Bilkent University
Department of Computer Engineering

CoinAMI
Coin-Application Mediator Interface

Supervisor
Can Alkan

Members
Ahmet Kerim Şenol
Alper Gündoğdu
Halil İbrahim Özercan
Muhammed Yusuf Özkaya

Jury Members
Buğra Gedik
Cevdet Aykanat

High-Level Design Report
January 1, 2015

Contents

1 Introduction

1.1 Purpose of the system

In this project we aim to create a cryptocurrency system which uses DNA sequence alignment as proof-of-work.

Our system consists of 2 parts, Transactions and Mining. Transactions represent money exchange between two parties. Simply, they are records of participants, input amount and output amount. Every time a transaction occurs, the original (input) coins are destroyed and new (output) coins are generated. With this scheme, every coin can be spent only once. Transaction operation is handled by other peers in the network and these machines are awarded new coins.

Generation of a new block (coin) is called mining. To mine a coin users should solve an assignment that is assigned to them by the server. This assignment will be a small part of a DNA sequence aligment problem for 2 or more people, so that the user can not build an individual's genome from the assignment. When the user calculates the result for the assignment, the server will validate the result and grant the user a coin upon a succesful calculation.

1.2 Design Goals

Security: Both the cryptocurrency and the DNA data should be stored and transferred in a secure way. This is the main design goal due to any leak of DNA data is irretrievable. In server's business logic part we will break up the personal data and distribute it in a way so that it could not be possible to merge the whole data on client's side. More on this will be discussed in this report.

Privacy: Although this was mentioned as a security goal, it is necessary to point out that the DNAs of the people are confidential. Therefore the whole DNA shouldn't be reconstructible from the data that is exchanged between server and clients. This is close to the security design goal

Reliability: The P2P network used to process the currency should be reliable and running. Error handling is a big issue during reliability.

User-Friendliness: Graphical interfaces and outputs should be user-friendly and understandable.

Efficiency: The system should provide high throughput.

1.3 Definitions, acronyms and abbreviations

DNA: Deoxyribonucleic Acid, the complete human genome.

BWA: Burrows-Wheeler Aligner (BWA) is a software package for mapping low-divergent sequences against a large reference genome, such as the human genome. http://bio-bwa.sourceforge.net/

Read: Every one of BWA input files. There are 2 reads for each DNA.

BTC: Bitcoin Currency.

CoinAMI: Coin-Application Mediator Interface, our project aiming to provide an interface between cryptocurrency and an application.

1.4 Overview

Our system will consist of three connected softwares. We are going to use third-party components to increase reliability and stability while decreasing the development's time cost. To decide which architectures we are going to use on each part of the system, it is needed to look into these 3 sub programs and their main functions.

First of all, we need a server to keep the data and respond to client requests. Two main functionalities of the server are allowing the user to join our cryptocurrency network by providing a wallet and assigning complex DNA data to client to analyze and produce coins which is called mining. These two functionalities might seem different but yet connected. Our first third-party component is Bitcoin which is a widely accepted cryptocurrency network and proved to be safe and secure. As it is open source [2], we will be modifying the code in order to add the interface for genome alignment system.

The mining clients are our second node. They will earn coins by doing genome alignment during a transaction and providing the output to the server.We will be using libbitcoin [4], an asynchronous C++ Bitcoin library for implementing the mining processes.

Regular clients are the clients that does transactions and uses the currency using a graphical user interface. As we will be planning the system totally compatible bit current clients, the users will be free to select any Bitcoin client for that purpose. We recommend Electrum [5] which is a sımple and lightweight Bitcoin client.

2 Current software architecture

BWA [1] alone can be used in a single machine to get output but the computing power of a single computer provides low throughput and it takes days to complete a genome alignment.

SEAL is a system that works on Hadoop clusters that adds distributed system functionality and duplicate removal to the BWA. While BWA itself gives output in 89 hours for a dataset, SEAL with 32 nodes provides output around 5 hours [6].

3 Proposed software architecture

3.1 Overview

In this section, the detailed description of the proposed software architecture will be given through different topics which are subsystem decomposition, hardware-software mapping, persistent data management, access control and security, global software control and boundary conditions.

3.2 Subsystem Decomposition

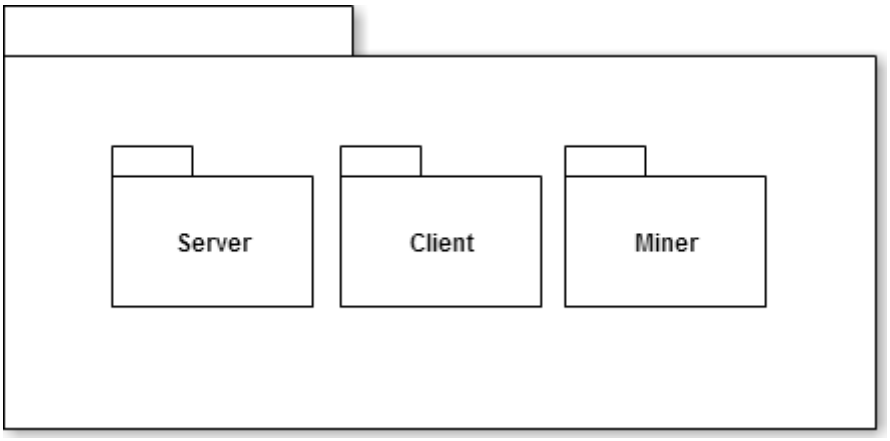

Figure 1: Subsystems

The system has three parts, namely servers, clients and miners. Server side is the main machines that deal with client connections, DNA fragmentation and distribution to clients.

Client side is the users of the CoinAMI currency. Miner machines solve the DNA data on transaction requests and send the solutions to the client.

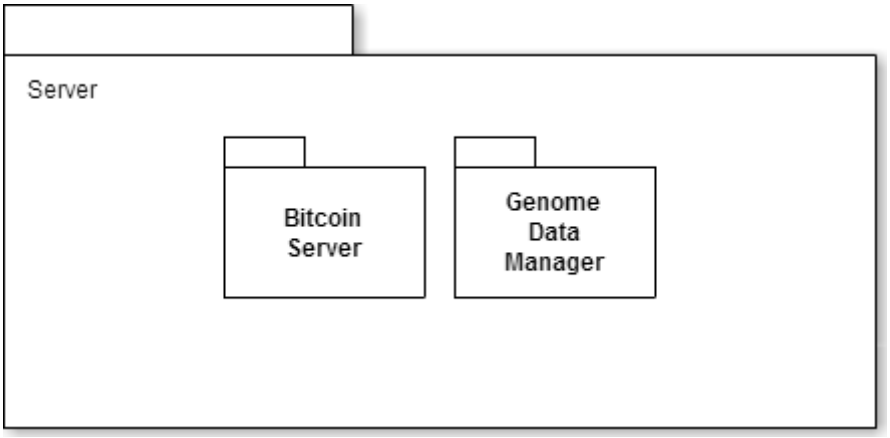

Figure 2: Server

Bitcoin server is our modified Bitcoin server which uses alignment process as a proof-of-work to provide coins to the miners. Genome Data Manager is our database which holds the genome data and processing progress of each genome.

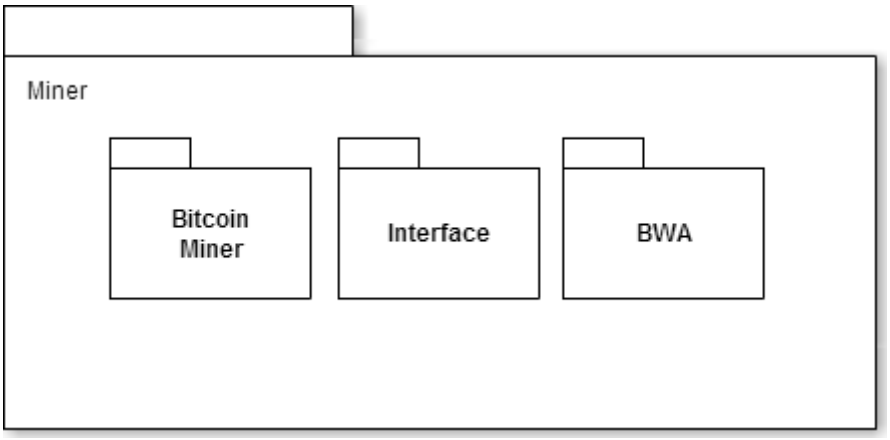

Figure 3: Miner

Bitcoin miner is our modified bitcoin client which communicates with the BWA using our interface in order to realize the transactions. Server and Miner will communicate using asynchronous requests.

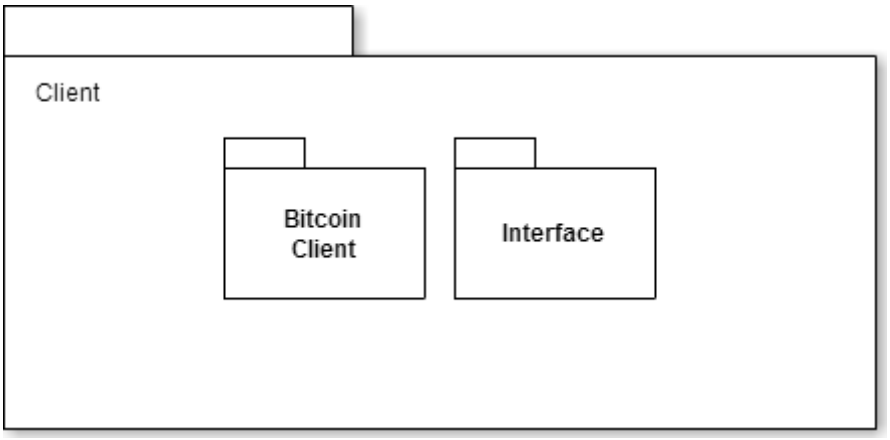

Figure 4: Client

Bitcoin client allows the user to use the cryptocurrency, make some transactions and handles the server communication. It also manages the wallet and is responsible for safety and security of the wallet.

3.3 Hardware/Software Mapping

Servers will run on Linux machines and they are in a peer-to-peer network. Bitcoin is implemented on C++ and we will use the same codebase for the beginning. Server will also provide a web interface that allows administrators to upload genome data and retrieve results.

Clients are connected to one of the servers and can run on any platform (Windows, Linux, Android etc.) as there are third-party bitcoin client implementations.

Miner component will be implemented on Linux machines and during the mining process, they all are connected to internet and in a peer-to-peer network handled by the server.

3.4 Persistent data management

We will be using MySQL for genome data management in all servers.

Bitcoin uses blockchain which is a shared transaction database among all the mining clients. In the past, blockchain forked one time (there were a conflict between network peers) because of differences between miner versions but Bitcoin is stable and running quite good for months.

3.5 Access control and security

In our proposed system, the DNA should be secured so that clients can not reconstruct the genome data. Therefore, clients and miners should be blind to which genome block they have and to whom this belongs.

The server (admin) has access to everything. Admin is the one to provide input data. Therefore it is logical that he has access to everything.

The clients and miners are Peers. Any peer may be miner of a transaction or a client who sends or reveives coin. The clients have no access to the DNA data, they only have access to the Block Chain which keeps the whole global transaction history. They don't have anything to do with DNAs and genomes in this case.

The security requirements are the confidentiality of DNA information and bitcoin wallets of clients.

The miners are the workers that solve genome data with BWA when there is a transaction request between 2 clients. They get a genome block from a DNA, however, they neither have the information about to whom this DNA belongs nor to which DNA that genome block belongs.

The whole transaction process is encrypted using various algorithms such as RSA. Wallet IDs of an account are known only by its owner. So, other people cannot know other people's account information if they don't share it themselves.

In another aspect, we need to secure the correctness of the solution data miners sent. They may try to hack the system and provide couterfeit data. To prevent that, the system includes some random decoy input, result of which is already known. If the miner is really solving the input, we can distinguish by the solution to the decoy input.

3.6 Global software control

Servers will hold genome files and send miners the data, so serverside control is administrator's control on genome files and retrieval of data from a web interface.

Client control depends on the platform, if the client works on an Android phone, controls will be using touch screen, if the client works on PC or Mac, controls will be done by Mouse and Keyboard interactions.

Mining client will run as a console application and collection of coins will be done automatically. Collecting the coins from the miner to a wallet will require a command line argument.

3.7 Boundary conditions

3.7.1 Server

Initialization: The server will be up and running all the time. IP address of the server will be sent to main server to be stored in a database. When it is first initialized, the reads from existing uninitialized genome files will be given unique IDs, not disclosing the identity of the person who donated the genome, and these IDs will be stored in a database.This database identify which read belongs to the which genome.

Termination: Should the server terminate because of an unexpected error, blockchain stops, so the transactions cannot be done by the clients. Upon termination, the IP address of the server should be removed from the main server.

Counterfeit data handling: The part of the data sent to a user to be aligned will contain some decoy reads, which doesn't belong to any whole genome that's in the list of the genomes. The alignment of these decoy reads will be already known beforehand. These already known alignments should match the actual alignment data sent by user. So if a user tries to counterfeit the aligment data, it will be detected when the decoy data doesn't match.

3.7.2 Client

Initialization: Starting the client application depends on the platform.

Termination: Client may be terminated by user successfully after all transactions are sent to the server and shared among at least one peer.

3.7.3 Miner

Initialization: Running the mining client with a server address will be done in command line. Mining application will automatically get a wallet and start processing transactions.

Termination: Mining application can be terminated whenever user wants. The collected coins will be persistent.

4 Subsystem Services

4.1 Server Decomposition

Genome Database Manager: Database manager will handle the database operations. This database will contain which read belongs to which genome. Each read will have a ID that is not disclosing the owner of the genome to the clients.

Bitcoin Server-Database Interface: This component is the main part of the project. This interface will provide an interface between bitcoin server and database which allows us to use the BWA results in order to validate and approve transactions.

Bitcoin Server: Bitcoin server will handle transactions and blockchain errors. This component's code will be modified from the open source Bitcoin project in order to make the system use the interface as a coin validator.

4.2 Client Decomposition

Bitcoin Client: Bitcoin Client is going to be used for wallet management and coin transactions between users. It is also necessary for miner application to transfer generated coins to a valid bitcoin wallet. We are going to use Bitcoin network on server side. Thus, clients can use the any client program that is designed for Bitcoin. This is going to be completely third-party component, meaning we are not going to modify anything about it.

Graphical User Interface: Client has many use cases such as creating a wallet id, restoring an existing wallet id, managing a contact list, sending and receiving coins and etc. Our client service should be able to provide all these features. The third-party component Bitcoin Client has all these features implemented inside.

4.3 Miner Decomposition

Bitcoin Manager: This service will be the default mining client for Bitcoin. The mining client code will be modified in order for it to require an approval from the server to realize a transaction.

Coin-Application Interface: The approval of the interface and server will be required to do a transaction by miner. This approval can be gotten by providing correct output for a

genome data that is received from server.

BWA: This application is run by the Coin-Application Interface. It takes the genome data as input, it aligns the data on the DNA and returns the result to the interface. [1]

5 References

[1] Bio-BWA SourceForge page http://bio-bwa.sourceforge.net/

[2] Bitcoin GitHub repository https://github.com/bitcoin/bitcoin

[3] Bitcoin Paper https://bitcoin.org/bitcoin.pdf

[4] Lib-Bitcoin, C++ Library for Bitcoin http://libbitcoin.dyne.org/

[5] Electrum Bitcoin Client https://electrum.org/

[6] P. Luca, L. Simone, Z. Gianluigi; SEAL: a Distributed Short Read Mapping and Duplicate Removal Tool http://bioinformatics.oxfordjournals.org/content/early/2011/06/22/bioinformatics.btr325.full.pdf