International Journal of Internet, Broadcasting and Communication Vol.8 No.1 83-90 (2016)
http://dx.doi.org/10.7236/IJIBC.2016.8.1.83

IJIBC 16-1-11

A Design of the Cloud Aggregator on the MapReduce in the Multi Cloud

Chigon Hwang*, Hyoyoung Shin**, Jong-Yong Lee***, Kye-Dong Jung3***

**Dept. of Internet Information, Kyungmin College, Korea*
***Dept. of IT Security, Kyungbok University, Namyangju, Korea*
****Department of Electronic Engineering, Kwangwoon University, 20 Kwangwoon-ro,*
Nowon-gu, Seoul 139-701, korea
e-mail : duck1052@kw.ac.kr, hyshin@kbu.ac.kr, jyonglee@kw.ac.kr, gdchung@kw.ac.kr

Abstract

The emergence of cloud has been able to provide a variety of IT service to the user. As organizations and companies are increased that provide these cloud service, many problems arises on integration. However, with the advent of latest technologies such as big data, document-oriented database, and MapReduce, this problem can be easily solved. This paper is intended to design the Cloud Aggregator to provide them as a service to collect information of the cloud system providing each service. To do this, we use the DBaaS(DataBase as a Service) and MapReduce techniques. This makes it possible to maintain the functionality of existing system and correct the problem that may occur depending on the combination.

Keywords: *Cloud Computing, Multi-Cloud, Cloud Aggregator, XMDR(eXtended Meta Data Registry), MapReduce, DBaaS(DataBase as a Service)*

1. Introduction

Cloud Virtualization is a technology that allows the consumer to use the computing environment through the service rather than building the environment. Accordingly, Virtualization and cloud are recognized in the same concept[1]. Cloud computing may utilize a large amount of computing power or a service at a lower cost, and their program and service can be easily built on top of the platform of the external[2]. With these cloud computing advantages, many conventional system has led to the introduction of cloud computing. Therefore, the service that integrates service provided by cloud system is required[3]. This form is called multi cloud, and multi cloud system has the problem of interoperability, data heterogeneity, and data migration between the platform and service[4]. With the advent of smart phone and SNS, new methodology have emerged, such as Big Data, Document-oriented database, and MapReduce. This is a way to solve the above problems[5][6]. In this paper, we propose a method for configuring cloud aggregator for multi cloud, and propose how to apply the above technique. Data transfer between cloud uses a document-oriented database processing technologies such as JSON(JavaScript Object Notation)[7]. In order to solve the

Manuscript Received: Jan. 25, 2016 / Revised: Feb.3, 2016/ Accepted: Feb. 11, 2016
Corresponding Author: gdchung@kw.ac.kr
Tel: +82-2-940-5288
Department of Electronic Engineering, Kwangwoon University, Korea

problem of the heterogeneity and the data necessary for interoperability, it uses knowledge base using XMDR(eXtended Meta Data Registry)[8]. The problem of increasing the amount of redundant data from the data generated by the multi cloud are resolved using the MapReduce techniques. The paper is organized as follows. Chapter 2 describes the related research, and Chapter 3 describes the configuration of the system and Cloud Aggregator. Chapter 4 shows the application examples and comparative evaluation. Chapter 5 describes the conclusions.

2. Related Research

Cloud computing is a technique that provides all of the computing resources as a service. A system that allows you to use a service provided by various cloud system is called multi cloud system. This system has a number of problems. Praraiso described about service migration between the cloud and described about multi cloud considering distributed service[9]. Ferry insists that multi cloud on cloud solutions are not compatible, and increase the complexity of development and management, and has a problem that hinder interoperability[4]. Alomari emphasizes that cloud computing is one of the growing challenges of interoperability and portability between platform, with the rapid spread of the number of applications that are hosted by the cloud system. He designed a tool to support the exchange of NoSQL-based data(Google Datastore, Amazon SimpleDB and MongoDB)[10]. In other words, the biggest problem of multi cloud is how to solve the interoperability between cloud. This paper proposes Cloud Aggregtor using the DBaaS to support interoperability. DBaaS is a service operation technique that provides a database as a service to the service customer on the cloud[11]. We can solve the interoperability and heterogeneity problems in data migration by building a knowledge base on DBaaS based on XMDR[8].Accordingly, the transmission of data utilizes the dynamic schema type document format such as JSON being used in NoSQL[7]. This is because the existing traditional table-based relational database structures need not be used, in accordance with utilizing the JSON. And JSON may provide data integration to a particular application easier and faster[10]. Integrated data can process large amounts of data through the MapReduce[12] method. The removal of the duplicated data and meaningless data is possible based on this.

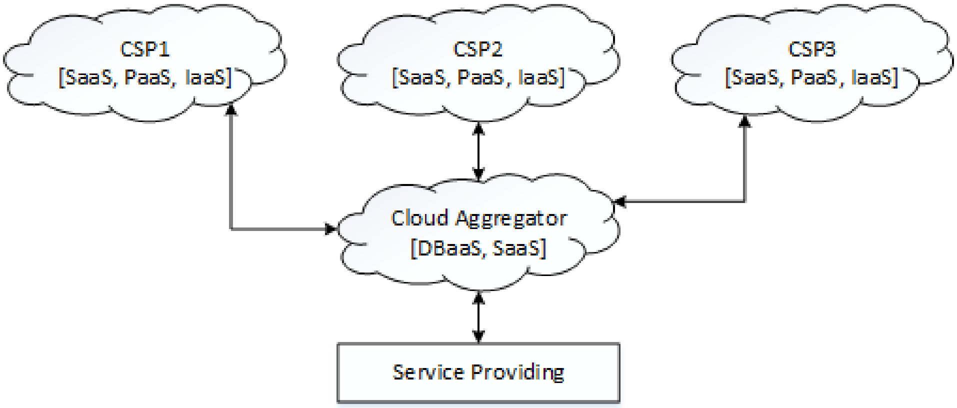

Figure 1. Location of Cloud Aggregator

3. Cloud Aggregator(CA)

The proposed CA integrate data for service to be provided by each cloud system using MapReduce method and it provides an integrated service through the data. Database integrating service is in charge DBaaS, and use of the service is in charge of the SaaS. Applying the necessary MapReduce, it removes the meaningless and redundant data. Configuration of the CA is shown in Figure 2.

3.1 DBaaS(DataBase as a Service)

This section describes the structure and the components of the DBaaS in CA. DBaaS extracts database information on each of the cloud, and provides the results to the user through the MapReduce method.

● CSP(Cloud Service Provider) Controller : This manages the service provider which are provided by cloud system participating in multi cloud.

● Heterogeneity Manager : This is the heterogeneity administrator to resolve the problems between the participating cloud.

● XMDR Manager : This controls the knowledge base so that the Heterogeneity Manager can resolve heterogeneity problem.

● Adapter Manager : This manages adapter installed on participating cloud. Adapter will convert the transmission requirement from the CA to be suitable for the type of provider and requirer. It converts the results to be passed to the CA with JSON document format.

● JSON Parser : The parser analyzes the JSON document sent from participating cloud and save on the Proxy.

● Converter : This changes the query appropriately to cloud for requesting the information and service. It performs a conversion operation for integrating the result.

● MapReducer : This eliminates duplicated data, meaningless data in the collected results, and serves to integrate into one. The process is further described in section 3.

● Proxy : This temporarily stores the collected results, and provides in accordance with the requirements of the user.

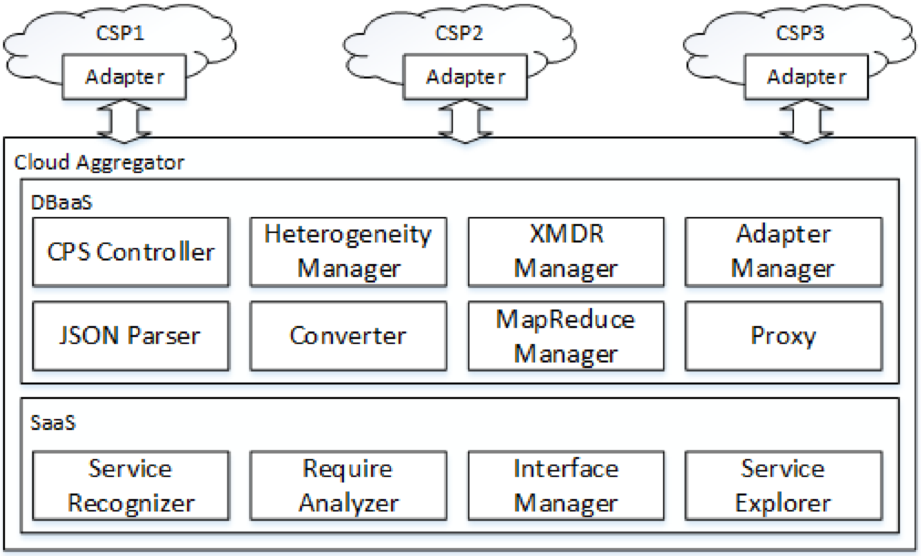

Figure 2. The Cloud Aggregator of proposed system

3.2 SaaS(Software as a Service)

SaaS areas that provide software (application) as a service in this paper is not directly configure the software to provide to the user, but recognize the needs of the user, find the service, configure the user interface, analyze the needs and user demands, and serves to link the software to the user.

● Service Recognizer : This recognize that the service will be suited to the needs of the user.

● Require Analyzer : This analyzes the needs of the user and allows them to find the appropriate service.

● Interface Manager : This will create the interface so that you can use it to find the required service, receives the input request, and prints the result.

● Service Explorer : This is used to search in order to find the service request conditions.

3.3 MapReduce Manager(MRM)

MRM is a service for the integrated management of document-oriented data present in each local database. This service is divided into three roles: the Map, Reduce, Merger. Map serves to convert the structure of document-oriented data and group data. This converts the structure of data in key and value, and groups the data according to the standard. Through this, the data of large amounts can be processed by distributing to each group. The Reduce checks redundancy of duplicate data which is distributed to the group and removes it. This makes it possible to maintain the consistency of the same data. The Merger integrates the data into one which is consisted of several document-oriented data. Alignment is based on the key of the attribute. Figure 3 shows a process of grouping and removing redundancy to handle the big data.

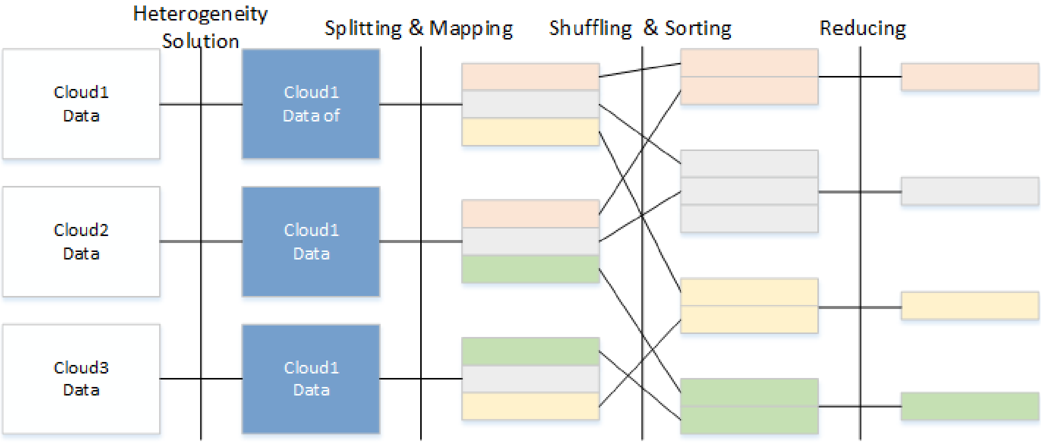

Figure 3. Process course of MapReduce

The process is carried out through the following steps:

● Heterogeneity Solution : This collected data from participating cloud has a structural, symantic heterogeneity because it collected in different system. This step uses XMDR Manager and Heterogeneity Manager to resolve this problem.

● Splitter & Mapper : This separates to a size suitable for MapReduce to remove duplicated and meaningless data of document-oriented data which is created in cloud. It is mapped in accordance with key and function of group.

● Shuffler & Sorter : Groups the duplicated data, based on the key value, and sorts them.

● Reducer : This combines the duplicated data classified by Shuffler and merges them after removing the redundancy of the data.

The manager shows the duplicate entries by examining the collected data only once. The manager eliminates the duplicates in the form of displaying the number of times the item is displayed on the item, and provides the resulting information. The algorithm is shown in Table 3.5.

Table 1. Algorithm of Reduce Process

```
public class groupkey_collection(){
  for var i each this.Data
     if Data[i].chk==1 then continue;
     for var j each this.Data
        if Data[i].key == Data[j].key && Data[j]!=1 then
          Delete Data[j];
          Data[i].count++;
          Data[j].chk=1;
        else
          save Data[j];
       end if
  sort Data;
  save Data;
}
```

4. Application and Evaluation Practices

The proposed system is applied to the Accommodation reservation information, such as on an airplane and hotels for a trip to the configuration of each airline and the CA to perform multi cloud through information provided by the hotel. Analysis of the two sites shows a difference in schema, meaning of data, the data unit, and expression. This should be settled through the MDR manager and Heterogeneity manager. This solves the heterogeneity of the data based on the knowledge base similar to the analysis of the structural and semantic differences. Through removing redundancy and similarity analysis of the Accommodation provided in each cloud, it uses MapReduce techniques to provide inappropriate duplicate search results and duplicate result. Table 2 is extracted as a hotel reservation information collected from sites that provide hotel information as JSON document format. This extracted information is represented by <key>:<value> with a pair <key> and <value>. For example, "hotelname": "OHANA Waikiki East", "hotelname" is the schema information of field indicating the name of the hotel. We know that "OHANA Waikiki East" is instance which represents the name of the hotel. And field name, <key>, is differently collected such as : hotelname" and "hName", "grade" and "star", "review" and "reviewer", "local" and "location". This problem is converted into the same expression through XMDR Manager and heterogeneity Manager. Next, it separates the "hotelname" item, and then reduces redundant data by merging together to perform a shuffling collected information on the same hotel.

Table 2. Search of JSON results for two sites

s	JSON data to searching	site	JSON data to searching

A	{
	“hotels” : [
	{
	“hotelname” : “OHANA Waikiki East”,
	“grade” : 3,
	“price”: “$145”,
	“review” : 3.9,
	“local” : “Koa Avenue, Honolulu”
	},
	{
	“hotelname” : “Hilton Hawaiian Village Waikiki Beach Resort”,
	“grade: 4,
	“price” : “$235”,
	“review” : 4.1,
	“local” : “Waikiki, Honolulu”
	},
	...
	]
	}
		B	{
			“Books” : [
			{
			“hName” : “pacific beach hotel”
			“stars” : 4
			“price” : “KRW 422059”
			“reviewer” : 1371
			“position” : “Waikiki, Honolulu”
			},
			{
			“hName”: “Hilton Hawaiian Village Waikiki Beach Resort”,
			“stars” : 4,
			“price” : “KRW 554,874”,
			“reviewers” : 1388,
			“position” : “Waikiki, Honolulu”
			},
			...
			]
			}

Table 3. Comparison with the proposed method and the others

Methodology Object	Ontology	MapReduce	Proposed Method
Count of Result	No Change	Reduce	Reduce
Semantic Consideration	Considered	Not considered	Considered
Redundancy Remove	Some removed	Almost removed	Almost removed
Integrating methodology	Data-driven	Algorithms-driven	Both

Table 3 is a comparative assessment of the proposed scheme with different integration methods. We compare the integration scheme using ontology[8] and MapReduce method[12] which is the data processing method of Big Data. We compared the differences and similarities to the following: the variation of results for the search results, consideration of the semantic heterogeneity, the degree of removal of duplicate data, and similarities and differences with respect to the techniques for integration.

5. Conclusion

We proposed the Cloud Aggregator. Cloud Aggregator services can provide a fusion of traditional or cloud system combining existing cloud system. This can be maintained without changing the existing system and has the advantage of scalability, convenience of access and transparency. As a way to integrate cloud system, document-oriented database processing techniques and MapReduce techniques were used.

References

[1] Kovári, A., & Dukan, P. (2012, September). KVM & OpenVZ virtualization based IaaS open source cloud virtualization platforms: OpenNode, Proxmox VE. In Intelligent Systems and Informatics (SISY), 2012 IEEE 10th Jubilee International Symposium on (pp. 335-339). IEEE.

[2] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), 599-616.

[3] Gong, C., Liu, J., Zhang, Q., Chen, H., & Gong, Z. (2010, September). The characteristics of cloud computing. In Parallel Processing Workshops (ICPPW), 2010 39th International Conference on (pp. 275-279). IEEE.

[4] Ferry, N., Rossini, A., Chauvel, F., Morin, B., & Solberg, A. (2013, June). Towards model-driven provisioning, deployment, monitoring, and adaptation of multi-cloud systems. In Cloud Computing (CLOUD), 2013 IEEE Sixth International Conference on (pp. 887-894). IEEE.

[5] Aisling O'Driscolla, Jurate Daugelaiteb, Roy D.Sleator, 'Big data', Hadoop and cloud computing in genomic, Journal of Biomedical Informatics, 46, 5, pp.369-392, 2014.

[6] Fernández, A., del Río, S., López, V., Bawakid, A., del Jesus, M. J., Benítez, J. M., & Herrera, F. (2014). Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, 4(5), 380-409.

[7] Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

[8] Kye-Dong Jung, Chi-Gon Hwang, Jong-Yong Lee, Hyo-Young Shin, "The Study of DBaaS Hub System for Integration of Database In the Cloud Environment", JOURNAL OF DIGITAL CONVERGENCE, 12, 9, pp.201-207, 2014.09.

[9] Paraiso, F., Haderer, N., Merle, P., Rouvoy, R., & Seinturier, L. (2012, June). A federated multi-cloud PaaS infrastructure. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on (pp. 392-399). IEEE.

[10] Alomari, E., Barnawi, A., & Sakr, S. (2014, December). CDPort: a framework of data portability in cloud platforms. In Proceedings of the 16th International Conference on Information Integration and Web-based Applications & Services (pp. 126-133). ACM.

[11] 11. Malliga, P. (2012). Database services for cloud computing–an overview. International Journal of Computers & Technology, 2(3).

[12] Yang, H. C., Dasdan, A., Hsiao, R. L., & Parker, D. S. (2007, June). Map-reduce-merge: simplified relational data processing on large clusters. In Proceedings of the 2007 ACM SIGMOD international conference on Management of data (pp. 1029-1040). ACM.