

# Matrix-based Data Center Selection Algorithm for a Federated Cloud

Amol Jaikar<sup>1,2</sup>, Gyeong-Ryoon Kim<sup>2</sup> and Seo-Young Noh<sup>2</sup>

<sup>1</sup> *University of Science and Technology, Daejeon, South Korea*

<sup>2</sup> *Korea Institute of Science and Technology Information, Daejeon, South Korea*

{amol, grkim, rsyoun}@kisti.re.kr

## Abstract

*Cloud Computing is a paradigm to provide the services like Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS). Virtualization technology plays an important role to provide these services. Due to lack of finance as well as requirement, data center providers are unable to establish data centers all over the world. To provide better Quality of Service (QoS) and maintain Service Level Agreement (SLA), data center providers collaborate with other data center which is named as a Federated Cloud. Load balancing process distributes the load or service request among the data centers by implementing distribution policies. Depending on the policies, the central manager chooses the location to deploy the virtual machine. This paper proposes the data center selection algorithm in a federated cloud to optimize the cost as well to improve the performance.*

**Keywords:** *Cloud Computing, Data center, Federated Cloud, Energy efficiency.*

## 1. Introduction

Cloud computing offers a wide range of services at different levels. Infrastructure, platform and software as a service provides to the user to use the data center for large complex computation, data storage and software execution [5]. The cloud computing data center uses virtualization technology to share the computing resources to the outside world through a virtual machine. Each virtual machine has its own characteristics as well as specifications. Depending on the job requirement, each virtual machine executes on the physical server, utilizes its resources and consumes the energy.

There are plenty of physics experiments are going on all over the world. These experiments require huge amount of computational power which is allocated by many data centers. Researchers from all over the world use this computational power to do their research. Static allocation of computing resource has been done from the data center. But some of the researcher does not use these resources all the time. Therefore computational resources will be idle which is allocated by static allocation policies. This problem leads to the under utilization of resources. Therefore dynamic allocation of computational resource will help to reduce the power consumption as well as increase the resource utilization of the data center.

Cloud manager dynamically selects the physical server to launch virtual machines by using administrative or dynamic policies. Each physical server can have multiple virtual machines. Selection of physical server is an important task which affects on performance as well as power consumption of the data center. Selection of best data center is also challenging task. It gets more importance in case of federated cloud

environment, where two or more data centers work together to complete the jobs. Therefore, a federated cloud manager needs to decide the location to deploy the virtual machine in some of the data center. There are two levels of decision, first one is which data center and the second one is which physical server in a data center. Federated cloud is nothing but the collaboration of private and public cloud. Therefore the solution of this problem can be centralized as well as distributed. A centralized system can decide which data center as well as which physical server to deploy the virtual machine. Otherwise one policy will be executed to decide the data center while another policy will be decided to select physical server.

The paper is organized as follows. Second section gives the previous work related to the data center selection as well as a description of the existing load balancing approach. Third section describes the algorithm to select the cost effective as well as better performance data center. Last section gives the conclusion and future work.

## **2. Related Work**

There are many existing algorithms that present to select the best data center in a federated cloud. We will discuss the existing algorithms with its advantages as well as disadvantages. All these algorithms come under centralized system approach. The network plays an important role in a distributed environment. The performance of the distributed service can be affected by network delay. The proximity based routing algorithm selects the closest data center based on the network latency. If there are many data centers in the same region then the algorithm selects the data center randomly. This algorithm does not consider the performance as well as cost [1].

Cost effective selection algorithm extends the service proximity based routing by selecting cost efficient data center. This algorithm does not consider performance as well as availability. It is important to consider the performance and availability in selection process at the same time [2]. The data center selection algorithm can choose the data center randomly. Due to this policy, such an approach increases not only the response time but also the cost of computation. This algorithm can lead to another problem of power consumption. All the data centers must be always ready or waiting to execute jobs. Power consumption will increase because of always active state. Round robin based data center selection algorithm in a single region for CloudAnalyst is proposed [3]. As per author, dynamic and massively distributed environment can be studied by simulation. This algorithm selects the data center for the execution of virtual machines by using the round robin technique. This algorithm can give equal usage of resources from all the data centers within that region without considering performance as well as availability, which again results in heavy power consumption because of always active state.

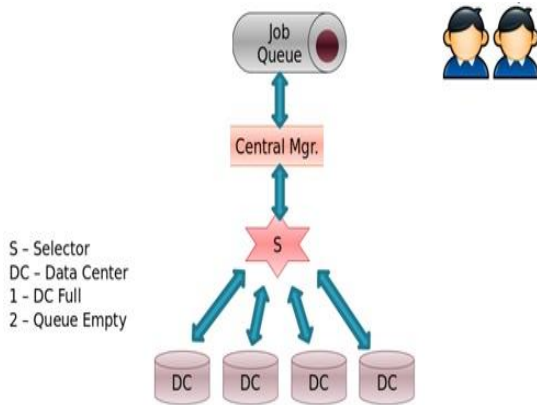
## **3. Distributed vs Centralized System Approach**

In this section, two approaches are described to choose the data center to support the service request. Cloud computing system architecture consists of the entities like central manager, data center, user, virtual machine, processing element (CPU cores) and many more. The user submits jobs while the virtual machine executes the jobs on a physical machine. The data center provides the resources like processing element of a physical machine for execution of a virtual machine. Following subsections give the pros and cons of the centralized as well as distributed approach.

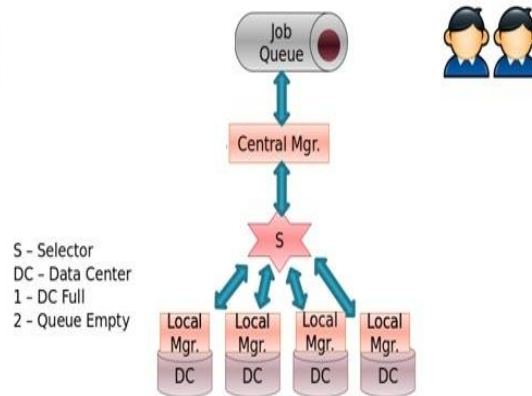
### 3.1. Distributed System Approach

In the distributed approach, two different policy makers take part in the allocation of virtual machines. Figure 1 shows the architecture of the distributed approach. The first central manager decides which data center will serve this request and sends it to local manager to decide where it is going to deploy. After getting information from central manager, the local manager decides which physical machine in a data center will be used to execute that virtual machine. Centralized policies can be executed by central manager and local policies can be executed by the local manager of the data center. Central manager takes the decision and sends the information to the local manager then the local manager can decide where to deploy that virtual machine to execute the job.

Advantages of this distributed system approach are: data center providers can use the resources dynamically for the other jobs if the resources are not fully used by the central manager because the cluster is not statically allocated to the system. On the other hand, a disadvantage is that distributed system is unable to make new policies dynamically depending on the existing performance because central policies cannot interfere or investigate the local policies. Therefore, it is difficult to maintain performance matrix in terms of the distributed system approach.



**Figure 1. Distributed System Architecture**



**Figure 2. Centralized System Architecture**

### 3.2. Centralized System Approach

In a centralized approach, the central manager is responsible for the allocation of the virtual machine to a physical machine of the data center using centralized policies as per Figure 2. These policies will be applicable to the entire data center uniformly. Therefore the data center providers do not need to take care of resource utilization. The cluster is statically allocated to the central system. Data center providers need to allocate the resources as well as to subscribe the central manager of the system. Centralized system reduces the administrator's workload of the data center.

Advantages of this centralized system are including unique policies applicable for all data centers, fare execution of policies; and availability of creating dynamic policies depending on the performance or logs. Creation of dynamic policies can improve the performance. Such characteristic has a capability to change the situation pleasantly. On the other hand, the disadvantage is the bottleneck. Each operation like virtual machine

migration, deletion, creation or consolidation needs to contact central manager which can increase the negligible amount of network traffic. In this paper, we have adopted centralized system approach.

#### 4. Matrix-based Method

As per last subsections, centralized system has many advantages than distributed system as well as centralized system has the capacity to create dynamic policies. In the proposed method, the data center is selected based on the matrix values. These values can be changed by an administrator or by request. Following subsections gives an explanation of the matrix.

##### 4.1. Cost vs Location Matrix

The data center is located in some of the region in the world. This matrix is created for each region. When the request comes from the user to the centralized system, it sees the request region and then access the respective matrix to decide the data center. In a region, there can be many data centers or single data center. If there is a single data center then the system will choose that data center by default. Otherwise, it needs to access the matrix to decide the set of data centers. This matrix helps to reduce the network delay between the user and the data center. Therefore, it can improve the performance of the system. The cost vs location matrix is as shown in Table 1.

This matrix gives the list of data center. Selection is based on the threshold value to optimize the list which means if the distance value is more than threshold value then remove these data centers from the expected data centers. This policy can limit the network delay.

**Table 1. Distance vs Cost Matrix**

Data Center	DC1	DC2
Distance (km)	<50	<100
Cost	0.1\$	0.15\$

##### 4.2. Performance vs Availability Matrix

This matrix is designed for each region. Performance is nothing but jobs completed verses required time to complete. Availability is nothing but the resources available for services. The reason behind this table is to choose a better performance data center as well as good availability. Availability is also an important point to consider for selecting data center. Therefore, performance and availability should be considered as important factors at the same time. There are many factors affect the availability like geographical location, natural disasters, availability of network and many more. The performance vs availability matrix is as shown in the Table 2.

**Table 2. Performance vs Availability**

Data Center	DC1	DC2
Performance (jobs/sec)	1000	800
Availability	98%	95%

### 4.3. Algorithm

The centralized approach has two steps to select the location to deploy the virtual machine. First one is to select the best data center and the second one is to select the physical machine to execute the virtual machine. The algorithm selects the best data center to execute the job. Algorithm 1 can be used for centralized as well as distributed system approach.

<b>Algorithm 1: Matrix-based Data Center Selection Algorithm</b>	
1	while(need) /*If the data center selection is required*/
2	do
3	for $i = 0$ to $n$ /*list of all matrices for each region*/
4	RegCvL[ ][ ] $\leftarrow$ Select the Cost vs Distance matrix
5	end
6	Set1[ ] $\leftarrow$ insert all the candidate data centers from RegCvL[ ][ ]
7	for $i = 0$ to $n$ /*list of all matrices for each region*/
8	RegPvA[ ][ ] $\leftarrow$ Select the Performance vs Availability matrix
9	end
10	Set2[ ] $\leftarrow$ insert all the candidate data centers from RegPvA[ ][ ]
11	Set3[ ] $\leftarrow$ Set1[ ] $\cap$ Set2[ ] //get the common elements
12	DC $\leftarrow$ Select the best data center from Set3[ ]
13	end /*end while loop*/

Step 1 describes that whenever the job comes to execute on the system, it executes this algorithm to decide which data center is perfect for this job. It is a continuous process that is why while loop is used. For each region, the cost vs location matrix is predefined. The performance vs availability matrix is calculated dynamically and updated regularly. After getting request from the specific region, step 4 selects the cost vs location matrix for the respective region. Step 6 selects the data centers which have a low network delay above of the threshold. The selected data center list is kept in the array. After this, step 8 selects the existing performance vs availability matrix for the respective region. From this matrix, the data centers can be selected based on the good performance as well as availability. Due to the threshold value, in step 10, only some of the data centers are selected from an entire matrix which will be stored in another array. Step 11 selects the common data centers from the previous two arrays. Step 12 selects the best data center from the common array created by step 11. After selecting efficient data center for the job execution, system can proceed to next step which is nothing but selection of physical machine to deploy the virtual machine.

### 5. Conclusion and Future Work

The proposed algorithm has capability to give best data center location to execute the job. This algorithm has the capability to design dynamic policies which can improve the performance. The cost vs location matrix helps to get the best price for the same service. It also gives low network delay to improve the response time. The performance vs availability matrix gives the best performed data center. It also gives the availability of the data center which can improve the response time of the job. Taking intersection of these two matrices can provide the best location of the data center.

In future work, we are planning to implement this algorithm and recheck the theoretical as well as practical results. Comparison of these results can able to describe the performance of the algorithm on real systems. Wrong selection of the data center may lead to increase the cost of job execution as well as response time due to unavailability of resources.

## Acknowledgements

This work was supported by the National Research Foundation (NRF) of Korea through contract N-13-NM-IR04.

## References

- [1] D. Limbani and B. Oza, "A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation", *International Journal Computer Technology and Application*, vol. 3, no. 3, (2012) May-June.
- [2] D. Chudasama, N. Trivedi and R. Sinha, "Cost effective selection of Data center by Proximity-Based routing policy for Service Brokering in Cloud Environment", *International Journal Computer Technology and Application*, vol. 3, no. 6, (2012) November-December.
- [3] R. Rathi, V. Sharma and S. K. Bola, "Round-Robin Data Center Selection in Single Region for Service Proximity Service Broker in CloudAnalyst", *International Journal of Computers and Technology*, vol. 4, no. 2, (2013) March-April.
- [4] M. Dash, A. Mahapatra and N. R. Chakraborty, "Cost Effective Selection of Data Center in Cloud Environment", *Special Issue of International journal on Advanced Computer Theory and Engineering*, vol. 2, Issue 1, (2013).
- [5] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of Cloud Computing" *Communications of the ACM*, vol. 53, no. 4, (2010), pp. 50-58.