K-Nearest-Neighbor-Based Service Broker Policy for Data Center Selection in Cloud Computing Environment

Emad AlSukhni
Computer Information Systems Department
Yarmouk University
Irbid, Jordan
ealsukhni@yu.edu.jo

***Abstract*— Cloud Computing offers various resources as services to the internet users. This arise a very important issue in selecting the appropriate data center(s) to serve users' requests. One of the important components in the cloud computing environment is the service broker, which's responsible for selecting an adequate data center to serve a user request. Many of broker policies are implemented in the cloud computing environment. A K-Nearest-Neighbor-Based service broker policy for data center selection in the cloud computing environment is proposed in this paper. This policy used a list of distinctive features and characteristics for each data center and user's request. The results show that the proposed policy outperformed the random-based policy in terms of Average Response Time (ART) and Average Processing Time (APT).**

***Index Terms*— Cloud computing, Data Center, Service Broker Policy, K-Nearest Neighbor.**

IRJECE

I. INTRODUCTION

Cloud Computing Services expand increasingly, providing the user with great prospects of working in flexible way in a dynamic, virtualized, and distributed environment. They allow users to find the best service with resonable cost. Cloud Computing is an internet-based computing technology where computational operations implemented over the cloud [1]. Such environment with high degree of dynamism arises a key question about how to select the most appropriate data centers for the needed service? In cloud computing the bet on minimizing Response Time (ART) and Average Processing Time (APT).

A service broker is responsible for selecting a data center, which provides the service to the user's request. Thus, the service broker controls and distributes the user requests into data centers that are distributed over many regions. This distribution is done according on a selection policy, called data center selection policy [7]. In most cloud computing environment, The service broker is used one of three policies, involves:

- Service Proximity Based Routing: this policy relieson listing data centers based on latency priority from lowest priority. So, the incoming request will be served by the data center that has the lowest latency.
- Dynamically Reconfiguring Routing: this policy extends the proximity policy with extra responsibility that is scaling the deployment of the application based on the load-balance of data center. This policy changes the number of Virtual Machine (VM) located in the data center.
- Performance Optimized Routing: the service broker observes the performance of the data centers at all the time, then, chooses the candidate data center based on the best response time.

In general, the Service Proximity Based Routing selects the closest data center as shown in Figure 1.This policy works as follow:

1. All data centers indexed based on their region. This index is maintained by the service proximity service broker.
2. When a user request is received,, then the service proximity service broker retrieves the region of that request and get back the proximity list for that region.
3. Based on the request characteristics, the service proximity service broker selects the highest data center from the retrieved proximity list. If more than on data centers are found in a same region, one of them will be selected randomly [1].

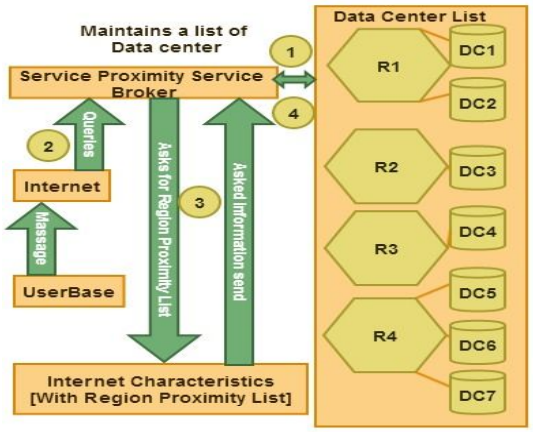


Fig. 1: Service Proximity Based Routing [1]

The problems of random selection in service proximity based routing is mentioned in [1]. It may pick the data center with high workload, high processing or response time, hence, that led to., under utilize the available cloud resources.

The remainder of this paper is organized as follow. In Section II, reviews the related works in this field. In Section III, we present our proposed approach in improving Proximity Based Policy. Section IV, we conclude our work and suggested some future work.

II. Related Works

Problems in service proximity based policy motivate many researchers to propose enhancements for service proximity based policy. In this Section, we provide some of these proposed enhancements.

Devyaniba et.al in [6] proposed an enhanced proximity-based policy. In the proposed policy, the data center with less cost (i.e. Data transfer cost) is considered as a candidate data center, instead of selecting the closest data center immediately.,. This proposal achieved cost effective results.

Dhaval et. al. [8] presented an enhancement on Proximity, in which the candidate data centers depending on its cost-effectiveness. The authors discussed the issues from two perspectives; maximizing efficiency and minimizing cost. Their experiments showed that promising results in terms of the cost. In addition to an improvement in processing time in comparison with the closest data center selection policy.

Sarafaz [9] focused on the enhancement of the response time. He proposed a modification on routing policy by redirecting a part or whole traffic to the next nearest data center in the specific region which results in less overloading.

Mishra et al. in [1] tried to overcome the previous problems by proposing a policy, which depends on Round Robin based approach. By using a simple Round Robin algorithm, the broker selects the data center by distributing various requests among all data centers that exist in the nearest region. They assumed that his proposition will lead to more utilization of resources. They also considered the data centers variation in the speed and cost. So, the faster data center could be selected many times to enhance the performance. They also proposed to assign the data centers with lower cost with more tasks. They showed with the simulation results of effectiveness of their approach.

A. Jaikar et.al [10] proposed an approach for data center selection, which relies on different matrices that are constructed for each region. Based on these matrices, a data center will be preserved or ruled out. When data center characteristics within the limits of predefined thresholds then it will be preserved, otherwise, it will be ruled out. They assumed this algorithm limits the network delay and able to design dynamic policies which can improve the performance.

P.Rani et.al [11] proposed an algorithm for choosing an appropriate data center by computational sequences, whichcalculats the percentage of processor utilization, and then defines the threshold value for every data center according to the preserved efficiency that is listed when a request received; the data center with the highest efficiency would be chosen, their proposed policy results show an improvement in the processing time.

V. Sharma [13] applied Round-Robin load balancing policies for distributing load over multiple data centers available in the same region. Instead of random selection, they proposed to maintain queue for pre-selected data centers. This approach guarantees suitable utilization for all data centers. Their experiments show that the proposed policy works efficiently in term of resource utilization with stability in cost.

III. Proposed K-Nearest-Neighbr-Based Service Broker Policy For Data Center Selection In Cloud

The Service proximity based policy relies on the random selection of the data center, in case there are many data centers in the same region (based on many measures). This caused a number of problems including the possibility of selecting data center with high workload, high processing time, high response time, and may be high cost. Moreover, it may lead to under utilization of the resources. As a result, we propose an approach to selectthe appropriate data center based on the K-Nearest Neighbor algorithm.

A. K-Nearest Neighbor Classifier

The Nearest-neighbor classifier is used in classification which based on learning based on the degree of similarity. That is done by comparing the new incoming tuple with old training tuples that are similar to it. The all tuples are characterized by *n* attributes. So, each tuple represents one point in an *n*-dimensional space. In this approach, all the training tuples are

IRJECE

stored in an *n*-dimensional pattern space. When a new tuple is given, a *k*-nearest-neighbor algorithm searches the pattern space. then lists the *k* training tuples, which are closest to this new tuple (*k* is a predefined constant). The *k* training tuples "Closeness" are defined in terms of some distance metrics, such as Euclidean distance. In other words, for X_1 and X_2 tuples, we calculate the difference between asymmetric attributes in both tuple X_1 and X_2. [14].

B. K-Nearest-Neighbor-Based Service Broker Policy for Data Center Selection in Cloud

In the same manner, we use *k*-nearest-neighbor algorithm in selecting the data center for user requests. As in Algorithm 1 in figure2, we consider the data centers as points in *n*-dimensional space pattern with specific attributes or characteristics. A tuple of similar attributes is specified as thresholds of these attributes.

When a request is received it must attached with the threshold tuple. If the closest region has only one data center, this data center will be selected and assigned to the user's request.

Otherwise, If more than one or less than *k* data center that are existed in the region, one of them will be selected or assigned randomly. In case of more than *k* data then the distance between the data center and incoming user's request can be calculated and then the data center with the minimum distance will be selected and assigned to the user's request.

T: Represents a user request tuple of predetermined thresholds.

D_i: Represents the tuple of characteristics collected for the i^{th} datacenter.

Dis_{i-t}: Represents the distance between T and the datacenter D_i.

For example, if we consider cost and the Response Time as two characteristics to represent the criteria for selecting data center, then we can calculate the distance between T and every data center based on these two criteria, otherwise, if we consider the cost only or the best response time only the result may differs. We expect this approach has the ability to overcome the problems of random selection, in addition to prevent the selection of data center with higher cost, higher response time and so on.

```
Algorithm 1 Data Center Selection
01  Get the DC index of selected region, T,KNN_list
02  Region list ←regionaDataCenterIndex.get(region)
03  K←KNN_list.size();
04  If regionaList is not NULL then
05  Listsize ←size(regionaList)
06  If listsize = 1 then
07  dcName ←regionaList.get(0)
08  Else if listsize <= K then
09  Int rand=(int)(Math.random()*listsize);
10  dcName ←regionalList(rand)
11  Else
12  Int rand=(int)(Math.random()*KNN_list.size());
13  dcName←KNN_list(rand)
14  end If
15  End if
16  Return dcName
```

Fig. 2: Data Center Selection Algorithm.

Algorithm 1 depends on preprocessing algorithm (Algorithm 2), which calculates the distances between T and data centers and then it returns the KNN_list that can be used in data center selection algorithm. The preprocessing algorithm is shown in figure 3.

```
Algorithm 2 Preprocessing
01  Input: DC index of selected region , K
02  Output: KNN_list
03  For All datacenters in a selected region do
04  Calculate distance between T and the datacenter
05  End for
06  Determine K minimum distances
07  Put DC names of K minimum distances in KNN_list
08  Return KNN_list
```

Fig. 3: Preprocessing Algorithm.

IV. SIMULATION AND RESULTS

We use the Cloud-Analyst to evaluate the proposed service proximity based routing algorithm, with Random and KNN-based data center selection. Also, we use the experiment configurations as mentioned in [1]. Table I shows data center configurations, which are the part of the experiment configurations. In our experiment, we implement the same scenarios as in [1]. These configurations are shown as part of Table II.

Table I: Data Center Configuration [1]

Name	Cost/Vm ($/Hr)	Data transfer Cost ($/GB)	Speed (MIPS)
DC1	0.08	0.08	500
DC2	0.09	0.09	700
DC3	0.10	0.10	1000

IRJECE

DC4	**0.11**	**0.11**	**1300**
DC5	**0.12**	**0.12**	**1600**

Now, each the Random and KNN-based selection algorithm has been run with these various configurations. Table II, Figure 4 and Figure 5 show the simulation results.. The average response time and average processing time for all scenarios are obtained with the random and KNN-based selection algorithm.

Table II: Simulation Results

Case	KNN-based		Random	
	ART (ms)	APT (ms)	ART (ms)	APT (ms)
4XDC4	117	64	66	94
DC1,3XDC6	102	48	51	87
2XDC1,2XDC6	110	58	60	108
3XDC1,DC6	129	79	79	150
2XDC3,2XDC6	106	52	53	76
2XDC3,DC4,DC6	106	55	57	79
DC1,DC2,DC3,DC5	125	73	74	122

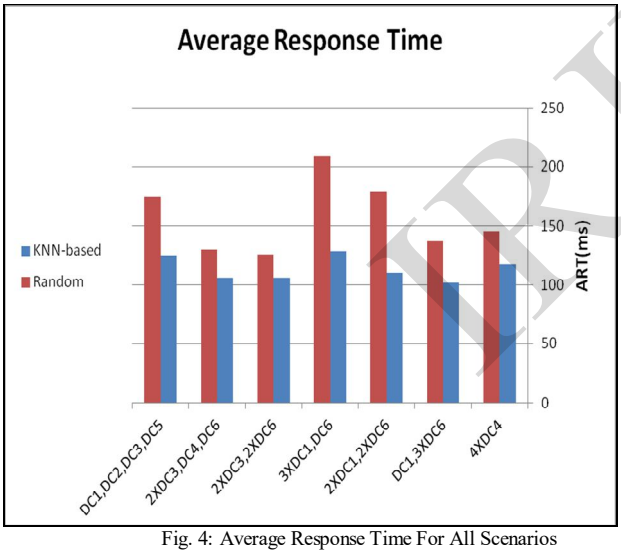


Fig. 4: Average Response Time For All Scenarios

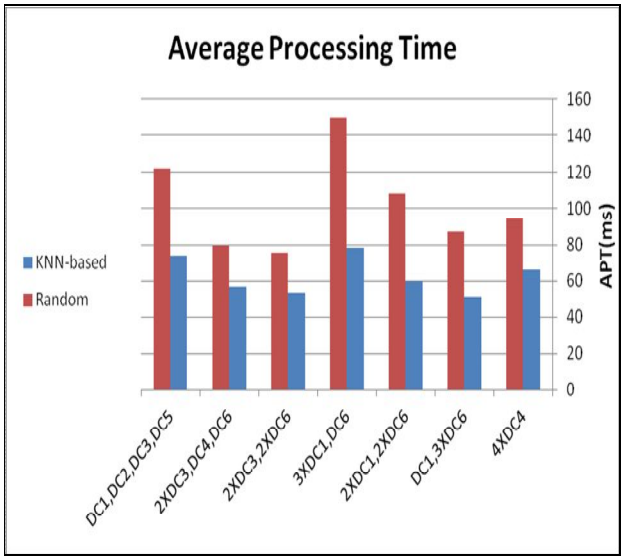


Fig. 5: Average Processing Time For All Scenarios

As shown in Table II and figures 4 and 5, the proposed KNN-based algorithm gives better response time than the Random selection in all scenarios. In addition, the results show that the proposed KNN-based algorithm gives better processing time than the Random selection algorithm in all scenarios. Our approach shows significant improvements in terms of average response time and average processing time in some scenarios.

On the other hand, it shows slight improvements in terms of average response time and average processing time, because of the randomness exists in the proposed algorithm.

V. CONCLUSION AND FUTURE WORK

In this paper, we propose a modification for Proximity Based Routing Policy; we take the advantage of the closest selection feature in proximity with attempting to avoid randomness problems. The proposed approach limits the selection process in a small list that contains the data centers, which have characteristics closed to the user's request characteristics. We assume that the algorithm is able to distribute the user requests into data centers more equitable, and to reduce the affect of random selection. In nutshal, the results show that the proposed policy outperformed the random-based policy in terms of average response time and average processing time.

In Future work, we are planning to study the way in which the value of k is determined.

REFERENCES

[1] Mishra, R. K., & Bhukya, S. N.,"Service Broker Algorithm for Cloud-Analyst", International Journal of Computer Science & Information Technologies, 5(3), 2014.

IRJECE

[2] Mishra, Rakesh Kumar, Sandeep Kumar, and B. Sreenu Naik. "Priority based round-robin service broker algorithm for cloud-analyst." Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014.
[3] Wickremasinghe, B., Calheiros, R. N., & Buyya, R., Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications. In Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on (pp. 446-452). IEEE, 2010.
[4] Sakellari, G., & Loukas, G., A survey of mathematical models, simulation approaches and testbeds used for research in cloud computing. Simulation Modelling Practice and Theory, 39, 92-103.
[5] http://www.cloudbus.org/cloudsim Access in Apr 20, 2016.
[6] Chudasama, D., Trivedi, N., & Sinha, R. Cost effective selection of data center by proximity-based routing policy for service brokering in cloud environment. International journal computer technology and application, 3(6), 2012.
[7] Patel, H. V., & Patel, R. Cloud Analyst: An Insight of Service Broker Policy.
[8] Limbani, D., & Oza, B., A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation. International Journal of Computer Technology & Applications, 3(3), 2012.
[9] Ahmed, A. S., & Ahmed, A. S., Enhanced Proximity-based Routing Policy for Service Brokering in Cloud Computing. International Journal of Engineering Research and Ap-plications, India, 2(2), 1453-1455, 2012.
[10] Jaikar, A., Kim, G. R., & Noh, S. Y. , Matrix-based Data Center Selection Algorithm for a Federated Cloud.International Journal of Multimedia & Ubiquitous Engineering, 9(6), 2014.
[11] PRani,Reena Chauhan and Ritu Chauhan, An Enhancement in Service Broker Policy for Cloud-Analyst.International Journal of Computer Applications (0975 – 8887), 2015.
[12] Selim, G. I., Sadek, R. A., & Taha, H., An Efficient Cloud Service Broker Algorithm. International Journal of Advancements in Computing Technology, 6(1), 2014.
[13] Sharma, V., Efficient Data Center Selection Policy for Service Proximity Service Broker in CloudAnalyst, 2014.
[14] Jiawei Han,Micheline Kamber and Jian Pei, Data MiningConcepts and Techniques, Third Edition.Elsevier Inc, 2011.

IRJECE