International Journal of Advanced Culture Technology Vol.9 No.3 321-326 (2021)
DOI https://doi.org/10.17703/IJACT.2021.9.3.321

IJACT 21-9-42

A Study on a Distributed Data Fabric-based Platform in a Multi-Cloud Environment

[1]Seok-Jae Moon, [2]Seong-Beom Kang, [3]Byung-Joon Park

[1]Professor, Institute of Information Technology, Kwangwoon University, Seoul, Korea
[2]School of Software, Kwangwoon University, Korea
[3]Professor, Department of Computer Science, Kwangwoon University, Korea
{msj8086, sbkang707, bjpark}@kw.ac.kr

Abstract

In a multi-cloud environment, it is necessary to minimize physical movement for efficient interoperability of distributed source data without building a data warehouse or data lake. And there is a need for a data platform that can easily access data anywhere in a multi-cloud environment. In this paper, we propose a new platform based on data fabric centered on a distributed platform suitable for cloud environments that overcomes the limitations of legacy systems. This platform applies the knowledge graph database technique to the physical linkage of source data for interoperability of distributed data. And by integrating all data into one scalable platform in a multi-cloud environment, it uses the holochain technique so that companies can easily access and move data with security and authority guaranteed regardless of where the data is stored. The knowledge graph database mitigates the problem of heterogeneous conflicts of data interoperability in a decentralized environment, and Holochain accelerates the memory and security processing process on traditional blockchains. In this way, data access and sharing of more distributed data interoperability becomes flexible, and metadata matching flexibility is effectively handled.

Keywords: *Multi-cloud environment, Data fabric, Knowledge graph database, Interoperability, Metadata.*

1. INTRODUCTION

In the era of big data, one of the pillars of the 4th industrial revolution, data that is difficult to grasp has already accumulated and the cloud has been introduced [1]. However, it only becomes more complicated and the accessibility is gradually decreasing. This means that as the enterprise expands, it will take advantage of a greater number of cutting-edge applications, but the storage system still relies on legacy legacy systems. And silos data makes it stagnant and inaccessible over time [2]. As a result, problems such as low productivity and efficiency, data accessibility, reliability for storage and security, and scalability occur. Big data centers and clouds also continue to divide and classify data, so in the end, visibility and accessibility are poor, so it cannot be a solution to this problem. There is a need to modernize silos and silos storage, not just focusing on advanced applications anymore, but on how to store the data it accumulates. When talking about ways to solve these problems, 'Data Fabric' would be the most appropriate [3]. For years, companies have been wanting to consolidate their data into one scalable platform, and Data Fabric is the all-encompassing way to do this. Data Fabric leverages the hybrid cloud to deliver a hybrid multi-cloud experience and modernizes storage through data management [4]. In other words, Data Fabric integrates all data into one scalable platform to provide an

Manuscript received: August 11, 2021 / revised: September 1, 2021 / accepted: September 4, 2021
Corresponding Author: sbkang707@kw.ac.kr
Tel:+82-01-4727-6815, Fax: +82-2-916-4751
School of Software, Kwangwoon University, Korea.

environment where companies can easily access and move data with security and authority guaranteed regardless of where the data is stored. In this paper, we propose a new data fabric-based platform centered on a distributed platform suitable for a multi-cloud environment that overcomes the limitations of existing legacy systems. It is configured to provide Client Interface, Data Builder, Data Fabric Zone, and Data Sources. In addition, in this paper, we use the knowledge graph database and holochain based on data fabric, a distributed environment platform that connects all data, to enhance the differentiation of development technology. Knowledge graph database [5, 6] connects distributed source data without building a data warehouse or data lake by minimizing physical movement for interoperability based on metadata. This knowledge graph database provides easy access to data from anywhere in a multi-cloud environment. In addition, by integrating all distributed data into one scalable platform, the holochain [7, 8] technique was applied to an environment where companies can easily access and move data with security and authority guaranteed regardless of the storage location of the data. Holochain [7, 8] is a technology that includes agent-centered microservices to compensate for the shortcomings of block chains, and is designed to change the data agent-centered, distributed hash table-centered, and consensus method for each device in this proposed platform. This paper is organized as follows. Chapter 2 describes the overview and operation principle of the proposed system, and Chapter 3 describes the knowledge graph database and Holochain application technique. Chapter 4 describes the performance analysis of the proposed system, and finally, Chapter 5 concludes.

2. PROPOSAL SYSTEM AND OPERATION PRINCIPLE

Figure 1 is the configuration of the data infrastructure construction platform optimized for the distributed environment based on the data fabric of the distributed system proposed in this paper. This platform integrates all data into a single extension in a multi-cloud environment so that companies can easily access and move data with security and authority guaranteed regardless of the data storage location.

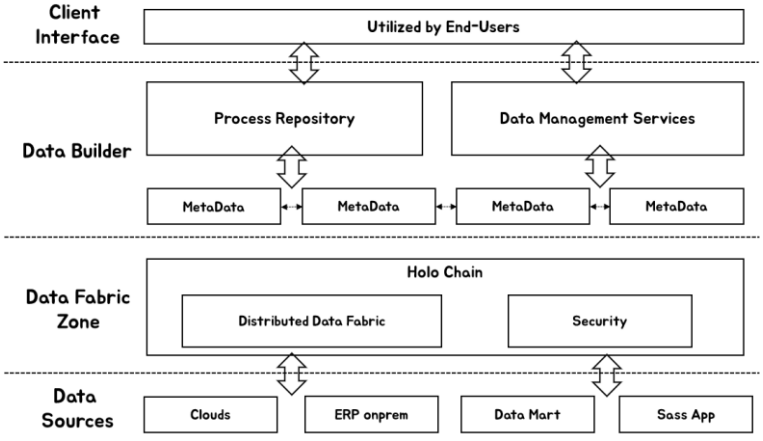

Figure 1. Proposal System Overview

The description of each component inside the proposed platform is as follows.

- Client Interface: A web interface that provides services to users, receives distributed data from the data builder, establishes the relationship between data, and delivers the set relationship to the data builder. It serves to deliver the data requested by the user to the data builder and then provide the retrieved data to the user.
- Data Builder: As a backend system that provides services to the client interface, it receives distributed data from the data fabric zone and provides data to the client interface to establish the interrelationship between data, and stores and manages the set relationship in the knowledge graph database do. It receives a data search request from the client interface, retrieves the relationship information between data sources necessary for the search from the knowledge database, integrates it with the data provided from the data sources, and provides

it to the client interface.

- Data Fabric Zone: As one of the key elements of this study, it is responsible for communication with each distributed data source by integrating the performance and storage that occur when the process of using data source sources is performed in a hybrid cloud environment from a business point of view. It serves as a window to the data builder and retrieves the data requested from the data builder from the relevant data source and delivers it. Manage end-to-end distributed platform support to efficiently support sharing and access between source data in an integrated distributed environment.
- Data Sources: This is the area of existence of the database where data source sources such as Clouds, ERP, Data Mart (DW), and Sass, which are the existing operating methods, are managed.

As shown in Figure 2, in the process of retrieving the desired data from the client interface, select the desired tables from the list of tables already registered. The process of displaying the table and column list along with the relationship already established between the selected tables on the screen and the process of selecting the columns of the table to be searched are carried out. When all tables and columns to be searched are selected, the request is transmitted to the Backend System through the API and the searched content is provided. And, the data builder of Figure 1 analyzes the search data requested from the client interface and classifies the integrated request content by data source required to search each data. After that, the necessary data search request is made to the DNA of the Data Fabric Zone in charge of each source, and the relationship information between the source data is searched from the Knowledge Graph Database [6]. It functions to transmit the integrated search result to the client interface by applying the searched relationship information to the data provided from DNA.

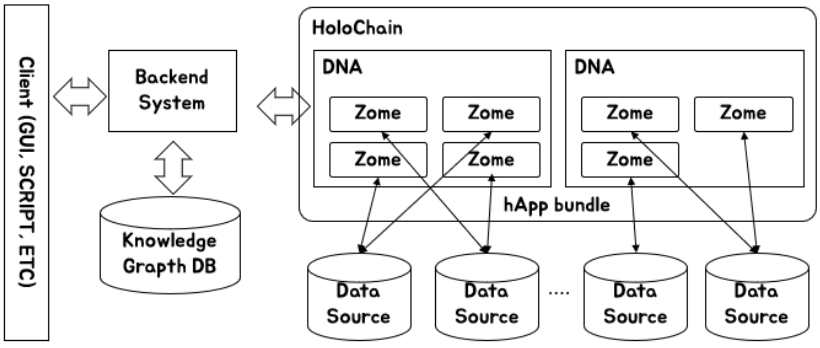

Figure 2. System Principle of Operation

The Data Fabric Zone uses the Holochain framework [8], and the hApp bundle manages the management of distributed data sources rather than a single complex monolithic application. It is a microservice-type agent, and DNA is responsible for each data source independently. As shown in Figure 2, each DNA is composed of one or more execution units, Zome, and serves as a window in charge of communication with each distributed data source. It shares/backs up each other's data between DNAs to support the function to retrieve and deliver the specified data requested by the application from the data source and the resilience of the entire system. Afterwards, when a specific DNA becomes inoperable, other DNA functions to perform it instead. Each Zome collects necessary data from specified specific data sources and delivers it to the DNA it belongs to. And, in addition to the function to take charge of all necessary tasks related to the data source and data storage, it also provides data encryption during transmission between DNAs to maximize system security.

3. KNOWLEDGE GRAPH DATABASE AND HOLOCHAIN APPLICATION

Figure 3 shows how the relationship between tables is designed and stored in the knowledge graph database [6, 10] in the distributed data source in this paper.

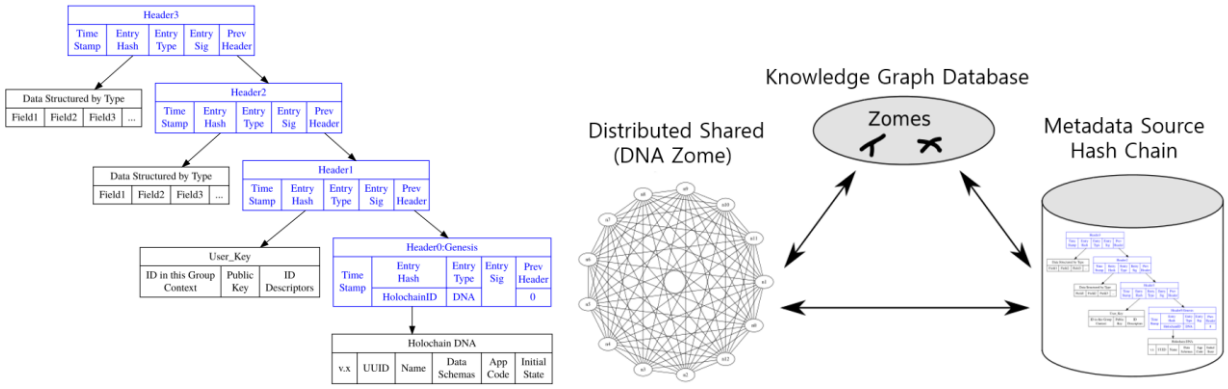

Figure 3. Knowledge Graph Database Operation Principles

If you select the headers to designate the table relationship from the list in Figure 3, each header becomes one node. If you select the node, the important column list of the node is expanded, and if you click it again, the column list is included. To designate columns connected between two tables, select the column list of each table, and then assign the column to the nearest table. When you want to designate more than one column, repeat the same operation as when you designate the first column. When the relationship between all tables (headers) is defined, it is transferred to the Data Builder of Figure 4 and stored in the knowledge graph database.

Algorithm 1. A step-by-step implementation of the holochain framework

1: hApp setup and verification process setup
2: Initializing DNA: Design DNA as the first entity in the legacy system (LS) $LhApp_i$,
the local source chain LSlocal for each $LhApp_i$. i=1, 2, 3, ..., n
3: def create DNA in LSlocal():
4: Set the metadata entity type as a validation rule.
5: Set executable function fm for metadata of specific $hApp_i$
6: Set the other expected parameter metadata to be unique $hApp_i$. m=1, 2, 3, 4, ..., n. 7:
8: DNA Zome generation capability for $hApp_i$.
9: Initialization Generate: Second entity in the local source chain LSlocal.
10: def create Zome in LSlocal():
11: Calculate the timestamp time of Zome creation.
12: Initialize private and public key set IDs (P mrk, P mbk).
13: Calculate the hash value of Zome.

14: def create holochiain metadata():
15: Create a new holochain metadata entity based on hash chain.

16: def demands for a new transaction_entity():
17: Calculate the timestamp of the new metadata entity.
18: Specifies a new item type.
19: Calculate your electronic signature using step 23.
20: Generate current data hash.
21: Calculate the hash of the previous header hashmetadata.
22: Storing signed metadata entities in LSlocal before broadcasting.

23: def creating a new holochain metadata entity():
24: Cryptographically sign each hashchain metadata entity.
25: For each new metadata entity in Holochain.
26: Calculate the electronic signature of the transaction using the agent's private key.
27: Store signed metadata entities in hashchain based LSlocal before broadcast.

Figure 4. A Step-by-Step Implementation Holochain Framework

The method of applying Holochain to the proposed platform in this paper is the same as <Algorithm 1. Configure *h*App bundle verification protocol. Design the DNA as a local source chain LSlocal first entity to the legacy system (LS) L*h*App$_i$ to initialize the DNA. where *i* is *i = 1, 2, 3, ...* means *N*. The process of generating DNA in the LSlocal function sets the entity type as a validation rule. After setting, set the executable function *efx* for a specific *h*App$_i$. And set the other expected parameter x to specify the unique *h*App*i*. Initialize the second entity of the local source chain LSlocal to create a DNA Zome for *h*App$_i$. Calculate the timestamp time of creating a Zome in the LSlocal function. Then, the private and public key set IDs (*P mrk, P mbk*) are initialized. And calculate the hash value of Zome. The metadata holochain creation function creates a new holochain metadata entity based on the hash chain. Functions that require a new transactional entity compute the timestamp of the new metadata entity. Then, specify the new item type, and calculate the 25-step digital signature. Thereafter, a hash of the current data is generated, and a hash of the previous header hash metadata is calculated. Save the signed metadata entity to LSlocal before broadcasting. The new Holochain metadata entity creation function cryptographically signs each hashchain metadata entity. After that, the digital signature of the transaction is calculated and stored by using the agent's private key for each new metadata entity of the holochain.

4. PERFORMANCE ANALYSIS

Figure 4 shows a comparative analysis of the performance of popular DLT in security from the perspective of applying Holochain on this platform. Memory requirements are more important than CPU cycle time because the functionality of DLT is different from traditional encryption mechanisms.

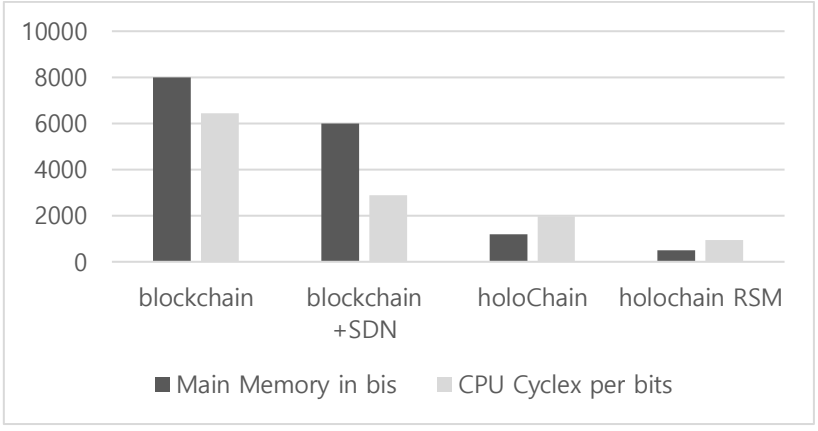

Figure 5. Comparative Analysis of Blockchain and Holochain-based Encryption Mechanisms in Networks

Figure 5, it was described that the hybrid technology of SDN (Software-Defined Network) and blockchain provides better performance compared to the existing blockchain. While blockchain includes requests from all users, SDN is a technology that guarantees secure connections and prevents unnecessary requests, reducing memory and CPU cycles per bit. Although this technology offers a new breakthrough in the world of blockchain, memory requirements and processing technology are still issues. However, holochain and a new

version of holochain (Holochain RSM) reduce the enormous data processing and storage load in dynamic and real-time implementations such as networks. In addition, the memory utilization and speed of holochain are fast.

5. CONCLUSION

We proposed a new platform based on Data Fabric focusing on a distributed platform suitable for multi-cloud environments. In addition, in this paper, data connection and security were strengthened by applying knowledge graph database and holochain framework based on data fabric, a distributed data environment platform that connects all data. Knowledge graph database minimizes physical movement for interoperability of distributed source data based on metadata, and holochain enables easy data access and movement with security and authority guaranteed regardless of the data storage location made it. The knowledge graph database alleviated heterogeneous problems in data interoperability in a distributed environment, and Holochain accelerated the memory and security processing process in the existing block chain. As a future study, it is possible to integrate distributed data in a multi-cloud environment on this platform, but the task of storage remains.

REFERENCES

[1] Skilton, Mark, and Felix Hovsepian. The 4th industrial revolution. Springer Nature, 2018.
[2] Patel, Jayesh. "Bridging data silos using Big Data integration." International Journal of Database Management Systems 11.2/3, 2019.
[3] Alvord, Micah M., et al. "Big Data Fabric Architecture: How Big Data and Data Management Frameworks Converge to Bring a New Generation of Competitive Advantage for Enterprises."
[4] Yang, Xi, and Tom Lehman. "Model driven advanced hybrid cloud services for big data: Paradigm and practice." 2016 Seventh International Workshop on Data-Intensive Computing in the Clouds (DataCloud). IEEE, 2016.Ling Zhang, System and circuit design techniques for wlan-enabled multi-standard receiver, pp.85.2005.
[5] Weedon, Daniel B., and Daniel Olsen. "Knowledge Graph.", 2021.
[6] Menon, Angiras, Nenad B. Krdzavac, and Markus Kraft. "From database to knowledge graph—using data in chemistry." Current Opinion in Chemical Engineering 26: pp33-37, 2019.
[7] Zaman, Shakila, et al. "Thinking Out of the Blocks: Holochain for Distributed Security in IoT Healthcare." arXiv preprint arXiv:2103.01322, 2021.
[8] Fritsch, Felix, et al. "Challenges and Approaches to Scaling the Global Commons." Frontiers in Blockchain 4, 9, 2021.
[9] Holochain zome, https://holochain.github.io/holochain-rust/zome/welcome.html
[10] Lee, Jong-Sub, and Seok-Jae Moon. "Business Collaborative System Based on Social Network Using MOXMDR-DAI+." *International Journal of Advanced Culture Technology* 8.3, pp223-230, 2020. DOI https://doi.org/10.17703/IJACT.2020.8.3.223.