A Modified Service Broker Policy Based on Cuckoo Optimization

Sakshi Tiwari[1], Prachi Parwar[2]
[1]Dept of CSE
[2]Assistant Professor, Dept of CSE
[1, 2] Takshila Institute of Engineering & Technology, Jabalpur, Madhya Pradesh, India.

Abstract- *Cloud federation is a group of aggregated providers, who are mutually cooperating and collaborating to share their resources in order to improve each other services. It has lured the attention of commercial industries towards itself for its effective utilization of cloud resources. Effective management of the resource is very much required in order to increase the profits of an individual service provider in federation, but a lack of proper business model hinders service provider in deploying its feature. Several new issues are generated because of these wide adoptions which are concern to both cloud service provider and cloud service users. Cloud service providers are facing the issue of resource limitations. Cloud service users are facing the issue of vendor lock-in. Federated Cloud through brokering can be used to solve the above mentioned issues.*

A cloud-based service broker provides intermediation to seek appropriate service providers in terms a suitable trade-off between price and performance. On the other hand, load balancing among cloud resources ensures efficient use of a physical infrastructure, and at the same time, minimizes execution time. This makes service brokers and load balancing among the most important issues in cloud computing systems. This thesis presents a set of novel market and economics-inspired policies, mechanisms, algorithms, and software designed to address the profit maximization problem of cloud providers.

This paper models and evaluates how the providers can manage the incoming request to changing environments for higher outcomes by means of a new brokering policy. This thesis defines and evaluates a brokering policies method that enforces providers to consider the impact of their decisions in the long term.

Keywords- Cloud Computing, Brokering Policy, Resource sharing, Availability, Cloud Analyst, Cloud Service Broker.

I. INTRODUCTION

Recently, the adoption of multiple clouds for running cloud based applications and services have been considered as a mitigation factor towards the vendor lock-in issue. In addition, a multicloud environment may be beneficial to cloud-based applications in many other ways. For example, some application services may have special functional and/or non-functional demands that cannot be fulfilled by a single target cloud. In this case, considering a multi-cloud scenario is simply mandatory [1]. Moreover, the multicloud scenario can show its advantage in terms of cost-saving for the users: since different services may have different requirements, simply choosing the cheapest provider by considering a single resource may not be cost-effective. Two orthogonal approaches are commonly exploited for addressing deployments across multiple clouds: Cloud Brokering and Cloud Federation [2]. Cloud Brokers can leverage abstraction APIs, such as Apache Libcloud or Delta Cloud for allowing users to exploit different providers at the same time whereas Cloud Federations provide common platforms providers must be compliant with. Even if Cloud Federation may subsume the Cloud Brokering approach, they can be considered orthogonal from the viewpoint of the goals they pursue. In fact, if on the one hand a Cloud Broker should always consider user profits neglecting provider ones, on the other hands the Cloud Federation must operate a trade-off between these two apparent discording objectives, for example ensuring fairness in exploiting resources belonging to the federated providers.

Cloud Computing is getting advanced day by day. Cloud computing is the technology of future which provides “IT resources as a service” on demand of the user following the “pay-per-use.” It is basically increasing the utilization of IT infrastructure. In order to provide a better service to the end user, issues such as reduce response time, optimize cost, and load balance over data centers are important factor that need to be studied. Selecting the suitable data center to handle the user request is affecting those factors directly. The Broker policy determines which data center should service the request from each user base; so choosing appropriate policy can improve the performance noticeably. One of the benchmarks policies is service proximity-based that routing the request to the data center, which has lowest network latency or minimum transmission delay from a user base. If there are more than one data centers in a region in close proximity, then one of the data centers is selected at random to service the incoming request.

However, other factors such as cost, workload, number of virtual machines, processing time etc., are not taken into consideration. Randomly selected data center gives undesirable results in terms of response time, data processing time, cost, and other parameters, in this study we discussed the limitations of the current proximity-based and proposed a solution to overcome those limitations.

1.1 Cloud Service Broker Management: Management and governance of cloud services can also be a core feature of a CSB that includes:

- ***Service provisioning:*** The ability to launch a service and allocate the cloud resources needed to support the service, such virtual machines or storage systems.
- ***Service monitoring:*** The ability to monitor cloud services during run time to determine how they are behaving as to policy, and reporting on what occurs to determine the future use of that service.
- ***Service performance:*** The ability to monitor service performance, and log that data as well.
- ***Service governance/policy:*** The ability to define and leverage policies around the use and execution of the service or services. For instance, limiting the amount of money that can be spent on a cloud service, the number of services you can launch during a given time, or security-related limitations.
- ***Service orchestration:*** The ability to orchestrate the cloud service to meet the needs of a core business application or business process.
- ***Service catalog management:*** The ability to manage the cloud services catalog (or service directory), including tracking specific services within the catalog.
- ***Service billing:*** The ability to track the ongoing cost and cost metrics around the use of the cloud services that are managed by the broker. This data may be leveraged for both show-back and charge-back.

In order to support this process, the functional architecture is represented in Figure 1. As shown in Figure 1, the following elements are KEY components as the key function for the SB to provide mechanism.

- o Matchmaking: It finds a proper couple between Service Provider and Service User according to Service Level Classes based on utility of service in the user side, and generates or updates the service matching table between Service Provider and Service User.
- o Analyzing: It analyzes various information resulted from monitoring users, services, and providers, and finally provides categorized and structured types of those information.
- o Scheduling: It allocates requested service to a specific service provider according to the result from matchmaking, and schedules it to be complete in resources of the allocated service provider with satisfying QoS constraints.
- o Monitoring: It monitors all of things in this service environment. For example, it monitors user's preference and user's state information. Also, it monitors service states running on resources of service providers and states of service providers such as reputation, capability and so on.

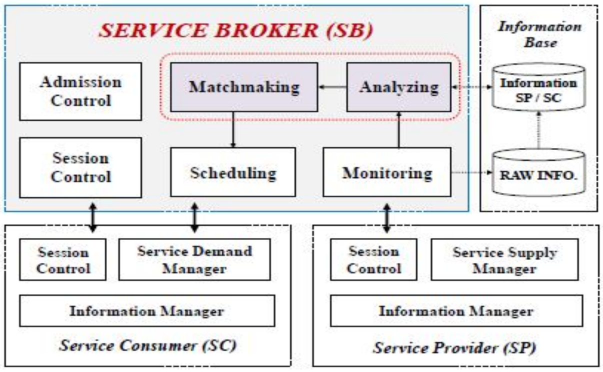

Figure 1: Different Authentication method in Cloud computing.

II. LITERATURE SURVEY

Since the main goal of the service brokers is to direct the user requests to the best DC with optimal performance, the service broker policy has to efficiently select the best data center for the job considering many factors such as time, cost, and availability. Based on existing three different broker algorithms that are proximity-based routing, performance optimized routing and dynamically reconfiguring routing. The Proximity-based routing selects the closest region depending upon the least network latency and from that region it selects the data center randomly. However, this policy has many limitations that affect the response time and may lead to overwhelm a certain data center.

Many researchers aim to overcome these problems. For instance, Instead the random selection of the data center

Kapgate [3] proposed round robin algorithm, this approach improve the resource utilization by selecting DC among all DCs available in single region in round robin manner. However, since the processing speed of DCs may vary, this approach may lead to resource starvation by chosen the fast DCs more often than slow DCs.

Mishra et al [4] in his work similarly used the round robin algorithm instead of random selection but with considering the DC priority, he presented a priority-based round-robin service broker algorithm that distributes requests depending on the DC priority, which enhances the performance comparing to original random selection. Other works focus on improve the cost in the current policy like Limbani et al [5] that present approach that focus on the cost, they modify the proximity-based routing policy to select the low-cost DC it considers VM cost alone) if the region contain more than one DC. This policy is efficient in selecting the lowest cost data center, but it has no consideration for other important factors such as the response time, the workload and the bandwidth.

Chudasama et al [6] in his work similarly presented policy that lower the cost by modifying proximity-based routing policy to select the DC that having less cost if more than one DC located in same region, this approach has good impact on the cost but the response time and load balance still giving poor results, So in order to reduce the response time and the overall load on DC, Kapgate [7] implemented a predictive service broker algorithm based on the weighted moving average forecast model. Sunny et al [8] proposed weight-based algorithm to remove the random selection, the weights assigned to each DC depending on the physical characteristics of the data center. This policy helps to distribute the load appropriately among the DCs, the response time was improved comparing to the proximity based policy, but this improvement was not so sufficient. Sarfaraz et al [9] to avoid overloading certain DC showed proximity-based routing policy that rout the traffics to the neighboring DCs in the same region, but this routing was not considering the physical characteristics of the data centers, which may affect the response time. Vibhavari et al [10] describes policy that eliminates the sequential selection of inter region data center with improvement in overall performance and the data center with less number of users is selected when network latency is same for all data centers. Semwal et al [11], proposed a new policy to select the data center with the highest configuration. The main goal of this policy is to optimize the response time.
From the routing of the user requests it is quite evitable that many of the issues arise while: Selecting the appropriate data center: And this is the responsibility of the broker policy, we have multiple polices that have major effect on the performance. Choosing appropriate data center by applying appropriate broker policy is an important step toward providing better performance. Presenting appropriate broker algorithm is the work of research. Selecting appropriate VM: After selecting the data center it’s important to select appropriate VM, this selection will affect directly the load balance within the data center. Various load-balancing techniques are present and proposed to enhance the cloud performance. The problems may arise from applying some broker policy that may route all the requests to only one data center. As a result, only one data center is highly loaded and others are not. The situation may arise that all the requests may go to only one data center. As a result, only one data center is highly loaded and others are not. This scenario may happened if the used policy was proximity based policy that route the user request to the closet data center, but if there are more than one Data center in the same region, the request directed to a random data center.

III. PROPOSED SYSTEM

It has been concluded that, in cloud environment, there is always been requirement to select data center in an efficient and cost-effective manner. Thus, Service broker policy plays an important role in selecting data centers in cost –effective way which may be beneficial to both cloud providers and cloud users. Efficient service broker policy is required for reducing response time and processing time.

These are the steps used in this research work:

1) Creation of Virtual Cloud using Java based Cloud Analyst (based on Cloud Sim toolkit imported in workspace to run) a simulator for analyzing cloud traffic based on user defined parameters.
2) Configuring the Cloud environment on Cloud Analyst.
3) Creation of Data Centers and User Bases.
4) Implementation of Service Broker Policies –

- Closest Data center Policy.
- Reconfigure Dynamically with Load Policy.
- Proposed Policy (Based on Cuckoo Optimization)

5) Performance Analysis of Service broker Policies based upon response time of data center (in ms).

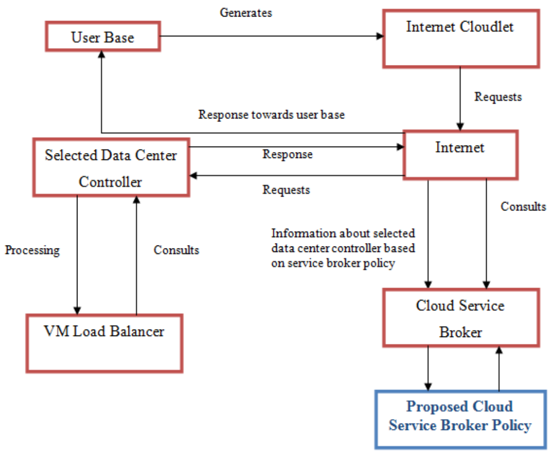

Figure 3.1: Proposed system.

Some important rules that should be considered in using cuckoo search are:

1. Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
2. The best nests with high quality of eggs (solutions) will carry over to the next generations. 3. The number of available host nests is fixed, and a host can discover an alien egg with a probability. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

Algorithm for using cuckoo search in service broker policy is shown below:

Set cloudlets List=null
temp_ List_ of_ Cloudlet=null
Put any incoming Cloudlets in Cloudlet List in order of their Arriving time
While cloudlet List not empty or there are more incoming Cloudlets
Set N=Size of VMs List
If Size of Cloudlet List greater than N then
Transfer the first arrived N Cloudlets from Cloudlet list and put them on temp_ List_ of_ Cloudlet
Else
Transfer all cloudlets from cloudlet list and put them on temp_ List_ of_ Cloudlet
End if
Execute cuckoo search with temp_ List_ of_ Cloudlet
End While
Algorithm for cuckoo search will be as follows:
Generate an initial population of host nests;
While (t<MaxGeneration)
Get a cuckoo randomly (say, i) and replace its solution by performing Levy flights;
Evaluate its quality/fitness Fi
Choose a nest among n (say, j) randomly;
if (Fi > Fj),
Replace j by the new solution;
end if

A fraction (Pa) of the worse nests is abandoned and new ones are built;

Keep the best solutions/nests;
Rank the solutions/nests and find the current best;
Pass the current best solutions to the next generation;
end while

IV. PERFORMANCE EVALUATION

Performance of proposed method will be evaluated on the basis of following parameters:

- Overall response time
- CSP processing time

Performance has been tested for 5 CSP, 25 users & 8 CSP, 25 users. Configuration of simulator remains same in all cases. Table below shows the performance evaluation of **average overall response time (in ms)** for CDC, RDL & proposed method.

Table5.1: Performance according to average response time.

Configuration	Service Broker Methods		
	CDC	RDL	Proposed
CSP-5, CU-25	72.72	78.04	74.24
CSP-8, CU-25	51.08	52.95	50.18

4.1 Results & Evaluation

Results of above evaluations show that proposed algorithm completes user allocation with lower response time and higher performance as compared to existing cloud service brokering algorithms. Performance of proposed algorithm is better than ORT, CDF and RDWL for different number of cloud users with different number of cloud service providers. Results shows that proposed algorithm behaves better in terms of response time after testing it with Cloud Analyst Simulator.

V. CONCLUSION AND FUTURE WORK

Cloud services are omnipresent nowadays. The standardization to describe these services is focused on deployment and management. This poses challenges for service consumers to select the most competent solution.

Current solution is to utilize regular web search engines and individually compare all the details. Manual search and selection is not only time consuming but also makes it impossible to automate user application. This lack of automation - requires source level changes in applications to find superior services or change in service providers. A service broker not only provides ability to automate user application but also facilitates a mechanism to select the most cost effective service providers with greater QoS.

Future work

As cloud computing becomes main stream and the relevance of cloud brokers becomes more important, many players will compete to take the position of cloud broker. Majority of the cloud brokers of tomorrow will be companies which will be born only to cater to this niche requirement. Telco's and data center hosting providers will morph themselves to become cloud brokers. Today's IT managed services providers who are providing managed services for traditional IT Infrastructure will be the best positioned to perform this expanded function. The primary reason being an in-depth understanding of the enterprise IT (Application, Infrastructure) landscape, security and compliance requirements. They also have an established trust with the cloud consumers due to virtue of incumbency. The Managed service providers have also rapidly up skilled their staff to gain knowledge on cloud architecture, Migration and implementation of hybrid cloud and in providing consultancy and professional services. These players have also established strategic relationships with cloud providers such as AWS, Azure, Google, Alibaba etc. and have become their partner of choice. Most importantly they carry a neutral view of the cloud service providers and can advise the cloud consumers truly based on the right fit. Last but not the least the managed services provider have a very strong established talent base and are also better poised for rapid scaling to meet the fast growing cloud market vs the niche or the Telco's, hosting providers.

REFERENCES

[1] Microsoft, Small and midsize businesses cloud trust study: U.S. study results, http://www.microsoft.com/en-us/news/download/presskits/security/docs/twcjune13us.pdf.

[2] R. Buyya, R. Ranjan, R.N. Calheiros, Intercloud: Utility-oriented federation of cloud computing environments for scaling of application services, in: Algorithms and Architectures for Parallel Processing 6081/2010, in: LNCS, vol. 6081, 2010, p. 20.

[3] D. Kapgate, "Improved Round Robin Algorithm for Data Center Selection in Cloud Computing," International Journal of Engineering Sciences & Research Technology, 3(2), 2014, pp. 686- 691.

[4] R.K. Mishra, S. Kumar, B. Sreenu Naik, Priority based Round-Robin service broker algorithm for Cloud-Analyst[C]//Advance Computing Conference (IACC), 2014 IEEE International. IEEE, 2014: 878-881.

[5] D. Limbani, B. Oza, "A Proposed Service Broker Policy for Data Center Selection in Cloud Environment with Implementation,". International Journal of Computer Technology & Applications, 3(3), 2012, pp.1082-1087.

[6] C. Devyaniba, T. Naimisha, "Cost effective selection of data center by proximity-based routing policy for service brokering in cloud environment," International Journal of Computer Technology and Applications, 3(6), 2012, pp. 2057-9.

[7] D. Kapgate, "Weighted Moving Average Forecast Model based Prediction Service Broker Algorithm for Cloud Computing," International Journal of Computer Science and Mobile Computing 3(2), 2014, pp. 71-79.

[8] N. Sunny, A. Mohit, S. Raveena, Weight-Based Data Center Selection Algorithm in Cloud Computing Environment, Artificial Intelligence and Evolutionary Computations in Engineering Systems. Springer, New Delhi, 2016: 515-525.

[9] A. Sarfaraz, "Enhanced proximity-based routing policy for service brokering in cloud computing," International Journal of Engineering Research and Applications, India, 2(2) , 2012, pp.1453-1455.

[10] P. Vibhavari, P. Nisha, "A proposed service broker policy for inter region data center selection in cloud environment," International Journal of Engineering Research and Applications, 3(4), 2013, pp.1699-1702.

[11] A. Semwal, P. Rawat, "Performance evaluation of cloud application with constant data center configuration and variable service broker policy using CloudSim," International Journal of Enhanced Research In Science Technology & Engineering, 3(1), 2014, pp. 1-5.