1. Introduction

When we have a look at our environment, we can see that technology is everywhere. We can see from daily practices to a very narrow study field that It is the way how our community is being developed. To develop technology, data is the most important resource to understand the needs or store the outputs of our researches. From this point of view, when we say “data”, we mean the all outputs that are taken from any research which are being hold right now or were already held. By the definition, we can deduce that this requires storage facilities for a huge amount of data. Engineers are continuously working on handling the data and maintaining resources for the on-going researches in this way. For these purposes, data distribution is a powerful technique which is commonly used all around the world because by distributing the data to different points, one both avoids overloads on a single source point and in this way satisfies the demand from different points of the world better. Of course, data distribution brings out some security related questions with itself but data encryption and replication are some other techniques to solve such problems. Our project, GenoDist, aims to create an off the shelf component that a developer can apply these modern techniques to his data storage facilities easily.

1.1.Description

GenoDist aims to provide fast, safe and secure data management online. The data which is planned to be deal with is huge that can be expressed even in petabytes, and there is an abstract data for each data set which contains brief information about real data.

In order to be capable of dealing with such big data, it will be distributed among multiple servers. Since whole data will not be used by a client, only abstracts of the data sets will be stored in client side. However, all of the clients should know about the content of the real data, so the abstracts will be stored as block-chain in clients, which is a data

structure used for storing continuously growing list of records without tampering or revision. By that way, clients can see the content of the real data and download whatever part of the data they need, manipulate it and upload it back to servers. But, the data in the block-chain cannot be manipulated by clients. In case of data addition, a block contains the abstract information will be added synchronously to block-chains on clients and the actual data will be distributed among servers.

For the data safety, the GenoDist will also have data backups to deal with arbitrary server failures. Also, to provide security for users' data, the data to be stored will be encrypted not to allow 3rd party users to access it. Also, the connections between connections and clients should not leave a back door for unauthorized people.

1.2.Constraints

- The bandwidth requirements of the blockchain must be lower than the capacities of about 75% of the network nodes in the system. (If the requirements are too high, only some of the nodes will be able to process blocks which will lead to centralization of control.)
- The timestamp contained in each block, which indicates its creation time, must have at most three hours of deviation.
- There must be a uniform access among all network nodes such that no network node must wait for another node to transmit more than one block.
- The blockchain must not accept a transaction which has more data than the maximum size of a block.
- The blockchain must not accept a block which does not a contain a particular key which indicates its validity.

- The time required for creating a new block must be less than 20 minutes.
- The block size must not be more than 1 MB.
- The blockchain size must not be more than 40 GB.
- The contents of the transactions must not be open to third parties.
- There must not be any method for decrypting the cryptography algorithm used for transactions except for brute force attack.
- Brute force attack to decrypt the cryptography algorithm must theoretically require at least 10000 years to execute.
- The speed of encryption algorithm must be at least 20 MB/s.

1.3.Professional and Ethical Issues

1.3.1.Data Encryption

Our software product will deal with large amount of human genome data. Since the data contains confidential information, we will use data encryption to prevent the unauthorized access and thus make our system trustworthy and reliable. In this way, only the authorized people will be able to access and utilize the data. In order to provide secure login authentication, the system should verify the identity of users by using their username and password information. Hence, each user should provide their username and password information to the system before starting to use it.

1.3.2. Data Distribution

We will distribute the data across multiple servers, which are located at different physical locations, to provide users with faster access time. The purpose of distributing the data is to process the data transaction requests on multiple servers instead of one server. By improving the performance of the back-end side to reduce the data transaction time, we are aiming to enhance the user experience at the end-user side.

1.3.3. Data Replication

In order to keep the database updated, we will use data replication on the servers. If a change has been made on a server, then the data replication updates the other servers to apply that change on them. The idea of data replication is to keep the data up-to-date in all distributed server locations.

2. Requirements

2.1.Functional Requirements

- GenoDist is a two-level database management system which is based on a distributed architecture. On the top level, blockchains, a distributed database, stores growing lists of records. These records provide information about how to access to data and summary of data. Each user must have access to blockchains. On the lower level, a distributed system stores the actual data on servers which must be hidden from users.
- Depending on the location of the request and content of the requested data, the most suitable server, in terms of distance and availability of data, must respond to the request.
- During an erroneous situation on a server, the flow of data continues operating via other servers.
- GenoDist encrypts the data to increase security of the system.

2.2.Non-Functional Requirements

- *Availability*: The amount of time the system is operational shall be increased with the distributed architecture compared to centralized architecture via distribution and replication of data to servers and quick recovery procedure when a failure is detected.
- *Continuity*: The system shall be able to handle major interruptions such as power outages of a server by continuing operation on other services.
- *Portability*: The system shall be usable in different environments.

- *Recoverability*: The system shall restore the data in the event of corruption or loss by copying the replicated data from other servers.
- *Response time:* Via distributed architecture, response time shall be decreased compared to centralized architecture by locating the request location and sending the data to requester from nearest server.
- *Reusability*: The system shall be reused across multiple products to store and retrieve the data.
- *Robustness*: The system shall be able to cope with the errors during execution time.
- *Scalability*: The system shall be practical and efficient when applied to large input data by adding new resources.
- *Security*: Security problem of the distributed systems shall be solved by encrypting the data.
- *Transparency*: The block chain part of the system shall be transparent while the resources where actual data is stored shall be hidden from user.
- *Resource constraints:* To meet the work load of the system minimum hardware requirements for each machine should be:
    - *Minimum disk space:* 40 GB [4]
    - *RAM:* 4 GB [4]
    - *CPU: 2 cores [4]*

3. References

[1] Greenspan, Gideon. "Blockchains Vs Centralized Databases | Multichain". *Multichain.com*. N.p., 2016. Web. 8 Oct. 2016.

[2] "How Might We Use Blockchains Outside Cryptocurrencies?". *Jenitennison.com*. N.p., 2016. Web. 8 Oct. 2016.

[3] "Performance Analysis Of Data Encryption Algorithms". *Cs.wustl.edu*. N.p., 2016. Web. 8 Oct. 2016.

[4] M. E., “System Requirements,” *Remote Windows Desktop Management and Administration Software -*. [Online]. Available: https://www.manageengine.com/products/desktop-central/system-requirements.html. [Accessed: 07-Oct-2016].