RESEARCH ARTICLE OPEN ACCESS

The New Service Brokering Policy for Cloud Computing Based on Optimization Techniques

Swati Raghuwanshi [1], Saurabh Kapoor [2]

[1] (M.Tech. Scholar, Gyan Ganga Institute of Technology and Science, Department of CSE, Jabalpur, M.P., India)
[2](Asst. Prof. Gyan Ganga Institute of Technology and Science, Department of CSE, Jabalpur, M.P., India)

Abstract:

Cloud computing is an area that is rapidly gaining popularity in both academia and industry. Service broker controls the traffic routing between user bases and data centers based on different service broker policies. Service proximity based routing policy selects closest data center to route the user request. If there are more than one data centers within the same region, it is selected randomly without considering the workload, cost, processing time or other parameters. Randomly selected data center is prone to give unsatisfactory results. In this work, we propose modifying that policy by applying new schedule algorithm that can improve the time response, control the load balance. The evaluation is based on the Cloud Analyst, and the results verified the effectiveness of our proposal that can reduce the response time and the average processing time.

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

I. INTRODUCTION

The radical growth of Internet users certainly necessitates excessive use of web and mobile applications that are now majorly hosted by clouds. According to the CISCO, nearly 92% of overall workload will be processed in the cloud by 2020 [1]. Moreover, with the rapid growth in information and communication technologies (ICTs), a large percentage of organizations and enterprises are turning towards the cloud computing paradigm that provides a cost-effective computing for resource-intensive applications. For many businesses, the elasticity, lack of upfront capital, and provision of services according to the degree of requirements (processing, memory and storage resources) are the key advantages of using various cloud platforms [2]. Efficient management of cloud resources help the cloud providers to effectively utilize the available resources that consequently improve the overall performance of the system. For efficient resource utilization, different allocation techniques are adapted by the cloud service providers, such as static and dynamic resource allocation [3]. The selection of the technique is dependent on the choice of service providers and users requirements.

In cloud computing environment, the resources are assigned to users according to their needs on demand. Due to the dynamic, heterogeneous and complex nature of requests in cloud computing environment, efficient resource management is quite challenging [4]. In order to allocate resources efficiently, it is very important to have knowledge of the current state of available resources. Similarly, the process of state information dissemination is equally crucial to design varied resource allocation strategies. Moreover, the number of machines participating in this process does have a significant impact on the performance of cloud services. Consequently, it is indispensable to synchronize state information collection and resource allocation policies. In contrast, a cloud service can face critical problems, such as resource contention, resource fragmentation, under provisioning, and over provisioning etc. The thesis comprehensively surveys the current state of the-art state information collection and numerous brokering decision mechanisms.

1.1 Cloud Computing Reference Model

Figure 1.1 below presents an overview of the NIST cloud computing reference model. This reference model identifies the major entity involved and their functions in cloud computing. The diagram presents a generic high-level architecture and is intended to facilitate the understanding of the requirements, uses, characteristics and standards of cloud computing.

Cloud resource management is a critical task and is related to three types of cloud models; IaaS, PaaS, and SaaS. The main purpose of resource management is to create a balance between user's requirements and availability of resources. As a result, resource management offers many advantages, such as, improvement in quality of service, decrease in cost of services, fault tolerance and scalability etc. [5]. Cloud resource management can be divided into two main categories, state information collection and brokering decision making.

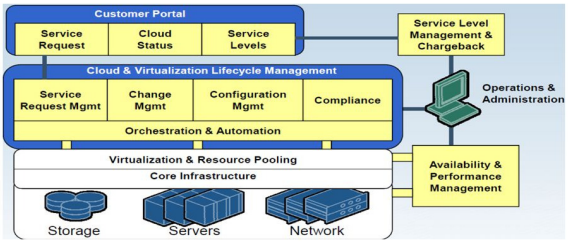

Fig.1 Cloud Reference Model

1.2. Cloud Consumer:

The cloud consumer is the main stakeholder for the cloud computing service. A cloud consumer represents a person or organization that maintains a business relationship with, and uses the service from a cloud provider. A cloud consumer browses the service catalog from a cloud provider, requests the appropriate service, sets up service contracts with the cloud provider, and uses the service. The cloud consumer may be billed for the service provisioned, and needs to arrange payments accordingly. Cloud consumers of PaaS can employ the tools and execution resources provided by cloud providers to develop, test, deploy and manage the applications hosted in a cloud environment. PaaS consumers can be application developers who design and implement application software, application testers who run and test applications in cloud-based environments, application deployers who publish applications into the cloud, and application administrators who configure and monitor application performance on a platform. PaaS consumers can be billed according to, processing, database storage and network resources consumed by the PaaS application, and the duration of the platform usage.

Consumers of IaaS have access to virtual computers, network-accessible storage, network infrastructure components, and other fundamental computing resources on which they can deploy and run arbitrary software. The consumers of IaaS can be system developers, system administrators and IT managers who are interested in creating, installing, managing and monitoring services for IT infrastructure operations. IaaS consumers are provisioned with the capabilities to access these computing resources, and are billed according to the amount or duration of the resources consumed, such as CPU hours used by virtual computers, volume and duration of data stored, network bandwidth consumed, number of IP addresses used for certain intervals.

1.3. Cloud Provider:

A cloud provider is a person, an organization which is responsible for making a service available to interested consumers. A Cloud Provider acquires and manages the computing infrastructure required for providing the services, runs the cloud software that provides the services, and makes arrangement to deliver the cloud services to the Cloud Consumers through network access. For SaaS, the cloud provider deploys, configures, maintains and updates the operation of the software applications on a cloud infrastructure so that the services are provisioned at the expected service levels to cloud consumers. The provider of SaaS assumes most of the responsibilities in managing and controlling the applications and the infrastructure, while the cloud consumers have limited administrative control of the applications.

1.4. Cloud Service Broker:

Broker is an entity that acts as an agent between two entities for negotiating a contract, purchase of sales and gets fee/commission in return. The concept of Cloud Broker was originally defined by the Gartner Research in 2009. The National Institute of Standards and Technology(NIST) [6] defines a Cloud Broker as "an entity that manages the use, performance, and delivery of cloud service and negotiates relationships between Cloud Providers and Cloud Consumers". The broker must support intermediation, aggregation, and arbitrage of services in cloud environment. One of the main benefits of cloud broker is that it enables users to interact through a single interface which connects to multiple service providers. Cloud broker is defined by [7] as a third party entity in federated environment as An entity that may play a role of third party in offering cloud service, adding value of negotiating with many Cloud Service

Providers or customer groups and in some cases managing complex multi-provider services. More specific form of cloud brokering is Cloud Service Brokerage (CSB), is the service partner that negotiates relationship between cloud service customers (CSCs) and cloud service providers (CSPs) and provides interoperability between them. Cloud brokering encompasses a wide range of activities including all intermediaries between CSC and CSP.

An important role in law/regulation compliance management of cloud services can be played by a cloud broker [8] that works as an intermediary in the service procurement process and as a third party controller during the whole service life cycle. The broker should provide services to both customers and cloud service providers, for example: discovery of services compliant with law and Service Level Agreements (SLAs); run-time monitoring of service level metrics; monitoring of legislation changes; law and QoS compliance checking during the service on-boarding phase and, at run-time, during the service evolution phase; aggregation, composition, optimization, orchestration of cloud services.

A cloud consumer may request service from a cloud broker instead of contacting a cloud provider directly. The cloud broker may create a new service by combining multiple services or by enhancing an existing service. Fig.2 below shows the actual positioning of cloud service broker. In this scenario the actual cloud providers are invisible to the cloud consumer and the cloud consumer interacts directly with the cloud broker.

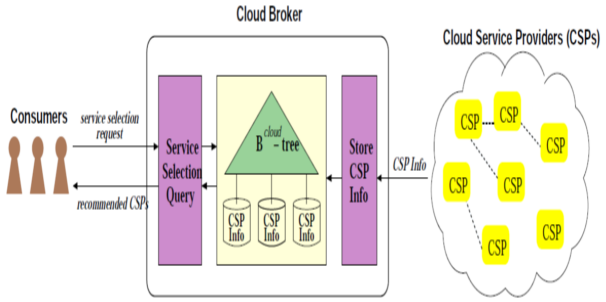

Fig.2: Cloud Brokers usage Scenario.

Cloud broker is a tool which can be used to get centralized access of all the services provided by federated cloud environment. It is used to avoid the difficulties in dealing with various cloud providers along their features. A cloud broker provides a uniform interface to different cloud provider technology, and also collects information from providers (instance availability, prices, etc.). Cloud broker can be used to deploy their virtual infrastructure on multiple clouds. Cloud broker acts as a third party in to offer cloud service along with it adds value of negotiating with many cloud service providers or customers and manage complex multi-provider cloud services. A cloud broker can help the user to choose the right provider based on his requirements. Cloud broker can act a centralized entity from where any user can purchase any computing facility.

II. Related Work

Cloud brokerage has attracted increasing attention from both industry and research communities. In industry, one of the earliest cloud brokers could be CloudSwitch [9], established in 2008 with service for only Amazon EC2. CloudSwitch has the ability to provide federated services on demand and make the cloud a secure and seamless extension of the enterprise data center. RightScale [10] is another cloud broker that offers a cloud management platform to facilitate deployment and management of applications across multiple clouds. More recently, Equinix [11] has also attracted more than 500 cloud providers to register. Many research efforts have also been devoted into cloud brokerage [12] which proposes different types of brokerage framework, service aggregation, resource sharing and allocation, etc. Specifically for the cloud service selection task carried by the cloud brokers, one early effort is by Han et al. [13] who proposed a ranking of available cloud providers based on QoS and Virtual Machine (VM) platform factors. To reduce the number of QoS criteria for evaluation and improve selection accuracy, Qi et al. [14] propose a service selection method based on weighted Principal Component Analysis dedicated to multimedia service selection in the cloud. Since many works consider mainly cost and performance as the selection criteria which may not be sufficient, Rehman et al. [15] defined a general mathematical model to support multi-criteria cloud service selection. To move theory to practice, Jrad et al. [16] developed a cloud broker system that is capable of automatically selecting cloud services based on user defined requirement parameters and the service level agreement attributes of the cloud providers. The limitation is that their algorithm needs to scan all cloud providers and compare the user's requirement with each of the service provider, which could be time consuming when the number of cloud providers is large. Recently, Qu et al. [17] extend the service selection criteria from objective metrics such as price to subjective metrics such as user feedback in order to make the service selection more effective. To further improve the accuracy of subjective metrics, Esposito et al. [18] address uncertainty in the expression of subjective preferences from customers by integrating the fuzzy set theory and game theory into the service selection process. Similarly, Sun et al. [19] also leverage the fuzzy logic to handle uncertainty during the service selection and provide a multi-criteria-based service ranking. Chang et al. [20] added another selection criterion which aims to maximize the data survival probability or the amount of surviving

data. They propose a dynamic programming algorithm based on the famous knapsack problem. Unlike most past works and our work which select a single service type for a cloud user, Wang et al. [21] point out the need to have a combined cloud service and propose an adaptive learning mechanism to help cloud users to combine different service types into an integrated cloud service.

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.

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 [38] 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 [22] 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 [23] 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 [24] 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 [25] implemented a predictive service broker algorithm based on the weighted moving average forecast model. Sunny et al [26] 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 [27] to avoid overloading certain DC showed proximity-based routing policy that rout the traffics to the neighbouring 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 [28] 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 [29], 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. However, this goal was achieved but at the same time increases the overall cost if the data centers process huge data.

III. PROPOSED WORK

3.1 Proposed Model

Generally, Cloud environments are populated with a huge number of heterogeneous data centers that communicate with each other in an ad-hoc manner to

provide the intended services to the end users. On the other hand, the user's satisfaction is measured by the QoS of the provided services. Therefore, the availability of data centers and the reliability of the services are important for better quality of services (QoS). Unfortunately, data centers might be overloaded (i.e. resource shortages) due to inequitable data center selection, load distribution or the increased number of user's and their requests. The influence of the overloaded data centers can be observed through a degraded QoS [30]. As a result, overloaded servers will drop new incoming requests (i.e. buffers are saturated) and new connections are refused (i.e. Queues are full). Since the response time is an estimation of the needed time from the moment a user sends a request to a data center, to the moment the user starts receiving the results, a high response time may indicate that a data center or a cloud resource is overloaded. Therefore, to ensure better performance of the cloud, tasks or jobs should be distributed to the most appropriate DC (service broker) and VM(s) (load balancing) to be executed with minimum response times. Therefore, a minimum response time indicates an efficient execution time (i.e. maximum number of jobs to be performed per unit of time). Hence, the overall performance of the datacenter is also enhanced and not overloaded yet. Given that the network latency is proportional to the response time, it can be in some cases higher than the processing time itself. For instance, small and critical jobs may require very low processing time and the least response time, which can be achieved by selecting the network path with the least network latency.

The following Figure.5.1 shows the basic communication between a Cloud Service requester and Cloud Service Provider through a Cloud Service Broker in Multi Cloud Environment.

Step1: The Requester request for a particular service by sending a request form to the Broker.

Step2: The Broker checks for the service Providers who offers these particular services by contacting multiple providers.

Step3: The Broker prepares a list of Providers who can efficiently provide the requested services.

Step4: The Broker sends this list to the Requester to choose a particular Provider.

Step5: Requester selects a Provider among the list of providers and communicates with that provider.

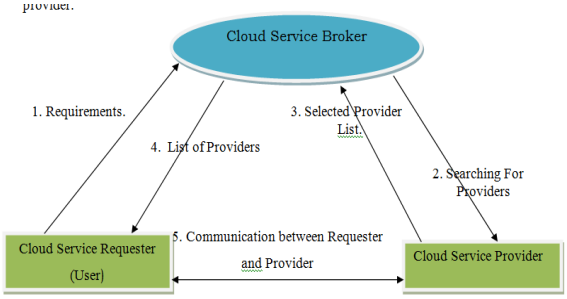

Fig.3 : Communication between a Cloud Service requester and CSP.

3.1: Proposed Algorithm

To overcome the problem found in existing algorithms we propose a new algorithm that will work with global optimal solutions and will always gives the best results by selecting proper values of CSP.

Ant colony optimization (ACO) is a population-based meta-heuristic for combinatorial optimization problems such as the communication network task scheduling problem (CNTSP). Using ACO whose colony scale is P, an individual ant simulates a source node, and its route is constructed by incrementally selecting a destination node until all nodes have been visited. The nodes which have already been visited by an ant or which have violated the capacity constraints are stored in the infeasible node list (tabu). The decision making about combining customers is based on a probabilistic rule taking into account both the visibility and the pheromone information. Thus, to select the next customer j for the kth ant at the ith node, the ant uses the following probabilistic formula.

$$P^{k}_{i,j}(t) = \{ \frac{\tau^{\alpha}_{ij}(t) * \eta^{\beta}_{ij}(t)}{\sum_{j \in allowed\ k} \tau^{\alpha}_{ij}(t) * \eta^{\beta}_{ij.}} \quad \text{..... (1)}$$

allowedk = {1, 2, 3, ..., n} is the list of the cities the k^{th} hasn't been visited yet. Weight α is pheromone factor, which denoted as the influence of pheromone concentration to the path choosing. When it equals to 0, the ant currently selects completely according to greedy rule for path planning. Weight β is the heuristic factor, denoted as the influence of distance of two cities to the path choosing. When it equals to 0, the path choosing depends entirely on the pheromone concentration.

The global pheromone update rule is give by:

$$\tau^{\alpha}_{i,j}(t) = (1-\rho).\tau_{i,j} + P * \tau_{0} \quad \text{.....(2)}$$

Where $\tau_{i,j}(t)$= is the amount of pheromone on the edge (i, j) at time t from one CSP to another CSP; ρ is a parameter governing pheromone decay such that $0 < \rho < 1$; and τ_0 is the initial value of pheromone on all edges between CSP. A higher pheromone values represents a dense route that helps

ants to search the next CSP path fastly and lower ones represents weak value that reduces the searching process. The Pseudo code of proposed algorithm is given below:

Input: Number of CU and List of Resources (VM'S).

Output: Find the best VM in the Datacenter (CSP) in that region.

Assumptions: Let us consider the Cloud User (cu) as an ant for proposed algorithm and Food source as the data centres. The goal is to find the providers with minimum latency.

Step I: For all Users cu in data centre assign Task and unique application id associated with every task. It will specify id for the application it is interested and also include the name of the DC itself as the originator for routing back the responses.

Step II: While k ≤ Number of Ants do

Step III: While CSP[i] list is not empty

Choose CSP which has nearest distance in the region

Assign CSP[i] to cu[k] //the user request.

End Inner while

Record the value of Pheromone of User Base to CSP[i] and store the result in the 2-d array. (Called as Concentration Matrix)

If CSP[i] given service time is not equal to cu[k] taken service time then

Go to **Step III.**

Step IV: Update the 2-d array for all paths using equation (1).

Step V: Go to step I.

Step VI: Repeat for next iteration until loop terminates for all paths. For all other Iterations the 2-d array will be used to find the next CSP [i] which will reduce the network latency for the all other iterations.

The proposed algorithm aims to show the effectiveness of 2-d array to select the data center in order to achieve better resource utilization and remove the disadvantages of existing selection method. Once the matrix is updated for complete cycle, it will be used very easily in the next cycle and hence it will reduce the data transfer time and hence the network latency.

IV. RESULTS

This section presents the simulation experiments to demonstrate the effectiveness of the Cloud service brokering strategies developed in this chapter by using Cloud Analyst tool. The CSB algorithm is implemented in cloud environment which consists of datacenters constituting the host/physical machine. Each physical machine hosts virtual machine with varying processing capabilities. The processing capabilities for virtual machines are computing power in million instructions per second (MIPS), storage in gigabytes (GB) and bandwidth in gigabyte per second (GBps) The metrics used to measure the performance obtained by Cloud service brokering strategies are mean response time of cloud users allocated VM to Cloud Service Providers. This section contains the implementation part of this research that means development and simulation process of the proposed algorithm using Cloud Analyst tool as well as software and hardware requirements for this experiment. The software used for this experiments are Netbeans 8.2 IDE and Cloud Analyst. We have developed the proposed algorithm in Java language. Then Cloud Analyst simulator is used to test the developed algorithm. Then proposed algorithm has been tested by using different number of Cloud users (CU) and Cloud Service Providers (CSP)**.** Then the number of virtual machines is increased by 5 and 10 respectively to test the algorithm on the same Analyst as mentioned above.

4.1 Running Cloud Analyst in Netbeans:

To run the Cloud Analyst simulator, the folder has been added to Netbeans IDE. It can also be run by using Eclipse ide. First of all the existing algorithm has been added to Cloud Analyst. Then we add the proposed algorithm to compare with the existing algorithms. Following are the steps to add the proposed algorithm in Cloud Analyst.

a) Open the folder in Netbeans.

b) We have created our proposed algorithm under CloudSim.ext.serviceBroker.java.

c) The algorithm is given a name called AntColonyOptimization.java.

We have conducted several experiments to test the performance of proposed algorithm in terms of response time and this section also shows performance comparison between the proposed algorithm and three existing brokering algorithms named Optimise Response Time, Closest Data Center and Reconfigure Dynamically with Load Balancing. Number of virtual machines used for simulation is 10 per CSP. Numbers of virtual machines are increased per evaluation to check the performance of proposed algorithm in each environment. Below subsections shows the results of each evaluation.

4.2 Response Time Evaluation for Cloud Users:

There are different sizes virtual machines are used for this evaluation. Table 1 shows the performance comparison between the proposed algorithm, ORT, CDF and RDWL with respect to makespan using different CSP. Figure below

shows the graphical representation of the results where CU = 5 and CSP=3.

Results show that the performance of proposed algorithm is better.

Number of CU and CSP	ORT	CDF	RDWL	PROPOSED ALGORITHM
5/1	310.32	310.52	318.60	310.12
5/3	180.40	180.62	181.69	180.33
10/3	199.59	199.62	200.79	199.52

Table 5.1: Overall Response Time Comparison of Brokering Algorithms.

CASE I: Result Chart for 5 Cloud users and three Cloud Service providers.

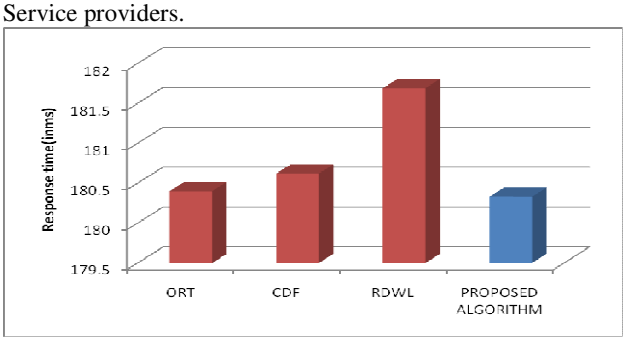

Fig. 4: Graph showing result of Proposed Algorithm.

CASE II: Result Chart for 10 Cloud users and three Cloud Service providers.

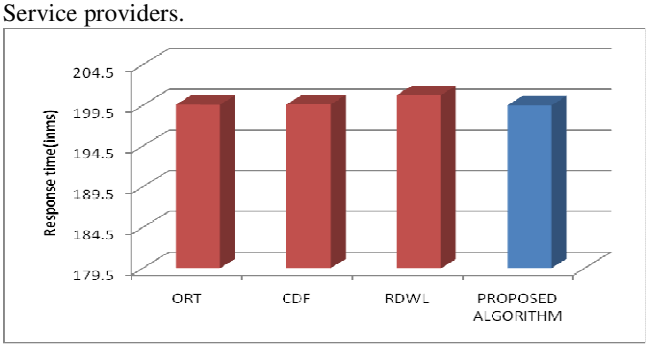

Fig.5: Graph showing result of Proposed Algorithm.

4.3 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

In recent years, cloud computing has become a most important operational model. The assortment of technologies used in cloud computing provides good interaction among a CSR and CSP and also different CSPs through CSB. We have surveyed current approaches in Cloud Service Brokering architectural frameworks, SLAs, Pricing schemes and QOS management. Cloud computing technologies offer major benefits to the IT industries, several benefits of cloud such as High availability, fault tolerance and Infrastructure cost reduction.

The main objective of the job scheduling is to decrease the makespan. In this paper we proposed an algorithm which combines the advantage of ACO. We scheduled the tasks based on the reduction of makespan. Experimental results show that the efficiency of the proposed algorithm is considerably increases. Makespan gets reduced by using a ACO algorithm. In future Hybrid algorithm can be utilized for more and more jobs.

Future work

Cloud Brokering Scheme has created many business opportunities and contributed new challenges to investigate in future. Some of are:

- Better mapping schemes are required to select the provider based on the users requirements.
- Need to enhance security parameters
- Cost optimization has to be improved
- Frameworks have to be developed to provide user friendly environment to the user.
- Proper mechanisms have to be developed to collect requirements from the users.
- Frame works have to be introduced to provide agent to agent communication.
- Proper mechanisms have to be developed to collect feedback from the users.
- Scheduling the request among multiple clouds to balance the load.

REFERENCE

[1]. C. V. Networking, Cisco global cloud index: forecast and methodology, 2015-2020 white papers (2017).

[2]. R. Kamakshi, A. Sakthivel, Dynamic scheduling of resource based on virtual machine migration, WSEAS Transactions on Computers 14 (2015) 224–230.

[3]. B. Rajasekar, S. Manigandan, An efficient resource allocation strategies in cloud computing, International Journal of Innovative

Research in Computer and Communication Engineering 3 (2) (2015) 1239–1244.
[4]. J. Younge, G. Von Laszewski, L. Wang, S. Lopez-Alarcon, W. Carithers, Efficient resource management for cloud computing environments, in: Green Computing Conference, 2010 International, IEEE, 2010, pp. 357–364.
[5]. A. Gulati, G. Shanmuganathan, A. M. Holler, I. Ahmad, Cloud scale resource management: Challenges and techniques. Hot-Cloud 11 (2011) 3–3.
[6]. M. Hogan, F. Liu, A. Sokol, and J. Tong, "Nist cloud computing standards roadmap," NIST Special Publication, vol. 35, 2011.
[7]. M. X. Makkes, C. Ngo, Y. Demchenko, R. Strijkers, R. Meijer, and C. de Laat, Defining Intercloud federation framework for multi-provider cloud services integration. IARIA, 2013.
[8]. E. Casalicchio, M. Palmirani, A cloud service broker with legal-rule compliance checking and quality assurance capabilities, Procedia Comput. Sci. 68 (2015) 136–150. 1st Int'l Conf. on Cloud Forward: From Distributed to Complete Computing.
[9]. Cloud Analyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments, MEDC Project Report-2009.
[10]. RightScale. http://www.rightscale.com/.
[11]. EQUINIX. http://www.equinix.com/industries/cloud-providers/.
[12]. P. Pawluk, B. Simmons, M. Smit, M. Litoiu, and S. Mankovski. Introducing stratos: A cloud broker service. In International Conference on Cloud Computing (CLOUD), pages 891–898, 2012.
[13]. S.-M. Han, M. M. Hassan, C.-W. Yoon, and E.-N. Huh. Efficient service recommendation system for cloud computing market. In Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, pages 839–845, 2009.
[14]. L. Qi, W. Dou, and J. Chen. Weighted principal component analysis based service selection method for multimedia services in cloud. Computing, 98(1-2):195–214, 2016.
[15]. Z. ur Rehman, F. K. Hussain, and O. K. Hussain. Towards multicriteria cloud service selection. In Proceedings of the Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2011, Seoul, Korea, June 30-July 02, 2011, pages 44– 48, 2011.
[16]. F. Jrad, J. Tao, A. Streit, R. Knapper, and C. Flath. A utility-based approach for customised cloud service selection. Int. J. Comput. Sci. Eng., 10(1/2), Jan. 2015.
[17]. L. Qu, Y. Wang, and M. A. Orgun. Cloud service selection based on the aggregation of user feedback and quantitative performance assessment. In 2013 IEEE International Conference on Services Computing, Santa Clara, CA, USA, June 28 - July 3, 2013, pages 152–159, 2013.
[18]. C. Esposito, M. Ficco, F. Palmieri, and A. Castiglione. Smart cloud storage service selection based on fuzzy logic, theory of evidence and game theory. IEEE Transactions on Computers, PP(99):1–1, 2015.
[19]. Cloud-fuser: Fuzzy ontology and fMCDMg based cloud service selection. Future Generation Computer Systems, 57:42–55, 2016.
[20]. C. W. Chang, P. Liu, and J. J. Wu. Probability-based cloud storage providers selection algorithms with maximum availability. In 41st International Conference on Parallel Processing, pages 199–208, 2012.
[21]. X. Wang, J. Cao, and J. Wang. A dynamic cloud service selection strategy using adaptive learning agents. Int. J. High Perform. Comput. Netw., 9(1/2):70–81, 2016.
[22]. Selection in Cloud Computing," International Journal of Engineering Sciences & Research Technology, 3(2), 2014, pp. 686- 691.
[23]. 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.
[24]. 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.
[25]. 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.
[26]. 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.
[27]. 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.
[28]. 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.
[29]. 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.
[30]. 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.