Improving The Cloud Service Broker Policy For Better Resource Utilization

Satya Narayan Pyashi[1], Nagendra Kumar[2]
[1] Dept of CSE
[2] Assistant Professor Dept of CSE
[1, 2] Shri Ram Institute of Science & Technology, Jabalpur, Madhya Pradesh, India

Abstract- *Cloud brokering facilitates Cloud Service Users (CSUs) to find cloud services according to their requirements. In the current practice, Cloud Service Broker (CSB) selects the cloud services from multi-cloud environment according to Service Level Agreement (SLA) offered by Cloud Service Providers (CSPs). Difficulty in selecting cloud services from multiple CSPs is; there is no guarantee that CSPs deliver cloud services according to SLA offer. In our observation, it is found that most of the CSPs do not fulfill the service commitment mentioned in SLA agreement. It is necessary to be ensured that CSPs are delivering cloud services according to SLA commitment before recommending the cloud services to the CSUs. In this work, we propose a selection method to the CSUs considering both SLA commitment and service delivery by CSPs. As CSUs' requirements are independent to each other and different like minimum Cost for services, high Availability, high Performance, high Security etc., there is not a single optimum solution and there is not a single objective.*

To provide the best cloud services according to CSUs' requirements, we propose Pareto solutions that consider multiple criteria to provide the optimum set of solutions to the CSUs from multiple CSPs.

Keywords- SLA; Multiobjective Optimization; Cloud Brokering; Cloud Service Measurement, Cloud Service Provider, Cuckoo Search Optimization.

I. INTRODUCTION

Two orthogonal approaches are commonly exploited for addressing deployments across multiple clouds: Cloud Brokering and Cloud Federation [1]. Cloud Brokers can leverage abstraction APIs, such as Apache Libcloud1 or Delta Cloud2 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 discarding objectives, for example ensuring fairness in exploiting resources belonging to the federated providers.

Additionally, such approaches can help to overcome the trust problem that limits the adoption of Cloud Computing, for instance by selecting time by time providers that are most suitable to fit the security needs of the users. As an example, the user may want to choose a particular provider location when submitting applications for ensuring law compliance in data management. Recent advances in this research field designed and developed in the Contrail approach to Cloud Federation [2-4], treat security needs by explicitly addressing Quality of Protection (QoP) terms as a special case of Quality of Service (QoS).

One of the most relevant research challenges focuses on the problem of scheduling complex applications by respecting user constraints that have to match the providers' offer. The related aspect to consider is the number of worldwide providers. While it can be considered acceptable to manually search for resources on handful of providers, this task becomes unfeasible when the number of providers grows up to hundreds. To address this issue we conceived, designed and developed a Cloud Brokering approach that provides an optimized deployment solution for a cloud-based application across multiple clouds. CSB exploits only the information that commercial providers are likely to made available for customers, such as Virtual Machine (VM) costs and their features in term of storage, memory, etc.

Let us consider a scenario in which customers submit their applications to brokerage requesting for a deployment configuration that meets QoS requirements that could be formally expressed by Service Level Agreements (SLAs). Such requirements may involve both non-functional aspects, such as security capabilities of providers, and functional aspects as coming from other specification formats, such as the Open Virtualization Format (OVF [5]). For example, application requirements may specify that VMs require at least a certain amount of memory, and a minimum number of physical CPUs, along with the exact match of geographic location where to place specific parts of the application. Such requirements are used as constraints by brokerage for choosing a set of Cloud providers that can host the services (appliances) and at the same time guaranteeing the respect of the QoS negotiated for the whole application.

In order to address a focusing topic in Cloud Service environments, at first, we define proper brokering model to represent users willing to pay for services in a cloud service environment. This brokering model is based on user's context and service context information including information in resource level such as resource status like an available capacity and resource reputation. Using the brokering model, we propose [6] utility adaptive cloud service brokering mechanism with a new brokering algorithm using the CSB function to support a user centric unified environment among multi-devices belonging to a specific user. In addition, we describe the detailed communication procedure based on the defined environment to provide the proposed user centric cloud service efficiently. The utility adaptive service broker matches a user to a specific service provider to maximize utility value of the user. Moreover, it concurrently considers a service provider to get adequate benefit such as cost effectiveness and utilization of resources as providing own services to the user. Finally, it can be appreciable to global Cloud Service environments with reasonable performance throughout simulation results.

1.1 Cloud Service Broker:

Cloud Service Brokers (CSBs) rose in importance due to the diverse number of public and private cloud providers, and the need to manage the consumption of those services within an enterprise. As you can see in Figure 1, CSBs sit between the service consumers, which is typically an application, and the cloud service provider [7].

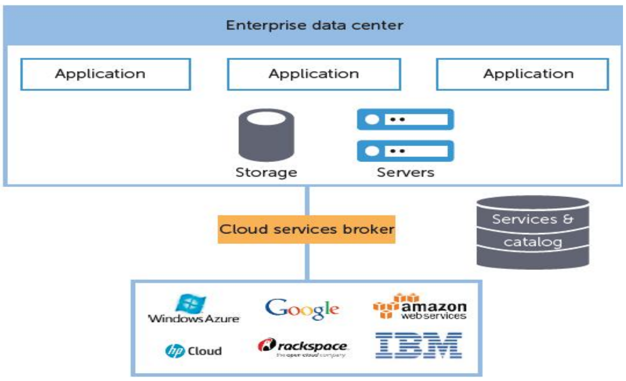

Figure1.1: Cloud Service Brokers (CSBs).

Basically there are two types of CSB: ***Internal CSBs*** are designed to sit within the enterprise firewall to provide core cloud service brokering functions. Internal CSBs are more of a fit for those enterprises that leverage a large number of private cloud services, or services externalized from more traditional systems. These are patterned after more traditional service directories that are leveraged around the use of service oriented architectures, including some public cloud services as well. The advantage of leveraging an internal CSB is the ability to provide unified security, compliance, license management, and support, as well as the ability to have direct control over the service directory. Performance may also be an advantage, as well as the ability to customize the CSB to meet the direct requirements of the enterprise [8].

Others may do the opposite. You need to understand the approach to cloud brokering that each CSB takes before you align with that CSB. In some cases, you'll need more than one CSB. For example, enterprises may leverage both an internal and external CSB [9]. The value of a CSB is based upon the capabilities of that broker. It's always a good idea to look at the core capabilities of a CSB, including: brokering, management, and analytics. Keep in mind that you need to create a business case for this technology, which will define the true value it will bring in terms of dollars saved. It's critical that there be a cost benefit, else the purpose of leveraging a CSB is not really there.

1.2. Brokering: Brokering is the provisioning of cloud services such as storage, compute, application, database, etc., upon request from a consumer. The consumer could be an application developer searching for the best cloud services for their application, an end user searching for an analytical service, or even an action done at runtime to provision the best services at an instance in time. Brokering typically means:

• ***Service discovery***, or the ability to leverage a service directory to locate and provision the correct services. For instance, searching for a bare metal storage service that provides the best service for the lowest cost.

• ***Service intermediation***, meaning the broker will provide the mechanisms that allow the cloud services to work and play well with other services, or applications consuming the services. Certain technical issues may be resolved within the broker, such as data and communications mediation.

• ***Service aggregation***, meaning that we're looking to turn groups of services into whole aggregated services, or services that together provide the right function for service consumer For example, providing a data storage service with an analytical service to create a service that can analyze data.

II. LITERATURE SURVEY

2.1 DIFFERENT ROLES OF BROKERS

Gartner [10] defines "brokerage" as a model of business in computing world. Moreover, the same analysts define a useful distinction among the terms "brokerage" and "broker", which are often alternatively used. However, they actually refer to different meanings. These roles of a broker are explained below.

- Aggregation broker: This broker delivers two or more services to consumers and providers. It does not involve any integration or customization of services.
- Integration broker: This broker makes independent services work together for customers. It can allow process integrations, creating new value through integrated results, one-to-many, many-to-one or many-to-many. This broker allows cloud to cloud integration, such as synchronizing between different applications, or cloud to on-premises integration, like netsuite and quickbooks synchronizing spread sheets.
- Customization broker: This broker can alter or add to the capabilities of a service to improve it. Characteristics include new functionality or new modified service. It uses the original cloud serviced enhanced, one-to-many or many-to-one capabilities to include modifications or combining services, and as a basis for implementation of new services.
- However, Gartner states that it is a vendor-driven market research, instead of a vendor-independent assessor of best practice, and that the views are forcibly shaped by the needs of constituencies that pay for its research: distributors, system integrators, and independent software vendors. Alternately, in [11], the term of the Broker is characterized as:
- A complex business model that offers a high value commitment in the rising cloud space. Basically, this model influences skills and abilities from every one of the three of the conventional business models; of software, consulting, and infrastructure. In addition, the work in [12] presents the term of Broker as "an entity that manages the use, performance and delivery of cloud services, while also negotiates relationships among service providers and customers". This work also separates brokers into another three categories, according to their functionality.

2.2 Cloud Brokering Open Challenges:

The computer science community enthusiastically welcomed the concept of cloud brokering. In the same way that it has created many business opportunities, cloud brokering has contributed new problems and challenges to investigate and solve. Cloud brokering research focuses on the development of brokering and multicloud platforms, and on the optimization of the offer presented by the broker to its customers. From the resource allocation perspective, a CSB can act as an intermediary in the process of workload submission. From this perspective, cloud brokering is the process of matching service requests from multiple users to the offers of multiple clouds. The type and granularity of requests depend on the cloud delivery model (for example, applications for SaaS or virtualized resources for IaaS). This approach can further extend the responsibilities of CSBs, which might need to ensure interoperability between clouds [13].

The first challenge to be reviewed by the research community is to create a framework that could practically exploit a wide range of cloud services. Such model could be based on a sample toolkit (for example, Optimis [14]), middleware (such as mOSAIC [15]), or even an open source cloud broker and facilitate the use of multiple clouds by users. With the support of such solutions, CSBs can focus on their core business—that is, supporting the relationships between CSPs and CSCs. CSB resource management problems are combinatorial problems related to the mapping problem. The price of the resource allocation is the first objective, but quality-of-service (QoS) objectives (such as response time and user satisfaction) are also important. Keeping in mind additional user requirements, such as security, reliability, and privacy, we can conclude that the problem is multiobjective. The CSB problems are typically NP-hard, similar to the mapping or bin-packing problems. As a result, they can't be optimally solved in a reasonable amount of time. In the simplified case of IaaS, where CSPs feature standard infrastructures and theoretically have no limit on used resources from the users' perspective, the CSB's problem consists of selecting a CSP and a virtual machine type for each user task. Such a problem is relatively simple when only a single objective is considered, but realistic scenarios often require more. Valid and good quality solutions can be found by tools such as evolutionary computation, including genetic algorithms, simulated annealing, and particle swarm optimization. During a stochastic process, candidate solutions are modified. The selective pressure of the environment, driven by the objective function, leads to convergence toward the best solution. To perform evolutionary computation, it's necessary to provide a common encoding of a solution. In practice for the mapping problem, a candidate solution is encoded as a vector. Each position of the vector corresponds to tasks. The value of each position determines the selected virtual machine type [16]. An alternative approach to solving NP-hard problems is to use problem-specific heuristics [17]. Another line of research focuses on the brokering market environment. In these works, the different actors in a brokering scenario are modeled as agents [18]. The optimization of the brokering is achieved by negotiations between agents [19] and auctions among vendors to offer the best price [20]. Agent models can be interesting for CSBs, as they inherently include distribution of control and market theory or game theory elements, such as models of rationality and iterative decision making. An important area of research is multiagent organizations, in particular the direction of dynamic and online reorganization, which is necessary in real-life CSB environments [21]. The state-of-the-art research addresses many challenges that aren't yet implemented in industrial and commercial solutions. On the other hand, researchers often neglect particularities of real problems, which can require further specialization and additional efforts at the implementation level.

2.3 Existing mechanism:

Since the main task of the CSB is to direct the user requests to the best DC with minimal 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 [22] 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 [23] 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 [24] 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 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 implemented a predictive service broker algorithm based on the weighted moving average forecast model. Sunny et al 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 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 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, 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.

A. Service Proximity Based policy: In order to explore the limitation of this algorithm we will present how it's work, the following steps show how Service Proximity Based handle the user request:

1) Service Proximity Service Broker maintains an index table of all Data Centers indexed by their region.
2) When the user request is received the Service Proximity Service broker retrieves the sender geographical region and queries for the region proximity list for that region from the Internet Characteristics.
3) The broker then route the sender request to the first earliest/highest region in the proximity list. If more than one data center is located in a region, one is selected randomly.

B. Service Proximity Based Drawbacks: The main problem with service proximity-based routing is the random selection of data center when there are more than one data centers present in a particular region with low latency; the results are different even though configurations are kept same. In addition, there is a high probability that the resources that are present are not utilized to their deliverable capability. Also it is possible that the selected data center will increase the response time or might have higher workload or may be of greater cost as compared to those available in same region. The aim of this study is to remove the random selection of the data center if there is more than one data center in the same region, because this random selection is the major problem that leads to all drawbacks.

III. PROPOSED SYSTEM

3.1 Proposed System

Service broker component of cloud is responsible for deciding which data centre should provide the service to the requests coming from each user base. It controls the traffic among various user base and data centres. The first step, for servicing a request to cloud service provider, is to call cloud service broker, which acts as the intermediary between a cloud user and the cloud service providers. The service broker makes use of any one of the available service broker policies in order to send the request to the most appropriate data centre. Service broker policy is data centre selection policy. The use of an efficient service broker policy is quite necessary to ensure that the later tasks are carried out with efficiency and least response time.

System shows various processing steps performed to apply proposed method in the system. Firstly configuration of CSP will be performed for various resources like processors, virtual machines, memory, storage etc. Than assignment of CSP to data centre will be performed and user requests are configured. Than system uses service broker policy to allocate user requests to CSP and data centre.

Architecture of proposed system will be shown in figure below.

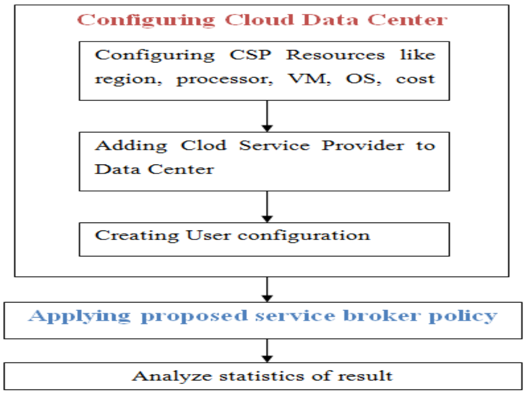

Figure 3.1: Proposed system.

3.2 Proposed Method

Proposed method of the system is to develop a cuckoo search based service broker policy for route user requests to optimal data centre, so that improvement in efficiency and reduction in response time is achieved.

The cuckoo is considered special bird because it has many of the characteristics that distinguish it from other birds. It is characterized by aggressive breeding strategy. Cuckoo lays their eggs in the nest of another species, sometimes the cuckoo's egg in the host nest is discovered may lead to the removal of other eggs or abandons the nest and builds their own brood somewhere else. The algorithms will work with an initial population. The population of cuckoos in the nest egg that they will have a host birds. Some of these eggs that are more similar to bird eggs host more chance to grow and become mature to be cuckoo. Other eggs were identified and destroyed by the home bird. Nest eggs grow the suitability of the area show. The more eggs in one area to be able to live and survive as much profit is more devoted to that area.

Cuckoos follow best place to maximize saving of their eggs. After the chicks hatched came into cuckoo matured, communities and groups, make some region. The best places of residence all groups is next destination in the other groups. All parties to the best of existing migrate. Each

group remains near the present situation. Considering the number of seed that the cuckoo will and are doves of the optimal current for the settlement of some radius egg calculated and shaped. Cuckoos in the nest were then start laying eggs within a radius of themselves. The process to achieve the best placement location continues. The optimal location is where the highest number of cuckoos gathers in it
Each cuckoo randomly laid eggs in the nest of host birds. Then some eggs less similar to hosts bird eggs are detected and thrown out of the nest. So after each egg p% of all eggs (usually 10%) of the amount of earnings function is less, destroyed. The remaining chickens were fed and grow in a host nest. Another interesting point about chicks is that only one egg per nest Cuckoo has possibility to growth. Because when the cuckoo chicks are hatched, are thrown host birds eggs from the nests and if the host bird chicks are hatched earlier than the cuckoo, cuckoo eat the largest amount of food and a few days, host bird chicks die of starvation and only the cuckoo chick survives.

Some important rules, which 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.

IV. IMPLEMENTATION AND EVALUATION

4.1 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.

Table 4.1: 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**

4.2 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

The federated cloud environment is useful to both the cloud service providers and cloud service consumers. The cloud service providers can trade their un-utilized resources through the broker and cloud service users can buy the cloud resources in its cheapest form from federated cloud environment through broker. The broker can also help the cloud service users to find the best service provider. Brokering in federated cloud is a further step to provide the computing facility in a utility-services-like way, similar to electricity, telephony, and water.

In this research, a methodology was developed for a CSB at the SaaS layer of cloud computing. The methodology was then implemented in a case study to evaluate and initially test the broker on two SaaS providers as a proof of concept of the application. The desired results were obtained.

REFERENCE

[1] 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.

[2] E. Carlini, M. Coppola, P. Dazzi, L. Ricci, G. Righetti, Cloud federations in contrail, in: Euro-Par 2011: Parallel Processing Workshops, in: Lecture Notes in Computer Science, vol. 7155, Springer, Berlin, Heidelberg, 2012, pp. 159–168.

[3] G. Anastasi, E. Carlini, M. Coppola, P. Dazzi, A. Lazouski, F. Martinelli, G. Mancini, P. Mori, Usage control in cloud federations, in: 2014 IEEE International Conference on Cloud Engineering (IC2E), 2014, pp. 141–146.

[4] M. Coppola, P. Dazzi, A. Lazouski, F. Martinelli, P. Mori, J. Jensen, I. Johnson, P. Kershaw, The contrail approach to cloud federations, in: Proceedings of the International Symposium on Grids and Clouds (ISGC'12).

[5] Open Virtualization Format Specification, Version 1.1, Specification, DMTF (Jan. 2010).

[6] R. Buyya, R. Ranjan, and R. N. Calheiros, Inter Cloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services, Proceedings of the 10th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2010), LNCS 6081, pp. 13-31, and 2010.

[7] Limbani, D., Oza, B., 2012. A proposed service broker strategy in cloud analyst for cost-effective data center selection. Int. J. Eng. Res. Appl. (India) 2 (1), 793–797.

[8] 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.

[9] Sharma, V., 2014. Efficient data center selection policy for service proximity service broker in Cloud Analyst. Int. J. Innovative Comp. Sci. Eng. (IJICSE) 1 (1), 21–28.

[10] D. Plummer, B. Lheureux, M. Cantara, T. Bova, "Cloud Services Brokerage Is Dominated by Three Primary Roles", Gartner Research Note G00226509, 2011.

[11] NIST, Cloud Computing Reference Architecture, National Institute of Standards and Technology, Special Publication 500-292, September 2011. Available on: http://www.nist.gov/customcf/get_pdf.cfm?pub_id=909505.

[12] N. Grozev and R. Buyya, "Inter-cloud Architectures and Application Brokering: Taxonomy and Survey," Software: Practice and Experience, vol. 44, no. 3, 2014, pp. 369–390.

[13] A. Ferrer et al., "Optimis: A Holistic Approach to Cloud Service Provisioning," Future Generation Computer Systems, vol. 28, no. 1, 2012, pp. 66–77.

[14] D. Petcu et al., "Experiences in Building a mOSAIC of Clouds," J. Cloud Computing, vol. 2, no. 1, 2013; doi: 10.1186/2192-113X-2-12.

[15] Y. Kessaci, N. Melab, and E.-G. Talbi, "A Pareto- Based Genetic Algorithm for Optimized Assignment of VM Requests on a Cloud Brokering Environment," Proc. IEEE Congress Evolutionary Computation (CEC), 2013, pp. 2496–2503.

[16] S. Nesmachnow, S. Iturriaga, and B. Dorronsoro, "Efficient Heuristics for Profit Optimization of Virtual Cloud Brokers," IEEE Computational Intelligence, vol. 10, no. 1, 2015, pp. 33–43.

[17] K.M. Sim, "Agent-Based Cloud Computing," IEEE Trans. Services Computing, vol. 5, no. 4, 2012, pp. 564–577.

[18] K.M. Sim, "Complex and Concurrent Negotiations for Multiple Interrelated E-Markets," IEEE Trans. Cybernetics, vol. 43, no. 1, 2013, pp. 230–245.

[19] A. Prasad and S. Rao, "A Mechanism Design Approach to Resource Procurement in Cloud Computing," IEEE Trans. Computers, vol. 63, no. 1, 2014, pp. 17–30.

[20] M. Guzek, G. Danoy, and P. Bouvry, "ParaMoise: Increasing Capabilities of Parallel Execution and Reorganization in an Organizational Model," Proc. 12th Int'l Conf. Autonomous Agents and Multiagent Systems (AAMAS 13), 2013, pp. 1029–1036.

[21] 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.

[22] 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.

[23] 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.

[24] 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.