based solution.

2.2.1.4 Proof-of-space

Instead of proving a capability in terms of computation, proof-of-space utilizes storage bound functions. This eliminates large electricity consumption of CPU bounded functions, while increasing the demand for larger space. There is also another approach in which users send files to each other and show a proof that file is stored on the other end.

2.2.1.5 Transaction

Every transaction can be considered as a state transition function. Blockchain starts in an empty state. A transaction moves a piece of data (coins) from an address to new address if 1) sender account has sufficient amount and 2) transaction issuer proves to be real owner of the sender address. If this state-transition-function returns *True*, then transaction is considered to be valid and added to a candidate block by miners. Every block has a dedicated section for transactions. They can be considered as Create, Replace, Update, Delete (CRUD) queries on a database and they are batch executed once block enter the chain.

2.2.1.6 Difficulty level

The difficulty stems from the proof-of-work, which entails finding a number called *nonce*. It cannot be calculated but can be found by trial-and-error. Therefore, higher trial capability increases the chance of finding a valid nonce. Difficulty level is updated at every 2016 blocks according to how long it took to generate the last 2016 blocks, which is around two weeks in the Bitcoin network. This effectively limits the amount of new blocks generated by the miners, preventing devaluation of the money as more blocks are mined.

2.2.1.7 Smart contracts

Briefly, smart contracts are sets of instructions that are enforced when certain conditions are met, and whose authenticity, conditions and necessities can be observed and approved by everyone. A smart contract operates as an autonomous account on the blockchain. It has a dedicated storage to keep details, objects and information related to its application. Transactions that are addressed to a smart contract causes an activation and the contract updates the records depending upon its predefined instructions.

2.2.1.8 Mempool

Mempool is the pool of unconfirmed transactions inside Bitcoin network. When a user makes a transaction, this intent is first propagated through the network to reach to the miner. Only the transactions that are approved by miners are eligible to get into the blockchain. Since miners take small cuts from each transaction they include in their candidate block, it is not beneficial for any miner to turn down a valid transaction. Until transactions are added to the blockchain, they are considered as unconfirmed and reside in the mempool.

2.2.2 Merits of Blockchain

Decentralization is the essential contribution of blockchain to modern consensus agreements like legislation, financial agreements or joint resolution. Most of current approaches require a third party, a governor, to reach and force an agreement between members. This central authority should be trusted by all participants to fulfil arrangement conditions. However, additional measures are required to keep the authority in check in regards to potential abuse of power. These measures are likely to only increase in numbers and introduce many more actors, also known as bureaucracy, that would clutter the system. Blockchain mitigates this trust to an algorithmic process which is approved by all the participants. Every member can inspect the actions of others in a timely and organized fashion, which facilitates reaching a consensus at any given time.

Although it was developed as an integral part of Bitcoin, the blockchain technology itself is loosely coupled to Bitcoin and other cryptocurrencies as described above, making it possible to be used in other cases. Almost all applications of the blockchain technology can be boiled down to two interconnected processes called Mining and Transaction [39].

Transactions are similar to procedure calls. They carry an intent to execute an action on resources that are available on-blockchain. In the case of abstract blockchain technology, transactions refer to a state transition rather than actual exchange of assets [17]. A transaction can include a piece of data to store or execute a term in a contract. This way, blockchain can be more than a central bank ledger but an abstract database of time stamped data.

Mining is more of an abstract concept since its only purpose is to add new blocks to the ongoing chain. There are multiple ways of doing mining which include proof-of-work, proof-of-stake and proof-of-space. The most important requirement behind any proof scheme is that there should be an investment of capital like time, fiat money or hardware. Mining is purposefully slowed down to an average constant time, which is called "dynamic difficulty level" to regulate block generation. The idea behind this slow down is that if a malicious party

intends to break the consensus or alter the blockchain, they need to prove that they put more amount of work than rest of the network in much shorter time. Also, economic integrity of currency is kept in check by adjusting coin generation in cryptocurrencies, similar to limiting banknote production in national mints. Slow-down is achieved by either a computationally difficult process called "proof-of-work" (PoW), or a generic algorithm called "proof-of-stake" (PoS) that relays the blocks by a schedule based on accounts' balance [37]. However, we emphasize that, although Bitcoin was the first successful blockchain implementation, it and other cryptocurrencies remain as merely financial applications of the blockchain technology.

After blockchain technology gained attention thanks to Bitcoin, its broader potential was unleashed by the Ethereum Project proposed in late 2013 [17]. Ethereum revolutionized the blockchain by adding a new type of autonomous account, called Smart Contracts, which enabled custom application development on blockchain. This is achieved by providing a virtual machine which can execute arbitrary set of instructions defined in a "smart contract". Briefly, smart contracts include terms of agreements (i.e., work to be performed, resources to be allocated, etc.), and reward and penalty mechanisms when the agreements are met or unmet, respectively (Figure 2.1). Through this generalization mechanism provided by Ethereum, the need to develop a new blockchain for each different purpose is eliminated. Current possible use cases of Ethereum include decentralized data feeds, cloud computing, prediction markets, and decentralized file storage [40].

As we describe above, Blockchain technology is still in development stage where problems such as scalability [41, 42, 43], performance [44, 45] and security [46, 47] are being addressed. Also, Blockchain utilizes wide range of cryptographic and computational tools to solve trust problems and build consensus. Expertise in cryptology may be required to fully understand the underlying technology, however a number of reviews are available to provide basic information for the general reader [48, 49].

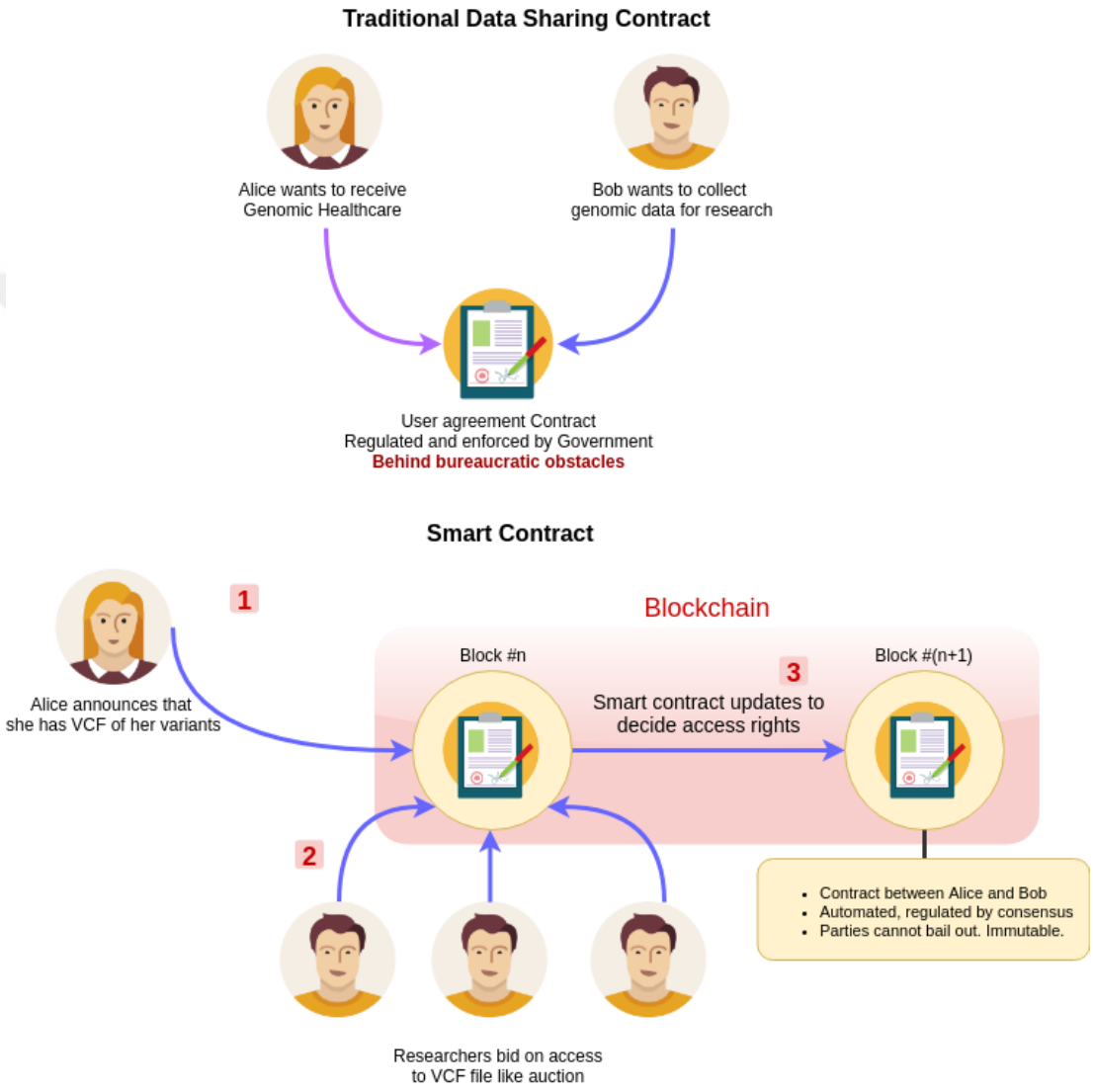

Figure 2.2: A smart contract to manage access management for genomic data. In traditional agreements, both parties sign a contract that dictates the boundaries in which the participants must obey. A third party is often required to enforce the agreement conditions. Smart contracts on the other hand, can eliminate the need of a third party. In this figure, Alice publishes an encrypted version of her file that contains genomic variants (called VCF). At first, no other participant, researcher, can analyze this file. In the second round, smart contract accepts bidding transactions for this file. Highest bidder is then selected to be the rightful owner and gets access to the file through an algorithmic process.

Chapter 3

Coin Application Mediator Interface for meaningful Proof-of-work

Coinami is a project that deals with DNA Read Mapping which consists of generating jobs from multiple FASTQ files, verification and demixing of results. The system also includes a separate, custom blockchain called *Halocoin* which serves as the basis blockchain that is used by Coinami to find farmers and distribute jobs. Implementation and design of Halocoin and Coinami grew together after the inception of the idea that scientific work could be a replacement for proof-of-work in blockchain. However, a generic interface that connects the computational distributed work to blockchain was later deemed necessary and we decided to separate these loosely coupled modules to generalize the Coinami platform. We will provide the motivation, challenges and details behind Coinami and Halocoin in this chapter.

3.1 Motivation

Since its proposal in 2008, Bitcoin proved to be useful in many occasions as a decentralized currency [50, 51]. In spite of its unreliable valuation [52], Bitcoin have had the highest market capitalization in digital currency world with a large margin [53]. Besides its financial success, it has been repeatedly pointed out that the real invention behind what made Bitcoin outstanding was the blockchain at its core. Researchers demonstrated that the Blockchain could be useful in data management, identification, e-voting, genomics, supply chain and many other areas [54, 55, 56].

However, blockchain continued to be a controversial subject since its popularity in finance was taken advantage of by criminals [57]. Its overuse in cases where alternative solutions have already existed also irritated many investors. Blockchain has also been considered as a solution that is trying to find a problem. Besides all this criticism on sociological aspects of Bitcoin and its community, the most reasonable one that has still not been addressed in practice is the environmental impact [58]. As of now, energy consumption of Bitcoin network is more than the total of Switzerland [59]. Although there have been propositions like proof-of-stake and proof-of-space, they never got of the ground.

We know that most scientific projects from physics, chemistry, micro biology and astronomy require immense amount of processing power. BOINC had set of to connect massive processing resources of idle CPUs from all around the world to help scientific communities but its grasp remained relatively small. We believe that only voluntary crediting system is not a strong incentivization model. Instead of providing volunteers with a badge of honor, problem suppliers can reward participants with a virtual currency. It can be again like a centralized credit system but experience shows us that cryptocurrencies are already gaining popularity among technology enthusiasts. To make this currency useful, sub-authorities can provide services like DNA sequencing, genotyping in exchange of this virtual currency.

3.2 Challenges

Coinami is an ambitious idea in the sense that whole framework should work as a single unit while it requires expertise in multiple areas of computer science. We give below some of the debatable decisions that we made during the design of Coinami.

3.2.1 Centralization Risk

Lack of a central bank and a governing unit is what makes cryptocurrencies conspicuous. They offer complete transparency by design. Introducing any kind of centralization would cause reluctance among participants. Nonetheless, even if read mapping can be computed by homomorphic encryption with almost 100% performance yield, data generation is locked by research centers. As long as individuals cannot inject new tasks, Coinami has to remain quasi-centralized. Furthermore, complete fair exchange without a trusted third party has been proven to be impossible [60]. This implies that even though we wish to decentralize Coinami, sub-authorities and farmers will not have a full resolution when a conflict occurs; e.g., non-rewarding a valid solution. We hope that early adopters are going to focus on the scientific benefits of this trade-off.

3.2.2 Data Availability

We believe that data availability would not be a problem in the short or long term. However, technically it is possible that no Sub-Authority would publish a job for some time. In this case, Coinami and Halocoin should continue to function normally. Halocoin achieves this by aforementioned separation of farmers and miners. When rewards drop, mining gets more difficult but it does not halt because it has a ceiling.

3.3 Architecture

Although decentralization makes cryptocurrencies open to public regulation which attracts certain people, other financial systems through history taught us that simplifying and diversifying the ways to earn commodity is much more attractive for regular users. However, most blockchain applications follow the footsteps of Bitcoin by tying minting solely to generation of new blocks also known as mining. Moreover, even transaction fees are collected by the miner of a block. This implies that there are two course of actions a user can take to gain coins if they lack the necessary equipment to regularly mine; either the user makes an off-blockchain exchange with a coin owner or participate in a mining pool. Mining pools are joined efforts of individual miners who would like to decrease their income volatility. They pose as insurance companies in terms of risk sharing. However, other studies have shown that the existence of mining pools increases the energy consumption by five fold [61]. Also, mining pools bring quasi-centralization to a decentralized environment [62].

It is clear that the perfect decentralization would take loads of resources to achieve when we consider the financial, behavioral and algorithmic aspects of the blockchain. Instead, we propose a federated authority model that would raise the attraction from diverse set of users, increase the usage-rate while decreasing the energy consumption and also use that said energy for a meaningful work.

There are three properties of reversing the hash problem that makes it perfect for proof-of-work in Bitcoin. First and most important, it is an NP-Complete problem, hard to solve, easy to verify. Secondly, the necessary information to calculate the criteria for a solution resides completely on-blockchain. This means that problem carries a dynamic nature. One can look at the recent history on blockchain and find out what the problem currently is asking for. Later, other participants can also look at the same history and verify that the problem was correctly solved. Lastly, since the problem is configurable, it has a never ending nature. It can always be repeated by looking at the recent history. When we want to replace this problem with a scientific one that supports all three requirements,

we realize that it is not simple to find a practical and feasible problem. For example, PrimeCoin utilizes prime-finding problem that supports all requirements but its practicality is open for debate since its product is highly fixated.

We chose Read Mapping problem as the proof-of-concept application for Coinami due to its embarrassingly parallel nature. However, an authority such as a sequencing center, university or research center is required to supply new alignment tasks to the system. Nonetheless, introduction of any kind of authority would raise a trust issue among the cryptocurrency users. Luckily, we can benefit from certification-chaining which is a battle-tested architecture that keeps the HTTPS-protocol safe [63]. It is mainly used to check authenticity of web pages. Halocoin offers a similar structure using certificate-chaining to decide who can supply problems to blockchain and who will be assigned to solve these problems, completely on-blockchain.

3.3.1 Federated Authority Model

Halocoin has a three-level federated structure, where one root authority (RA) tracks and validates middle level authority servers that supply HTS data to the system and checks for validity of alignments, and the third level is composed of miners and farmers (Figure 3.1). The RA must be trusted by the entire system to validate only "trustable" authority servers. RA is also hard-coded in Halocoin client. Although it is possible to replace RA by a decentralized autonomous organization (DAO), we leave it as a future work because it requires an in-depth review of DAOs and their applicabilities. In the remainder of this thesis, we refer to middle level authority servers as authorities.

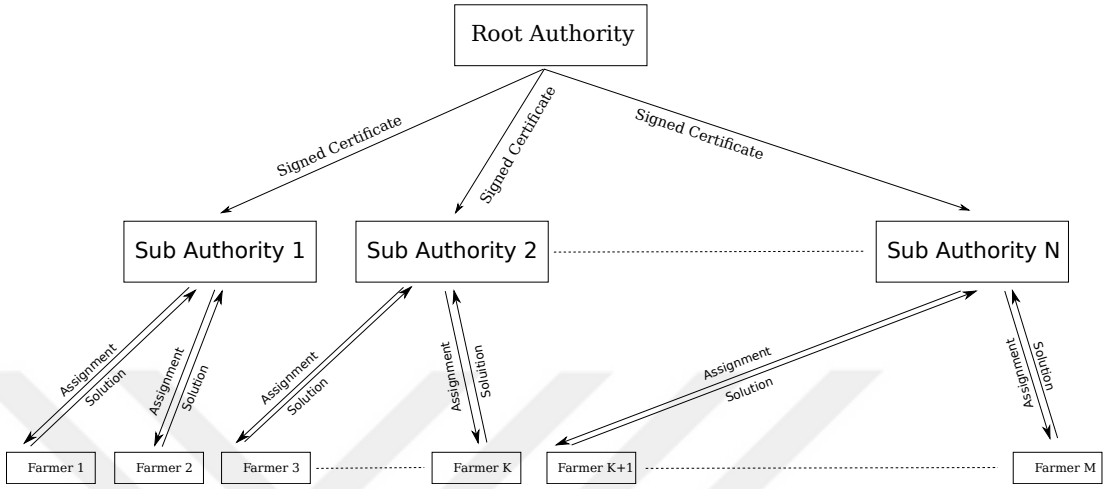

Figure 3.1: Three level structure for the Halocoin network. The single root authority issues certificates to the sub-authorities. The sub-authorities (middle level) sends assignments to multiple farmers, and validate the results they receive. If the alignments are valid, they sign reward transaction and publish it with the network. A farmer can work on assignments from multiple sub-authorities.

3.3.2 Authorities

The root authority assigns certificates to the middle level authorities, validating the authorities and allowing them operate within the Coinami network. The certificates and their corresponding private keys are used in special mint transaction signing. Authorities fulfill two main roles in Coinami: 1) they inject new assignments (i.e., "alignment problems") into the system, and 2) check the validity of the results to prevent counterfeit. If the results are valid, the authority publishes a special transaction called *JobReward* that gives fresh new coins to the farmer. For the validity test, in an earlier version of Coinami [64] we proposed to use decoy reads. However, using decoy reads introduces a new challenge of supplying already aligned read data back to the system. Instead, we developed a new tool called *verifybam* that checks the correctness of each alignment record in the bam file. This method guarantees that each individual read is accurately aligned to the correct position and all the reads are exactly the same between assignment and returned solution. Rest of the Coinami workflow is discussed further in the following sections.

3.3.3 Miners and Farmers

As in Bitcoin, miners are regular users who opt-in to create new blocks by demonstrating proof-of-work which stems from hash solving problem. Their role is crucial since the ledger must grow to accommodate new transactions. Coinami also introduces a new user role which is called *farmer*. Farmers are the ones who opt-in to solve scientific problems such as read-mapping to earn coins. The allegory in their names is that miners need luck to struck the motherload but when they do, their reward is invaluable. On the other hand, farmers need next to no luck because if they keep practicing what they know, their yield is almost guaranteed but also less than what miners can earn. Farmers only need to download sets of problems, follow the instructions to get results and finally upload the results back to authority servers.

3.4 Mixer and Demixer

Coinami has to overcome security and privacy concerns which are considered huge obstacles that are almost always present in bioinformatics applications. Genomics data is the most secret and revealing information of an individual. One might infer many physical attributes, disposition to possible diseases. For example, leaking this kind of information to insurance companies may result in violation of basic human rights. Furthermore, DNA's reach does not end in one's self but also includes relatives since hereditary knowledge can be inferred from it and there are studied attack vectors on kin privacy [65]. Coinami must take extra caution dealing with genomic data due to the fact that blockchain is completely open to public. The most extreme case would be that all farmers come together as a single adversary. Coinami must guarantee the privacy off all individuals even in this situation. We analyzed Homomorphic Encryption and Hybrid-Cloud approach. We presented details in previous chapter. Although Homomorphic Encryption technically checks all the boxes, yet it failed to provide a high performer solution for now. Hybrid Cloud is almost identical to Coinami architecture. However,

those solutions are focused on executing a single read mapping task in which it is difficult to preserve privacy. Coinami can leverage the fact that it can shuffle hundreds or even thousands sets of FASTQ files before publishing them publicly. Hence, we chose to use *anonymous mixing* technique to maintain privacy and performance at the same time. One concern related to mixing was how modern alignment tools would behave when read pairs are shuffled from multiple individuals. Detailed review suggests that scatter/gather kind of alignment results in undeterministic behavior when reads come from repetitive regions of the genome [66]. We altered our verification method to overcome this problem.

3.4.1 Mixer

We developed a new tool called *mixer* to convert initial FASTQ file pairs (paired-end reads) to Coinami compatible jobs. *Mixer* basically takes an input destination directory and opens a file read stream to all FASTQ files. It starts randomly selecting a pair and adding it to ongoing job file. When ongoing job reaches the size limit, *Mixer* continues with another job. Until all reads in all pairs are processed, this procedure repeats. This method does not have any memory footprint which is significant because input files are usually at terabytes scale. Figure 3.2 shows the flow of a *Mixer* run.

Mixing also produces a lookup table since all the read labels are replaced with index numbers during the process to provide anonymity. We emphasize that each additional read data from a different individual would contribute to security and the privacy of overall mixing process. Moreover, mixing also calculates hashsum for each job to later use in verification.

3.4.2 Demixer

Demixer is a straightforward process to merge all received solutions once they are fully verified. *Demixer* loads up all the lookup tables from mixing process

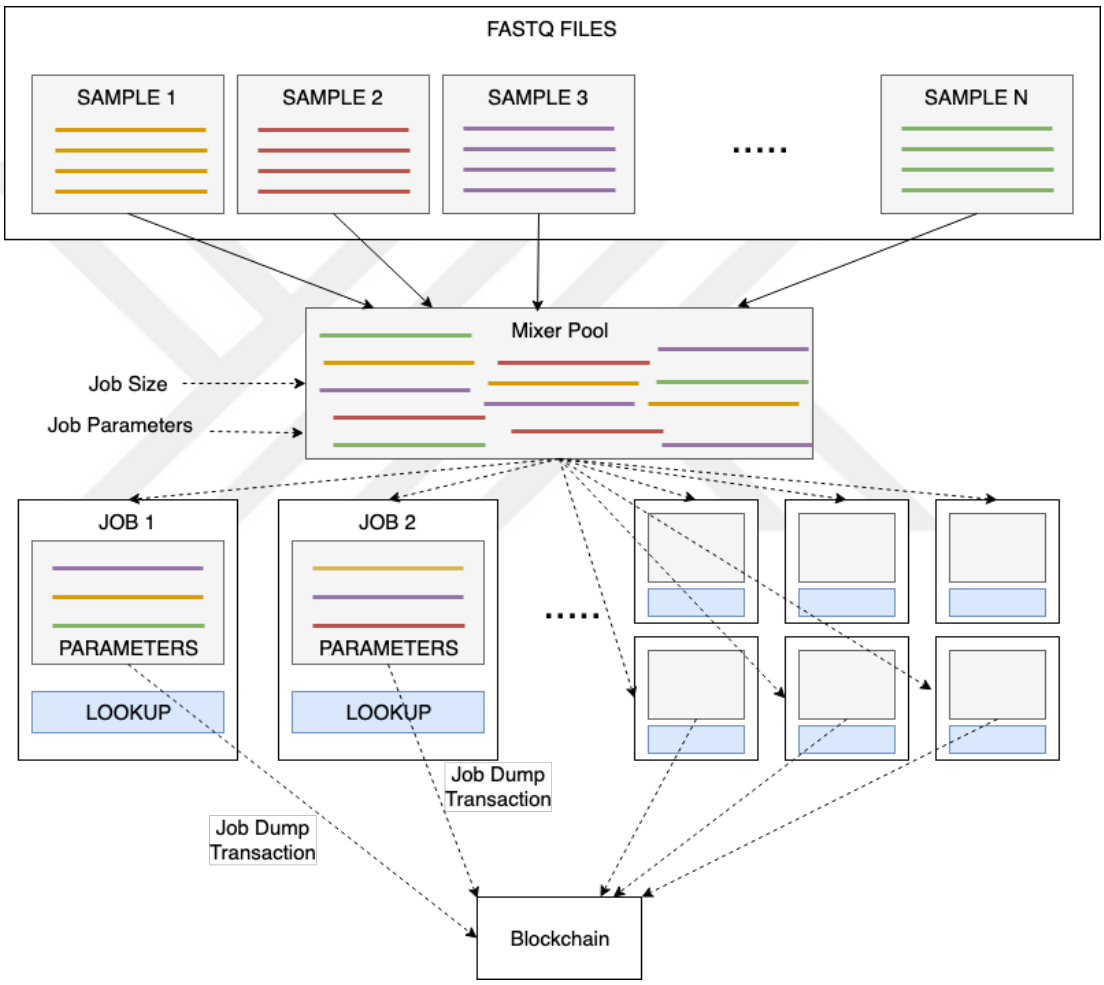

Figure 3.2: Mixing starts with gathering multiple FASTQ pairs in a pool. Then task packages are generated from this pool with respect to arguments like job size and reward coefficient. Finally, packages are separated into two parts called Job and Extras. Job is destinated to send out to farmers. Extras are set aside to be used in verification process. All jobs are published on blockchain using a special transaction called *job-dump*.

into memory and iterates through received BAM files. After fixing each read's label, they are added to their respective final BAM files.

3.5 Verification

Cryptocurrencies require the block generation to be controlled just as minting in fiat currencies. Otherwise, inflation attacks would render it useless. To achieve this, every miner is responsible for showing a proof-of-work that they actually worked the required amount of time on this problem. However, a suitable problem for proof-of-work needs to be open, adjustable, and easy to verify.

First of all, blockchain has no private data other than anonymized identities of participants as wallets. Everything on blockchain is published to everyone. Also, blockchain has no other connection to any other data source. This property is called the Single Source of Truth (SSoT). This brings us to openness of proof-of-work problem. It should be solvable and verifiable by the data that is present on the blockchain. Any data source that is subject to privacy concerns or does not reside on the blockchain cannot be used by proof-of-work algorithms.

Second, contrary to popular belief proof-of-work does not have to be a difficult problem that millions of GPUs are needed to solve. Instead, it needs to be adjustable by the size of the network. Because, while blockchain can be used by the entire world, it can also be used by small number of individuals to keep track of the financial records of a closed-market. In other words, proof-of-work should also scale by the number of nodes. It must take some constant time even there are 10 single core CPU miners or millions of specialized FPGA miners since inflation must remain constant in the system.

Finally, verification of proof-of-work should be trivial at any size. No matter how hard the problem becomes, verification must always remain simple to keep the synchronization of individual nodes stable.

In sight of these requirements, Bitcoin's proof-of-work algorithm that depends on cryptographic hashing is a perfect match. Basically, Bitcoin requires miners to generate a block whose hash value is smaller than some target value that is also calculated from the chain's history. Miners use a random constant called nonce that affects the final hash of the block to vary the solution. Target is also adjustable by looking at how fast or slow the recent blocks are generated. There have been multiple candidates which targeted replacing Hash-based PoW due to its environmental impact. In spite of their technical sufficiency, they had failed to gain any traction. One reason might be losing interest of already invested miners who have purchased mining rigs as investment. As discussed before, Coinami does not replace Bitcoin's PoW but introduces an alternate way of minting. This way, specialized miners can still get returns for their investments while motivated users, farmers, can solve scientific problems and earn virtual currency. Although, verification of a solution is complicated due to the privacy concerns regarding the raw data, actually we show that our concept implementation using read alignment checks all boxes to be integrated as an alternative minting option.

3.5.1 Verifying mapping results with Verifybam

First part of read alignment verification is to check whether the submitted solution is correctly aligned. Currently, there is only ValidateSamFile that reports the validity of a SAM/BAM file with respect to SAM format specification. However, this tool only covers format errors. Coinami needs to check whether the reads are correctly aligned to reference genome. Thus, we developed a new tool called *verifybam* which evaluates every single mapping result in a SAM/BAM file.

Verification of a read alignment is a straightforward process. Every alignment should include CIGAR and MD fields alongside of exact mapping location. These fields define how the read was mapped onto reference genome. Our new tool, verifybam, scans every read record in a SAM/BAM file and applies the modifications according to CIGAR/MD and validates whether they are correctly aligned to the reference genome. However, this solution posits two threats; with the right

CIGAR/MD configuration every read can be aligned anywhere and most of the reads can be discarded as unmapped. Therefore verifybam offers two parameters that puts a threshold on allowed edit distance and lower limit on unmapped reads.

However, verification process should be as fast as possible since default Bitcoin PoW can be verified in microseconds. Since verifybam loads the entire reference genome into memory for every run, reference loading takes approximately 6 seconds even when used with an SSD with 500 MB/sec read speed. Considering the fact that lots of submissions will be processed by the server, reloading reference genome every single time will amount to 10 minutes of excessive disk read for 100 tasks for the best case. Verifybam comes with a threaded-daemon mode which runs in background with reference loaded into memory once. Whenever a task comes through, verifybam daemon receives the request from client node, processes it with specified amount of threads and returns the result to client process. This eliminates the overhead of loading reference genome for each task.

3.5.2 Checking the completeness of work performed

Although verifybam correctly and efficiently validates the result of a task, it does not guarantee that the submission actually matches the task that was assigned to the user. For that purpose, we use a similar approach to Bamhash [67] excluding read names and qualities. While checking every read, verifybam also calculates an aggregating SHA256 hash of all the reads that is present in the SAM/BAM file. Resulting hash must match the hash of the reads in the FASTQ file which was generated during mixing. This also implies that hashsum must also be calculated while mixing. In the end, even if a single base changes, aggregated hash will be completely altered.

Only down-side of this approach is some post-processing tasks like duplication removal cannot be performed on client side since they will cause hashsum to fail. Fortunately, this is not a big concern because tasks include genomes of multiple individuals and duplicate removal would yield faulty outcomes. SHA256 is a

cryptographically safe hashing algorithm which is difficult to crack but it adds significant overhead while running verifybam.

3.6 Halocoin - Abstract Coinami Blockchain

Although read mapping has been the starting point for Coinami, it merely demonstrates the capabilities of the cryptocurrency based parallel task execution. Hence, we developed a new blockchain called Halocoin which acts as the underlying database and consensus protocol for Coinami and many other possible scientific projects. It supports multi-layered authority structure, job distribution and farmer/miner model. In this section, we will explain how Halocoin is capable of running any Map-Reduce like solution. Map-Reduce is a framework for big data analysis problems where parts of analysis can be split up between multiple nodes that include threads, processors or even separate machines. It was influenced by "map and reduce" from functional programming but its usage area is loosely related. While Map-Reduce offers fault tolerance in parallel execution environments, Halocoin provides an incentivization framework to run respectively large map-reduce solutions in a grid computing environment. We show the complete flow of tasks in Halocoin in Figure 3.3.

Halocoin introduces 5 new different transaction types that would facilitate job distribution and rewarding mechanics. Moreover, difficulty calculation of Halocoin is also altered to find a balance between mining and farming. If farming does not yield expected results, mining should also get harder to encourage a more balanced network. If mining does not attract individuals, then farming should slow down because the integrity of blockchain lays behind mining. We acknowledge that all the requirements of Halocoin are actually possible to be executed by a smart contract on a blockchain like Ethereum. However, we focus on eliminating or at least decreasing the environmental impact that is caused by proof-of-work mining. It would not make sense to only promote methodical problem solving on a traditional blockchain.

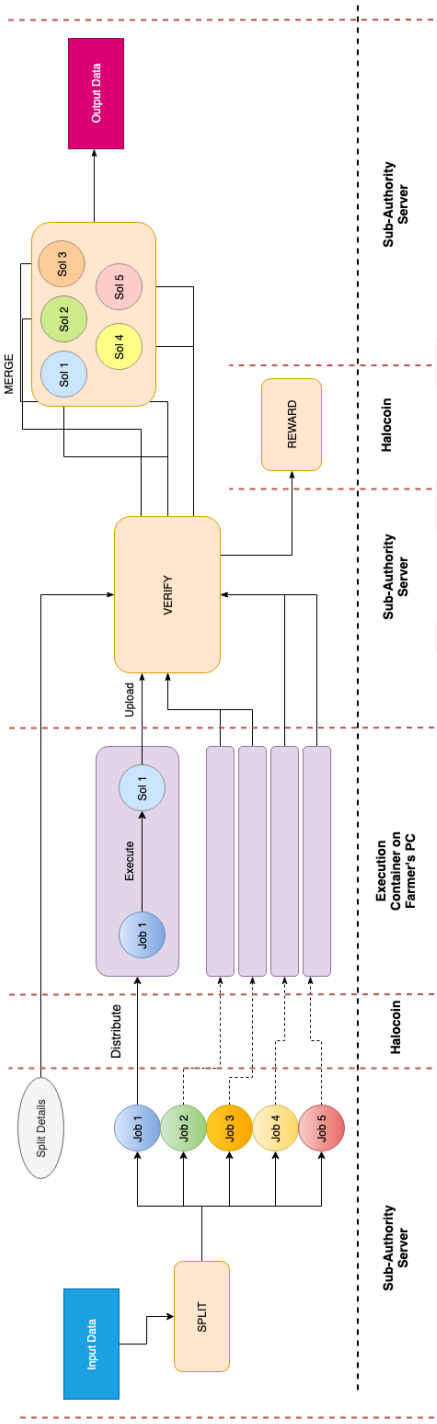

Figure 3.3: A timechart of how Coinami and Halocoin functions.

3.6.1 Transactions

3.6.1.1 AuthReg

Sub-Authorities are registered to Halocoin through *AuthReg* transaction. It includes a certificate which is signed by RA, title and description of the authority and initial supply of Halocoins that can be distributed.

3.6.1.2 PoolReg

Halocoin requires an initial investment of coins to participate as a farmer to prevent spam and inoperability attacks. This investment turns into a behavior score for the farmer. Farmer gets positive or negative scores for each job that is respectively rewarded or failed. Farmers will be no longer eligible to get assigned for a job if they fall below a threshold score. *PoolReg* transaction simply declares the desire to convert coins to positive behavior score to join farmer pool.

3.6.1.3 Application

Pool registration makes a wallet, a farmer candidate. However, because uncompleted jobs result in a negative score, farmers should declare their intention to farm when they are available. *Application* transaction contains two fields: 1) Mode, 2) List. Mode can be *single* or *continuous* which tells the network that whether they wish to get assigned once or continuously. List parameter is to select which sub-authorities they want to work for.

3.6.1.4 JobReward

Although assignments are calculated automatically according to hard-coded rules, verification result should be declared by sub-authorities on Halocoin to reward

farmers. *JobReward* references an already assigned job to indicate that the associated farmer should be rewarded for their work.

3.6.1.5 JobDump

Sub-Authorities announce the availability of jobs through a transaction called *JobDump*. It contains job identifier, download and upload URLs, possible reward amount, and hashsum of the job. Only wallets that have previously execute *AuthReg* can publish this type of transaction.

3.6.2 Definitions

Mixer, Demixer, Verification and Read Mapping terms are associated with proof of concept implementation of Coinami dealing with DNA Sequence Alignment. Halocoin uses similar but different terms to cover more area of computational problems. Coinami, on top of Halocoin, actually consists of 6 fundamental parts. We can categorize them according to their execution environment. Sub-Authority Private Server handles; 1) Split, 2) Verify, 3) Merge. Halocoin handles; 4) Distribution and 5) Rewards. Farmers handle 6) Job Execution. In this section, we provide definitions to some of the terms that are used to explain Halocoin and the broader Coinami framework.

3.6.2.1 Job

A job is a package that is generated by sub-authorities to be published on Halocoin. Its execution details, possible reward amount and issuer authority is bundled together with input data. Each job is considered to be unassigned and immediately open for assignment when first published. When a job is assigned to a wallet, Halocoin client downloads the assignment and executes necessary actions defined by Job Manifest which includes aforementioned details. At this point,

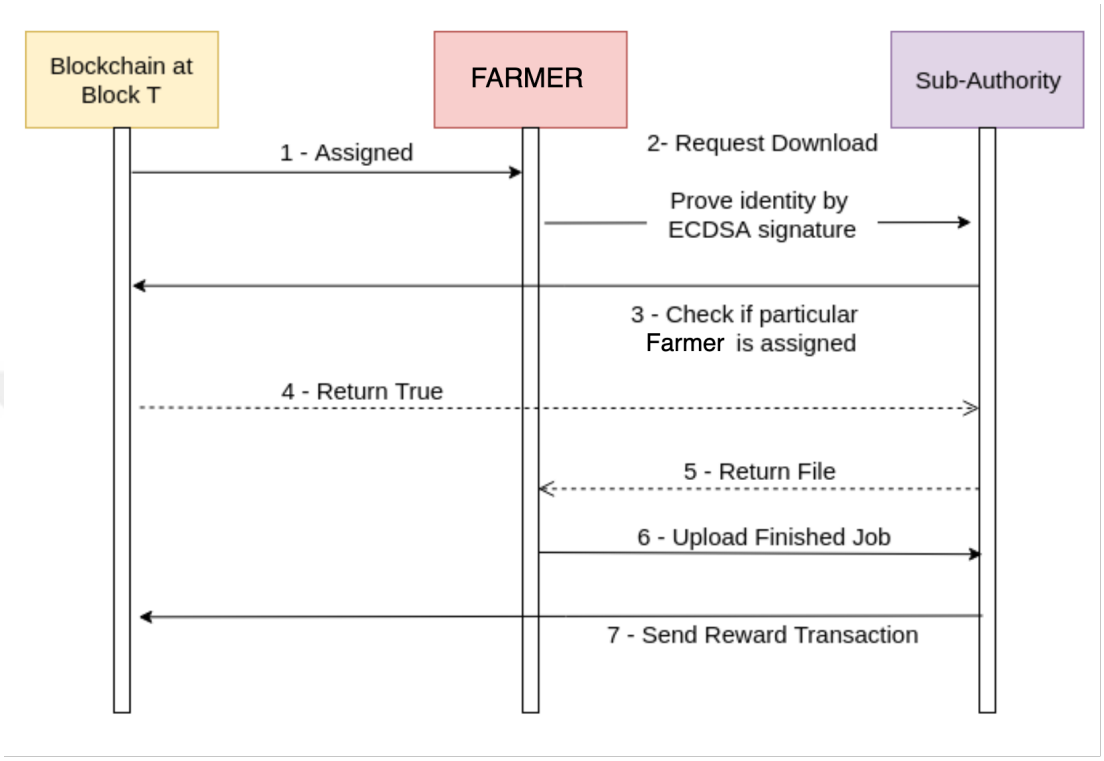

Figure 3.4: Workflow of Farmer service implemented in Halocoin. Blockchain acts as the source of truth between client and sub-authority. File transfer can be achieved through traditional HTTPS or FTP protocols, or secure channels over UDP.

a job is transformed into a solution and ready to be uploaded to Sub-Authority servers.

3.6.2.2 Distribution

The most important contribution of Coinami besides providing incentive for farmers is the consensus model for load-balancing. In grid computing, authorities solely decide how to distribute their jobs and this directly affects the voluntary credit system. If credits ever meant something, an authority could easily assign jobs to whomever they want. To eliminate such a problem, job distribution is implemented as an integral smart contract inside Halocoin. Distribution happens at every constant number of blocks periodically. Each client must calculate the same result for ditribution. Distribution algorithm is given in Algorithm 1.

Algorithm 1 Job Assigment

```
1:  procedure UnAssign(currentBlock)
2:      jobList ← getAssignedJobs()
3:      unassignmentPeriod ← unassignmentMultiplier * assignmentPeriod
4:      unassignBefore ← length(currentBlock) − unassignmentPeriod
5:      for job ∈ jobList do
6:          assignedBlock ← lastAssignedBlock(job)
7:          if assignedBlock ≤ unassignBefore then
8:              unassignJob(job)
9:  procedure Assign(currentBlock)
10:     accounts ← emptyArray()
11:     for address ∈ getWorkerPool() do
12:         account ← getAccount(address)
13:         if account.assignedJob ≠ None then
14:             accounts.add(account)
15:     for (account, address) ∈ accounts do
16:         for auth ∈ account.applicationList do
17:             availableJobs ← getAvailableJobs(auth)
18:             sort(availableJobs) by reward amount
19:             if length(availableJobs) ≥ 0 then
20:                 assignJob(account, availableJobs.first())
```

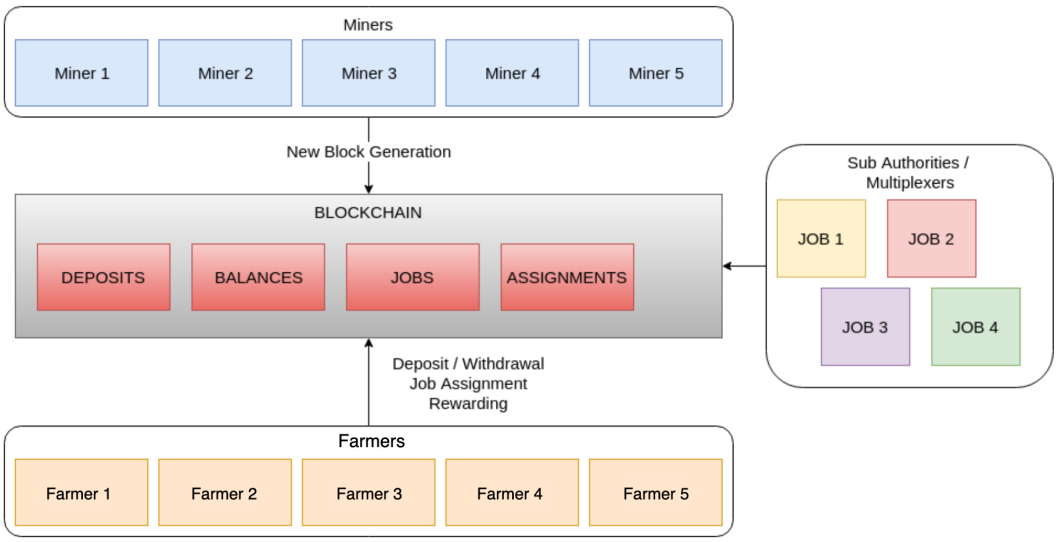

Figure 3.5: A general view of how multiple parties use blockchain to facilitate their communication.

3.6.2.3 Split

Either by mixing or just splitting the initial task into many smaller parts, every concurrent computation starts with deciding how to break down and distribute the original sequential execution. This process is called "Split" in Coinami. Every Sub-Authority implements the split interface with their Docker container and Sub-Authority instruments of Coinami handles the identification, rewarding and distribution of each job to Halocoin. Split executable takes 4 arguments: 1) Image Name (Docker), 2) Host Input Directory (Where initial data is located), 3) Coinami Pool Directory (Where to store intermediate files), 4) Additional Image Parameters.

3.6.2.4 Verification

Verification is executed similar to Split. Received solution is fed to a Docker container with Coinami Pool Directory in case additional information is needed for verification. A boolean result is used to decide whether the reward transaction should be published. If verification fails more than some constant number of tries, the user is blocked from uploading to server.

3.6.2.5 Merge

Merge is the final step in overall Coinami workflow. Merge is executed when all the solutions are obtained for a task. Required parameters are: 1) Solutions (list of uploaded files), 2) Coinami Pool Directory, 3) Final Output Directory, 4) Image name, 5) Additional image parameters.

3.6.2.6 Job Execution

Job execution is conducted by Halocoin farmers on their personal computers. All worker executables must be implemented inside a Docker image because cross-platform support is crucial to ensure a wide adoption. Docker enables any kind of image to run on modern desktop operating systems [68]. After the worker produces a result, it is uploaded to Sub-Authority servers by a URL that is defined in *JobDump* transaction.

Chapter 4

Results

In this chapter, we provide detailed analysis of how Coinami and Halocoin performs in experimental settings. Although our proposed framework does not produce quantifiable results by its nature, we evaluated the time spent on computation and IO. The goal of Coinami has always been transfering the computation cost to personal computers. We reviewed each stage while confirming that the yielded results stayed correct.

We discussed possible bottlenecks like communication bandwidth in the preceding chapters. These complications can be dealt with by carefully adjusting many parameters in the system; e.g., single job size, overall task size, job publish rate. As we discussed previously, most parts are decoupled and can function independently following configurable rules. Given results usually keep some of these parameters constant because evaluating every configuration in Coinami would be inefficient.

4.1 Halocoin

Blockchain implementation is a difficult process. Starting from simple cryptographic functions to sign transactions, to handling multiple candidate blocks while

being able to revert changes, there are lots of quirks and twists that can cause a bug. Being peer-to-peer also introduces another level of challenge since small bugs are echoed through the network until all the nodes become unfunctional.

Our tests with Halocoin have been successful most of the time. It could handle small number of nodes with moderate number of transactions per block. However, testing becomes harder when multiple sub-authorities are introduced. Especially *jobDump* transactions usually triggered automatic assignments which then started farmer modules. It has never been comfortable to test in a practice environment. That is why we left Halocoin field practice as a future work. We believe that its capabilities are sufficient but scaling requires rigirous testing.

4.2 Mixer Performance

Mixing and demixing are expected to be IO bounded. However, mixing includes secure hashing, which might present an uncertainty whether cryptographic hashing contributes to undesired increase in CPU time. We provide a comparison of *Mixer* results where hashing is toggled in Table 4.1. Table 4.2 also provides the ratio between *Mixer* duration and task size.

Mixer presents little overhead, which is negatable compared to its IO/CPU rate.

4.3 Verification Modes

Halocoin expects verification to run in Sub-Authority servers after a solution is received. Coinami does not immediately evaluate the results, rather a queue is used to increase the efficiency in verification nodes. However, validity check of a bam file requires a reference that is loaded in memory. We explained earlier how our new tool *verifybam* offers a daemon mode which is rare in bioinformatics

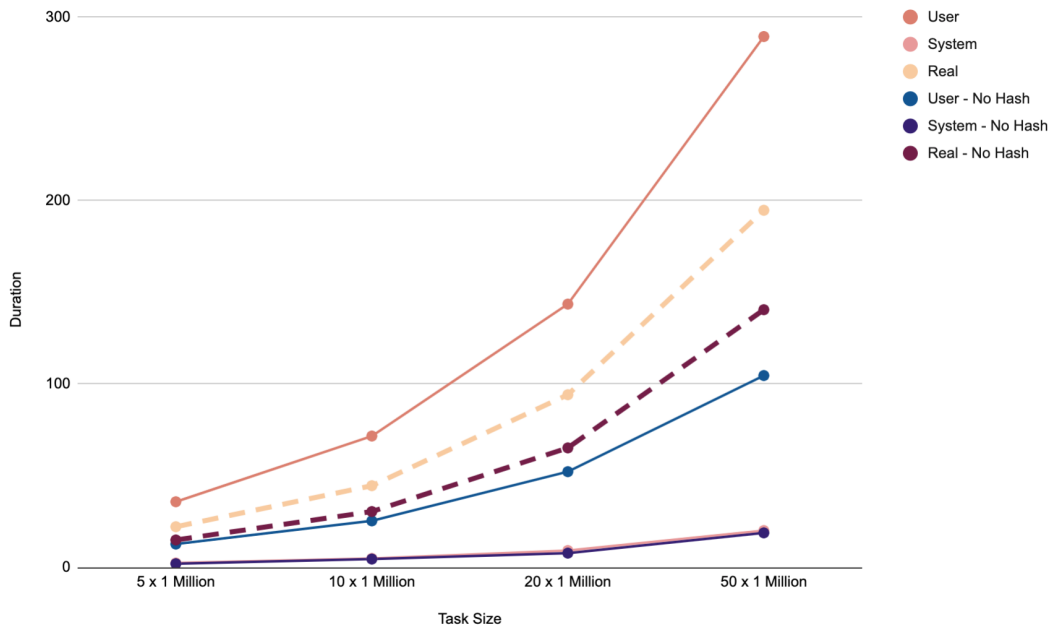

Figure 4.1: This graphic visualizes the data in Table 4.1. Important property real time is differentiated using dashed lines. The jump in user time seems to have no effect on real time. The reason behind that is multi-threading combined with high IO time.

Table 4.1: Mixer duration in various task sizes.

	5 x 1 Million	10 x 1 Million	20 x 1 Million	50 x 1 Million
User	35.81	71.63	143.53	289.35
System	2.43	4.92	9.24	20.13
Real	22.2	44.56	94.15	194.67
User - No Hash	12.74	25.4	52.2	104.6
System - No Hash	2.03	4.58	7.81	18.86
Real - No Hash	14.98	30.46	65.16	140.48

User time, system time and real time respectively corresponds to total time that is spent in user mode, spent in kernel mode and actual wall clock time. We are insterested in two properties; real time and the scaling of user time with respect to task size. Real time tells us the overall time it takes to finish mixing. It has a linear growth shown in Figure 4.2. Although user time gets affected by hashing, it does not reflect to real time thanks to multithreading performance of *Mixer*.

applications. Instead of running it from scratch everytime, *verifybam* stays in the background, waiting for new tasks. Tables 4.3 and 4.5 shows the outcome of using *verifybam* in daemon mode for various sizes of alignment files.

Although reference loading can be considered as I/O operation, it has a client facing effect. Users would disapprove if rewarding takes longer times. We believe that daemon mode has a potential in state-of-the-art read mapping tools.

Similar to how we tested *Mixer* with respect to its hashing performance, we also evaluated *verifybam*. The difference between the two is that *verifybam* offers a more mature parallel execution capabilities because it does not have a lock for I/O write operations. Table 4.5 shows that when used in single threaded mode, verifybam performance quickly drops when hashing is enabled. On the contrary, hashing almost has no affect in multi-threaded mode. This presents many advantages while maintaining a strong security for validity of solutions.

Table 4.2: Ratio of *Mixer* duration to task size.

	5 x 1 Million	10 x 1 Million	20 x 1 Million	50 x 1 Million
Real	4.44	4.456	4.7075	3.8934
Real - No Hash	2.996	3.046	3.258	2.8096

Here we show by experiment that cryptographic hashing does not cause exponential growth in execution time. Figure 4.2 visualizes the data given in this table.

Table 4.3: Verifybam performance in sequential mode.

	1k	10k	100k	1m	5m
Wall Clock	21.975	22.116	22.59	26.429	46.415
Total CPU	20.732	20.904	22.58	39.36	113.196
Kernel CPU	1.412	1.54	1.704	2.764	12.676

Verifybam performance results with various input sizes. Thread count is 4 and daemon mode is not used. Sizes refer to read pairs i.e., 1k set includes actually 2000 total reads.

4.4 Coinami overall Execution Time

Evaluating each component has its benefits in terms of pinpointing potential areas of issue. Given that Coinami and Halocoin together are composed of six high level modules, we have a lot to cover to completely evaluate our new framework. Instead, we are going to focus on Coinami's contributions to Sub-Authority servers. In the end, what Coinami had set out to achieve was taking the burden off of research centers. In this section, we present our main findings in total server cost of running read alignment including post-processing on on-premise clusters against using Coinami to distribute CPU intensive parts.

First, we decided to run a full course of Coinami including mixing, execution in

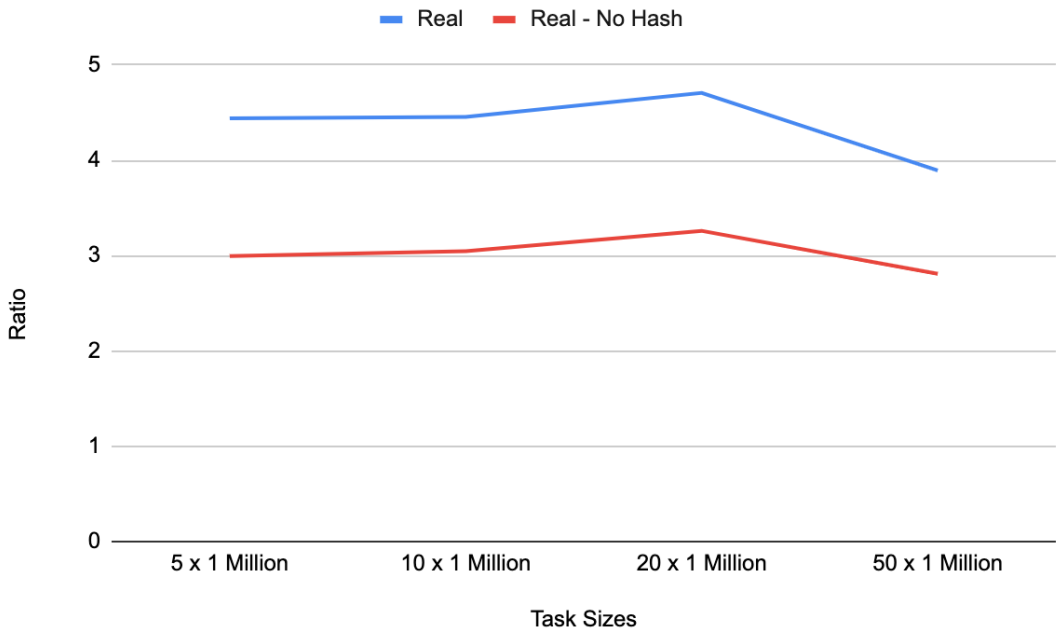

Figure 4.2: This figure visualizes the Mixer growth rate. Ratio refers to the proportion between total mixing time and the input size. It is evident that both hashing and no hashing variants have linear growth rates.

client computers, verification of all the jobs and demixing. We are not interested in execution details in client computers since it does not have any affect on server load. We already presented these findings in earlier sections but now we are going to gather them to compare with modern read mapping tools. Initial results from our first experiment is given in Table 4.6.

There are couple of conclusions we can draw from 4.6. First of all, as we have discussed before, *Mixer* is not able to take advantage of parallelization. Its ceiling is limited by IO speed. Moreover, *verifybam* operates as theoretically defined it would. Thus, its operation cost is almost negligible. However, the point of interest in this table is the difference between user and real time for Coinami total and BWA-Samtools total. BWA-Samtools flow utilizes CPU to the fullest with more than 10,000 seconds. We are not claiming there is a wastage by BWA, on the contrary BWA performs above expectations and we have chosen it for client side execution. Yet, there is a big difference in user time between Coinami and BWA. This difference is less nuanced in real time due to Coinami's expected

Table 4.4: Verifybam performance in daemon mode.

	1k	10k	100k	1m	5m
Wall Clock	0.024	0.108	0.650	5.187	25.545
Total CPU	0.0	0.0	0.0	0.0	0.0
Kernel CPU	0.0	0.0	0.0	0.0	0.0

Since we evaluate performance of the client, there is no CPU operation reported. Still, we can infer that daemon mode gets rid of a constant overhead.

Table 4.5: Hashing's effect on performance.

	100k	1m	5m
Single Threaded - Hashing Enabled	1.35	14.34	71.82
Single Threaded - Hashing Disabled	1.06	9.75	46.77
4 Threaded - Hashing Enabled	0.56	5.33	27.84
4 Threaded - Hashing Disabled	0.51	4.26	25.80

These results are taken when verifybam is run in daemon mode. As we can see, hashing causes significant overhead. It is more nuanced in larger inputs with single threading. Luckily, this is resolved by multi-threading.

I/O overhead.

In conclusion of this experiment, Coinami lowers the cost of wall clock time by 75%. The most significant improvement comes in CPU as Coinami offers a 95% more idle time. We also provide similar results with different parameters below in Table 4.7.

Besides overall execution details on server side, we also conducted a whole genome experiment in Coinami. We use a 40x coverage genome with approximatelly 800 million read pairs. This experiment was focused on how farmer count affects the overall time it takes for Coinami to finalize a batch. Before delving

Table 4.6: Coinami vs State-of-the-art Read Mapping, Variation 1.

	Mixer	Verifybam x 10	Demixer	Total	BWA - Samtools
User	71.63	4.21	477.4	591.13	10919.62
System	4.92	0.16	7.23	13.75	602.16
Real	44.56	4.38	151.69	240.05	855.22

In this experiment, 10 FASTQ files with 1,000,000 reads were mixed to create 10 jobs with 1,000,000 reads each.

Table 4.7: Coinami vs Read Mapping Tools, Variation 2.

	Mixer	Verifybam x 10	Demixer	Total	BWA - Samtools
User	289.35	26.98	2187.37	2746.52	49926.27
System	20.13	0.91	34.15	63.38	2866.26
Real	194.67	27.84	724.65	1197.72	1982.48

In this experiment, 50 FASTQ files with 1,000,000 reads were mixed to create 10 jobs with 5,000,000 reads each.

into results of the experiment, we need to explain what kind of results we are expecting. To this end, total duration of Coinami has to be formulated.

We assume each farmer is identical in terms of their network bandwidth and computational capabilities for this equation. Moreover, we calculate a single batch of jobs which consists of a single split and merge operation.

There are two assumptions to be considered. If all farmers download at the same time and their total download bandwidth does not exceed server's upload bandwidth, there will be no overhead while downloading jobs and uploading solutions. They can all function synchronously. Another assumption is that total download bandwidth of farmers exceeds the upload bandwith limit of the server. In this case, all farmers downloading at the same time, sacrificing from their

download speeds can incur additional duration per download and upload cycle. If they do not sacrifice and rather wait in turns to download tasks, it generates another pipeline of job flow. In our experiment, we have had farmers sacrifice their bandwidth for synchronized workflow. The equation for that setup is given in Equation 4.1.

$$total\_duration = \lceil job/farmer \rceil * (js/dbf + e + v + js/ubf) \tag{4.1}$$

Total duration equation for complete Coinami workflow. It is assumed that total download/upload bandwidth of sub-authority servers exceed the total bandwidth of all farmers. Variables referred as; js: Job Size, dbf: Download bandwidth per farmer, ubf: Upload bandwidth per farmer, e: Execution duration, v: Verification duration.

As we can infer from Equation 4.1, duration is strictly linear in terms of client performance. Execution, verification and IO operations like download and upload takes linear time. Only thing that needs to be taken care of is the ceiling function used to determine cycle count. For example, if there are 19 farmers for 40 jobs, ceiling hits 3 but if there are 20 farmers, ceiling hits 2. This accounts for a whole cycle duration.

In our experiment we varied the farmer count starting from 20 to all the way down to 0, which would render Coinami useless. You can see how total duration is affected in Figure 4.4. As we increase the number of farmers, we see less improvement at each iteration. This is due to the fact that there are bandwidth limitations between sub-authority and the farmers. Coinami hits a scaling obstacle when it comes to networking aspect as we discussed in earlier chapters.

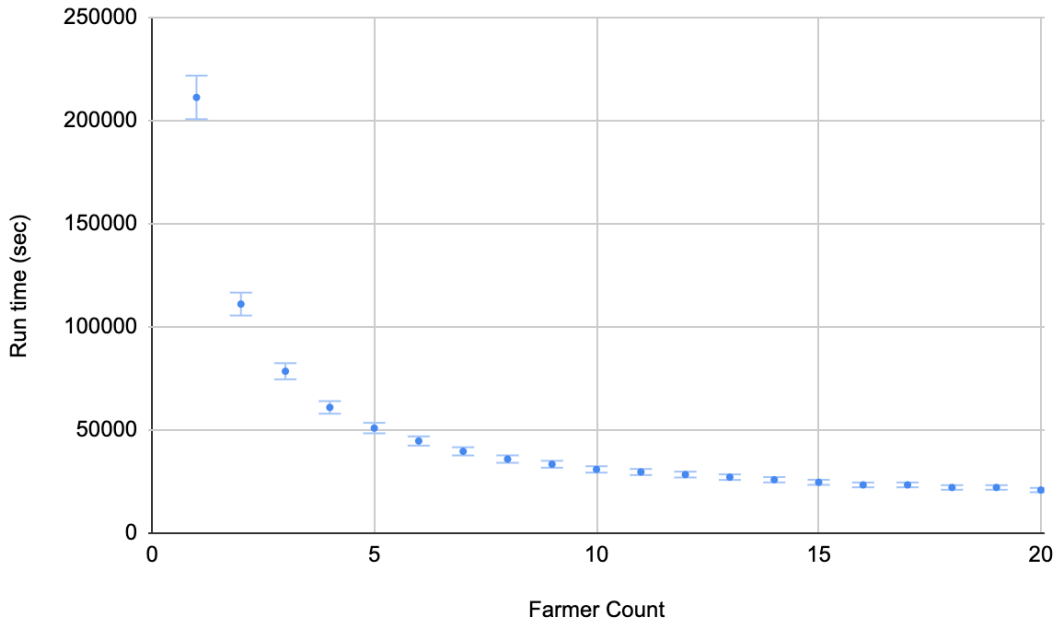

Figure 4.3: Effect of farmer count on total execution time of Coinami. In this figure, farmer count is varied between 0 and 20. At the start, it is technically impossible to run anything on Coinami without a farmer. The graphic also includes 5% error margin for the experiment.

Chapter 5

Discussion and Future Work

We present a novel framework that connects the idle CPUs from all around the world to promote scientific progress while offering a unique incentivization model. Modern tools for read mapping such as BWA, minimap2 and Bowtie are based on established algorithms which have small room for improvement. Coinami shifts the focus from analytical aspect to architecture. Vertical scaling of computing resources is linear and not cheap [69]. Through horizontal scaling, much more efficient advancements can be achieved. Although this kind of scaling is best practiced on Cloud systems, they also represent privacy, single source of failure, and financial problems even it's not close to vertical scaling costs. Instead, we proposed an alternative solution using grid computing and blockchain.

In our experiments, Coinami demonstrated that it is viable to be used as an alternative clusture. Almost all of the computation is diverted from servers to farmers while increasing swapping CPU demand with IO demand in the server. Moreover, we suggest that CPU time which is salvaged by Coinami can be utilized while IO is busy. Also, not all scientific problems would be IO demanding as Halocoin can host many other Sub-Authorities. Furthermore, we developed a new blockchain that offers an equilibrum between PoW and farming, theoritically promotes both options giving opportunity to both professional miners and science ethusiasts to be active participants. Finally, Halocoin also is utilized for job

distribution. This could have been done in Sub-Authority servers but blockchain is capable of running decentralized autonomous applications [17]. We believe that it is necessary to publicize this process because favoritism might be a concern among farmers.

On the other hand, Coinami poses some disadvantages over traditional server infrastructures. It puts a heavy toll on disk usage due to many intermediate operations. Mixing does not provide complete privacy enhancement. Quasi-centralization can be a blocker for most cryptocurrency users. Also, it is not suitable to use in limited networks due to the fact that a single job can exceed 10 GBs at time.

In the end, we have high hopes for Coinami and Halocoin to find success in practice. The disadvantages that are listed above can be mitigated by following future work we planned.

5.1 Future Work

We categorized the upcoming improvements into 3 parts; 1) Technical Development, 2) User Experience, 3) Community.

5.1.1 Technical Development

Current implementation of Coinami is strictly a proof-of-concept which must not be used in public production. There are possibly countless vulnerabilities present in the code. First of all, Halocoin is implemented in Python, a language that enabled us to quickly develop a prototype. However, Python is not a performance tuned programming language [70]. As most cryptocurrencies, Halocoin should be written from scratch in C/C++ or any other native compiling language to boost its ability to quickly synchronize.

Coinami also currently does not house a user interface for server users. All operations on server side can be execute only using a command line interface. It is crucial to offer a better experience for Sub-Authorities to easily onboard to Coinami. Split, Verification and Merge commands should be easily performed at any given time. Moreover, there should be a way to track the progress of a batch of jobs. Additionally, Sub-Authorities do not have any right to decline an incoming download request if it is legitametely signed by the owner. To prevent any malicius behavior on server side, Coinami must hide any identifiable information from Sub-Authority administrator.

5.1.2 User Experience

User experience is proven to be one of the most important aspects of a new product. It affects the general public adoption rate [71]. Contrary to most research projects, Coinami's success depends on gaining traction in cryptocurrency world. We hope to provide a simple, robust, and capable interface that achieves our higher goal which is rapidly integrating new users. For this purpose, first course of action must be preparing a better onboarding documentation, explaining the requirements like Docker and why they are needed. Moreover, fading technical information away while prioritizing farming related information should help to convey our mission to new adopters.

Halocoin currently lacks a restore/backup wallet feature. This is going to be fixed in future releases. Farming requires a functional Docker core, which is not common personal computers. Halocoin should also provide an uncomplicated Docker installation guide.

5.1.3 Community

A cryptocurrency means nothing if there is no official way to trade it outside of blockchain. Bitcoin would have no value against U.S. Dollar if there were no

exchange sites. A market should be established for Halocoin but if it comes from the root authority, it would not feel honest. We believe that third parties will step up if there is enough demand.

Halocoin and Coinami should not be released to public directly. We recommend that they should be field tested among researchers beforehand to get ready for a larger audience. However, blockchain must be purged when public release is announced.

Bibliography

[1] F. Sanger, G. M. Air, B. G. Barrell, N. L. Brown, A. R. Coulson, J. C. Fiddes, C. A. H. III, P. M. Slocombe, and M. Smith, "Nucleotide sequence of bacteriophage x174 dna," *Nature*, vol. 265, pp. 687 – 695, February 1977.

[2] M. L. Metzker, "Sequencing technologies—the next generation," *Nature Reviews Genetics*, vol. 11, no. 1, p. 31, 2010.

[3] E. R. Mardis, "The impact of next-generation sequencing technology on genetics," *Trends in genetics*, vol. 24, no. 3, pp. 133–141, 2008.

[4] Illumina HiSeq X Series. Available at: https://www.illumina.com/systems/sequencing-platforms/hiseq-x.html. Accessed 08-2019.

[5] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with bwa-mem," *arXiv preprint arXiv:1303.3997*, 2013.

[6] H. Li and R. Durbin, "Fast and accurate short read alignment with burrows–wheeler transform," *bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[7] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J. O. Kitzman, C. Baker, M. Malig, O. Mutlu, *et al.*, "Personalized copy number and segmental duplication maps using next-generation sequencing," *Nature genetics*, vol. 41, no. 10, p. 1061, 2009.

[8] F. Hach, F. Hormozdiari, C. Alkan, F. Hormozdiari, I. Birol, E. E. Eichler, and S. C. Sahinalp, "mrsfast: a cache-oblivious algorithm for short-read mapping," *Nature methods*, vol. 7, no. 8, p. 576, 2010.

[9] B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short dna sequences to the human genome," *Genome biology*, vol. 10, no. 3, p. R25, 2009.

[10] H. Xin, D. Lee, F. Hormozdiari, S. Yedkar, O. Mutlu, and C. Alkan, "Accelerating read mapping with FastHASH," *BMC Genomics*, vol. 14 Suppl 1, p. S13, 2013.

[11] N. A. Fonseca, J. Rung, A. Brazma, and J. C. Marioni, "Tools for mapping high-throughput sequencing data," *Bioinformatics*, vol. 28, no. 24, pp. 3169–3177, 2012.

[12] SetiHome. Available at: http://setiathome.ssl.berkeley.edu/. Accessed 08-2019.

[13] RosettaHome. Available at: https://boinc.bakerlab.org/. Accessed 08-2019.

[14] FINDHome. Available at: https://boinc.berkeley.edu/wiki/FiNDHome. Accessed 08-2019.

[15] W. Dai, "b-money: a scheme for a group of untraceable digital pseudonyms to pay each other with money and to enforce contracts amongst themselves without outside help." http://www.weidai.com/bmoney.txt, 1998.

[16] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system." 2008.

[17] V. Buterin, "A next-generation smart contract and decentralized application platform," *white paper*, 2014.

[18] B. Chase and E. MacBrough, "Analysis of the xrp ledger consensus protocol," 2018.

[19] Blockchain Statistics. Available at: https://www.blockchain.com/en/charts/hash-rate. Accessed 08-2019.

[20] BOINC Statistics. Available at: https://boinc.netsoft-online.com. Accessed 08-2019.

[21] J. D. Watson, F. H. Crick, *et al.*, "Molecular structure of nucleic acids," *Nature*, vol. 171, no. 4356, pp. 737–738, 1953.

[22] R. D. Fleischmann, M. D. Adams, O. White, R. A. Clayton, E. F. Kirkness, A. R. Kerlavage, C. J. Bult, J.-F. Tomb, B. A. Dougherty, J. M. Merrick, *et al.*, "Whole-genome random sequencing and assembly of haemophilus influenzae rd," *Science*, vol. 269, no. 5223, pp. 496–512, 1995.

[23] P. Edman *et al.*, "Method for determination of the amino acid sequence in peptides," *Acta chem. scand*, vol. 4, no. 7, pp. 283–293, 1950.

[24] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *J Mol Biol*, vol. 48, pp. 443–453, Mar 1970.

[25] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences," *J Mol Biol*, vol. 147, pp. 195–197, Mar 1981.

[26] S. Canzar and S. L. Salzberg, "Short read mapping: an algorithmic tour," *Proceedings of the IEEE*, vol. 105, no. 3, pp. 436–458, 2015.

[27] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, F. Hormozdiari, J. O. Kitzman, C. Baker, M. Malig, O. Mutlu, S. C. Sahinalp, R. A. Gibbs, and E. E. Eichler, "Personalized copy number and segmental duplication maps using next-generation sequencing," *Nat Genet*, vol. 41, pp. 1061–1067, Oct 2009.

[28] W.-P. Lee, M. P. Stromberg, A. Ward, C. Stewart, E. P. Garrison, and G. T. Marth, "Mosaik: a hash-based algorithm for accurate next-generation sequencing short-read mapping," *PloS one*, vol. 9, no. 3, p. e90581, 2014.

[29] NovoAlign. Available at: http://www.novocraft.com/. Accessed 08-2019.

[30] Smalt. Available at: https://www.sanger.ac.uk/science/tools/smalt-0. Accessed 08-2019.

[31] P. Ferragina and G. Manzini, "Indexing compressed text," *Journal of the ACM (JACM)*, vol. 52, no. 4, pp. 552–581, 2005.

[32] P. W. Payne Jr, "Genetic information nondiscrimination act of 2008: The federal answer for genetic discrimination," *J. Health & Biomedical L.*, vol. 5, p. 33, 2009.

[33] S. Soini, "Genetic testing legislation in western europe—a fluctuating regulatory target," *Journal of community genetics*, vol. 3, no. 2, pp. 143–153, 2012.

[34] M. Kim and K. Lauter, "Private genome analysis through homomorphic encryption," in *BMC medical informatics and decision making*, vol. 15, p. S3, BioMed Central, 2015.

[35] A. Srinivasan, M. A. Quadir, and V. Vijayakumar, "Era of cloud computing: a new insight to hybrid cloud," *Procedia Computer Science*, vol. 50, pp. 42–51, 2015.

[36] V. Popic and S. Batzoglou, "A hybrid cloud read aligner based on minhash and kmer voting that preserves privacy," *Nature communications*, vol. 8, p. 15311, 2017.

[37] D. Tapscott and A. Tapscott, *Blockchain Revolution: How the technology behind Bitcoin is changing money, business, and the world*. Penguin, 2016.

[38] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *arXiv preprint arXiv:1710.09437*, 2017.

[39] C. G. Akcora, Y. R. Gel, and M. Kantarcioglu, "Blockchain: A graph primer," *CoRR*, vol. abs/1708.08749, 2017.

[40] S. Raval, *Decentralized Applications: Harnessing Bitcoin's Blockchain Technology*. O'Reilly Media, 2016.

[41] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, *et al.*, "On scaling decentralized blockchains," in *International Conference on Financial Cryptography and Data Security*, pp. 106–125, Springer, 2016.

[42] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare, and A. Granzotto, "Bigchaindb: a scalable blockchain database," *white paper, BigChainDB*, 2016.

[43] J. Teutsch and C. Reitwießner, "A scalable verification solution for blockchains," *arXiv preprint arXiv:1908.04756*, 2019.

[44] M. Scherer, "Performance and scalability of blockchain networks and smart contracts," mathesis, 2017.

[45] J. Spasovski and P. Eklund, "Proof of stake blockchain: performance and scalability for groupware communications," in *Proceedings of the 9th International Conference on Management of Digital EcoSystems*, pp. 251–258, ACM, 2017.

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

[47] P. McCorry, S. F. Shahandashti, and F. Hao, "Refund attacks on bitcoin's payment protocol," in *International Conference on Financial Cryptography and Data Security*, pp. 581–599, Springer, 2016.

[48] M. Crosby, P. Pattanayak, S. Verma, V. Kalyanaraman, *et al.*, "Blockchain technology: Beyond bitcoin," *Applied Innovation*, vol. 2, no. 6-10, p. 71, 2016.

[49] J. Bonneau, A. Miller, J. Clark, A. Narayanan, J. A. Kroll, and E. W. Felten, "Research perspectives and challenges for bitcoin and cryptocurrencies (extended version)," *Cryptology ePrint Archive, Report 2015/452*, 2015.

[50] A. H. Dyhrberg, "Bitcoin, gold and the dollar–a garch volatility analysis," *Finance Research Letters*, vol. 16, pp. 85–92, 2016.

[51] C. Richter, S. Kraus, and R. B. Bouncken, "Virtual currencies like bitcoin as a paradigm shift in the field of transactions," *International Business & Economics Research Journal (Iber)*, vol. 14, no. 4, pp. 575–586, 2015.

[52] X. Li and C. A. Wang, "The technology and economic determinants of cryptocurrency exchange rates: The case of bitcoin," *Decision Support Systems*, vol. 95, pp. 49–60, 2017.

[53] Publicly available cryptocurrency financial statistic. Available at: https://coinmarketcap.com. Accessed 08-2019.

[54] G. Zyskind, O. Nathan, *et al.*, "Decentralizing privacy: Using blockchain to protect personal data," in *2015 IEEE Security and Privacy Workshops*, pp. 180–184, IEEE, 2015.

[55] P. McCorry, S. F. Shahandashti, and F. Hao, "A smart contract for boardroom voting with maximum voter privacy," in *International Conference on Financial Cryptography and Data Security*, pp. 357–375, Springer, 2017.

[56] F. Tian, "An agri-food supply chain traceability system for china based on rfid & blockchain technology," in *2016 13th international conference on service systems and service management (ICSSSM)*, pp. 1–6, IEEE, 2016.

[57] A curated list of Bitcoin related crime activities. Available at: https://www.coindesk.com/category/bitcoin-crime. Accessed 08-2019.

[58] J. Truby, "Decarbonizing bitcoin: Law and policy choices for reducing the energy consumption of blockchain technologies and digital currencies," *Energy research & social science*, vol. 44, pp. 399–410, 2018.

[59] J. Vincent, "Bitcoin consumes more energy than switzerland, according to new estimate," July 2019. https://www.theverge.com/2019/7/4/20682109/bitcoin-energy-consumption-annual-calculation-cambridge-index-cbeci-country-comparisontheverge.com [Online; posted 4-July-2019].

[60] H. Pagnia and F. C. Gärtner, "On the impossibility of fair exchange without a trusted third party," tech. rep., Technical Report TUD-BS-1999-02, Darmstadt University of Technology ..., 1999.

[61] L. W. Cong, Z. He, and J. Li, "Decentralized mining in centralized pools," tech. rep., National Bureau of Economic Research, 2019.

[62] N. Leonardos, S. Leonardos, and G. Piliouras, "Oceanic games: Centralization risks and incentives in blockchain mining," *arXiv preprint arXiv:1904.02368*, 2019.

[63] Z. Durumeric, J. Kasten, M. Bailey, and J. A. Halderman, "Analysis of the https certificate ecosystem," in *Proceedings of the 2013 conference on Internet measurement conference*, pp. 291–304, ACM, 2013.

[64] A. M. Ileri, H. I. Ozercan, A. Gundogdu, A. K. Senol, M. Y. Ozkaya, and C. Alkan, "Coinami: a cryptocurrency with dna sequence alignment as proof-of-work," *arXiv preprint arXiv:1602.03031*, 2016.

[65] E. Ayday and M. Humbert, "Inference attacks against kin genomic privacy," *IEEE Security & Privacy*, vol. 15, no. 5, pp. 29–37, 2017.

[66] C. Firtina and C. Alkan, "On genomic repeats and reproducibility," *Bioinformatics*, vol. 32, pp. 2243–2247, Aug 2016.

[67] A. Óskarsdóttir, G. Másson, and P. Melsted, "BamHash: a checksum program for verifying the integrity of sequence data," *Bioinformatics*, vol. 32, pp. 140–141, Jan 2016.

[68] C. Boettiger, "An introduction to docker for reproducible research," *ACM SIGOPS Operating Systems Review*, vol. 49, no. 1, pp. 71–79, 2015.

[69] J. Yang, C. Liu, Y. Shang, B. Cheng, Z. Mao, C. Liu, L. Niu, and J. Chen, "A cost-aware auto-scaling approach using the workload prediction in service clouds," *Information Systems Frontiers*, vol. 16, no. 1, pp. 7–18, 2014.

[70] X. Cai, H. P. Langtangen, and H. Moe, "On the performance of the python programming language for serial and parallel scientific computations," *Scientific Programming*, vol. 13, no. 1, pp. 31–56, 2005.

[71] M. Hassenzahl and N. Tractinsky, "User experience-a research agenda," *Behaviour & information technology*, vol. 25, no. 2, pp. 91–97, 2006.

Appendix A

Code

All written code is open-source, hosted at
https://github.com/BilkentCompgen/coinami