Rough Set Based Quality of Service Design for Service Provisioning in Clouds

Praveen Ganghishetti, Rajeev Wankar, Rafah M. Almuttairi, and C. Raghavendra Rao

Grid Computing Labaratory, Department of Computer and Information Sciences, University of Hyderabad, Hyderabad - 500046, India
{gpraveen86,rajeev.wankar,rafahmohammed}@gmail.com,crrcs@uohyd.ernet.in

Abstract. Quality of Service (QoS) is a broad term used to describe the overall experience a user or application will receive over a network. A rough set based approach is used to design a modified Cloud-QoS Management Strategy (MC-QoSMS). MC-QoSMS is a component of cloud broker that is used to allocate resources based on Service Level Agreement between users and providers for Infrastructure as a Service (IaaS) provisioning of cloud. Concept of reduct from rough set theory is used to allocate the best service provider to the cloud's user with minimum searching time. The performance of the proposed system has been analyzed in terms of number of requests. It is reported that the system outperformed random algorithm by 25% and the round robin algorithm by 30% for 100 requests.

Keywords: Rough Sets, Reduct, Quality of Service (QoS), Cloud Broker, Infrastructure as a Service (IaaS).

1 Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of resources that can be rapidly provisioned and released with minimal management effort or service provider interaction. By dynamically provisioning of resources enables cloud computing infrastructure to meet arbitrary varying resource and service requirements of cloud customer applications. The application requirements can be characterized by quality of service (QoS) requirements such as availability, security, reliability etc., as mentioned in the Service Level Agreement (SLA). In our earlier work, a rough set based Cloud-QoS Management Strategy (C-QoSMS) framework has been proposed to be employed in a cloud broker of cloud environment [1]. In the present work a modified framework MC-QoSMS is employed in the cloud broker to discover the best service provider among all the available service providers in minimum searching time. The term "best" means, the elements of all QoS parameters in the provider match the requirements of the cloud user/ application. We have used the MC-QoSMS framework and analyzed the performance of the frame work in terms of number of requests from the cloud users. The results of the present framework have been compared with the cloud brokers using round robin (RR) and random (RAND) algorithms.

J.T. Yao et al. (Eds): RSKT 2011, LNCS 6954, pp. 268–273, 2011.

2 Background

In order to increase the computational capacity of the local resources by hiring IaaS provider to optimize time and cost two market oriented scheduling policies have been proposed by M.A.Salehi and Rajkkumar Buyya [2]. In the present framework, the searching time is optimized by using a concept called reduct [3,4] of rough set theory [5]. The reduction of attributes is achieved by comparing equivalence relations generated by sets of attributes. Attributes are removed so that the reduced set provides the same predictive capability of the decision feature as the original. A reduct is defined as a subset of minimal cardinality R_{min} of the conditional attribute set C such that $\gamma_R(D) = \gamma_C(D)$ [5] where γ represents dependency of attributes, C and D represents conditional and decision attributes of given information system respectively.

$$R = \{X : X \subseteq C; \gamma_X(D) = \gamma_C(D)\} \tag{1}$$

$$R_{min} = \{X : X \epsilon R; \forall Y \epsilon R; \mid X \mid \leq \mid Y \mid\} \tag{2}$$

3 System Overview

3.1 Architecture of MC-QoSMS Framework

QoS parameters used in MC-QoSMS framework can be classified into three layers namely, Application layer QoS (ALQoS), Middleware Layer QoS (MLQoS) and Netware Layer Qos (NLQoS). ALQoS parameters include Accessibility, Security and Availability; MLQoS parameters include frequency of CPU, secondary memory storage, cluster and GPU while NLQoS parameters include network availability, bandwidth, latency and error rate. In the present model, various parameters namely Availability, Security, Processor frequency, Main Memory Storage, Secondary Memory Storage, GPU, I/O performance and Network Round Trip Time(RTT) of the three QoS layers have been considered while all other parameters mentioned in the Service Level Agreement (SLA) are assumed to be met by the provider. The MC-QoSMS framework has been divided into four components as shown in Fig.1. Cloud Registry(CR) is registry service used to record service capability and QoS provisions for different cloud managers by different service providers. Cloud Manager (CM) is used to host the application of the client by a service provider. *Eucalyptus* [6] environment may be considered by a service provider for service and execution environment with QoS specifications for IaaS. Network Resource Manager (NRM) uses the concept called Bandwidth Broker which is responsible for managing the communication between the user and the cloud manager. Application QoS Manager (AQoSM) has a specific strategy using MC-QoSMS algorithm to find service providers adhering QoS requirements as specified for a service in an SLA. When a service provider has been selected, the AQoSM coordinates with the service providers cloud manager for subsequent service execution.

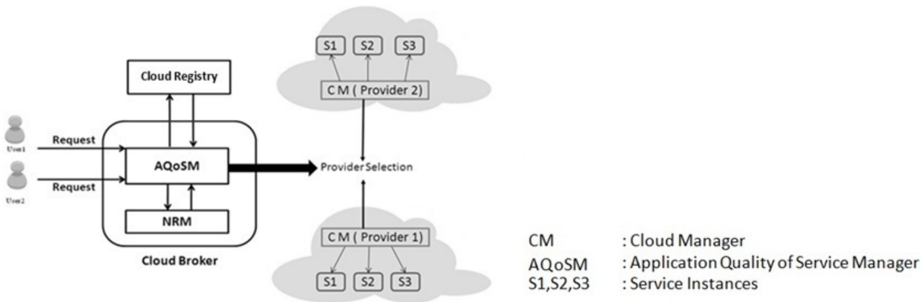

Fig. 1. MC-QoSMS Framework

Cloud providers publish their service along with all types of QoS parameters in the CR then MC-QoSMS algorithm will be invoked. The MC-QoSMS algorithm proposed in the present proposal is given in section 3.2.

3.2 MC-QoSMS Algorithm

A (i)Classification of Applications
B (i)Obtain user choice of Application
(ii)Fetch QoS parameters(PA)(ALQoS and MLQoS parameters)from expect compilation(for instance Amazon classified QoS parameters into six types of applications [1]).
(iii)Fetch the QoS parameters favorable values (PV) of B(ii).
C (i)Fetch all the QoS specifications for each provider from the cloud registry by using cloud brokerage through AQoSM.
(ii)Extract the IaaS providers from the data obtained in C(i) and organize as an Information System (IS).
(iii)Obtain all possible Reducts for above IS.
(a)Group homogeneous instances of the IS and attach an apt cluster label.
(b)Create Reduct Repository (RRP).
D (i)Select closest Reduct from RRP to the QoS parameters(PA) fetched in B.
(ii)Select nearest cluster (restricted to the Reduct selected in D(i)) of C(iii)(a).
(iii)Rank objects of the selected cluster in D(ii) based on RTT(NLQoS param) from least to large(which is achieved by AQoSM interfacing with NRM).
(iv)Return first rank object id(Service provider(SP id)) and user application id to Initiate network allocation.
E (i)AQoSM coordinates with selected provider in D(iv) for subsequent service execution

3.3 Interaction with NRM

NRM is used to monitor the network links between the users and the providers. AQoSM interacts with NRM at two different occasions: (1) AQoSM after applying reducts gives minimized set of providers. Then AQoSM interacts with NRM to monitor the reduced set of provider's network links which closely matches with Network QoS parameters as specified in the SLA. NRM then gives back provider's RTT to AQoSM. (2) Once an appropriate service has been discovered by the strategy used by AQoSM then it interacts with NRM for provisioning of network resources monitoring and supporting admission control. The NRM determines a communicated path between user and the destination. Then the Network path gets reserved using protocols like RSVP as per the Network QoS parameters specified in the SLA for the service.

There is no single universally accepted standardization available in the literature. In our work we have used Amazon's various MLQoS parameters standardized on 10 point scale [7]. The details of Standardization of QoS parameters of various applications selected by the client are given in [1]. One of the applications selected by the client will be processed against the list of providers that the cloud broker has and maps a cloud provider which closely matches the QoS requirements as given by Amazon.

4 Results and Discussion

Cloud Providers with varying performance in main memory, processor, storage, availability, and RTT for a database application [1] (IaaS Service Model) have been studied and analyzed as a function of number of provider requests by measuring Euclidean distance between the ideal and provider's QoS parameters of an application. The present results of the proposed MC-QoSMS framework have been compared with the results obtained using Random algorithm and Round robin algorithm for the same data set. The performance of the proposed framework is found to be closer to the ideal provider and gave best results compared to other providers. The performances of the three algorithms are shown in Fig 2. Fig 2(a) shows the performance of the three systems with 20 requests. The results obtained from the MC-QoSMS framework were found to be ideal provider consistently, and displayed a trend to reach it. The ideal model value is the zero line (the x-axis) which indicates main memory, processor frequency, storage, availability was maximized and round trip time was minimized. The MC-QoSMS selects the best provider at any given time as shown in Fig. 2, where the line represented by the present system is consistently very close to the x-axis. The present system selects best provider out of available ones at any point of time. The chances of selecting the best provider may or may not appear to decrease for subsequent requests. However, the system allows other providers to recover or finish processing of some executing jobs and release more system resources. Thus, it will enhance chances of selecting best provider for the next request.

The ability of MC-QoSMS framework has an added value, in contrast with the other two systems namely, RR and RAND methods. The provider selected by these systems(RR and RAND) is overloaded or is not a best provider, as they have high Euclidean Distance(ED)(shown in Fig. 2) indicating bad performance. Instead if allowed to select best provider by the MC-QoSMS, it completes the task and releases resources thereby increasing system resources. The very low

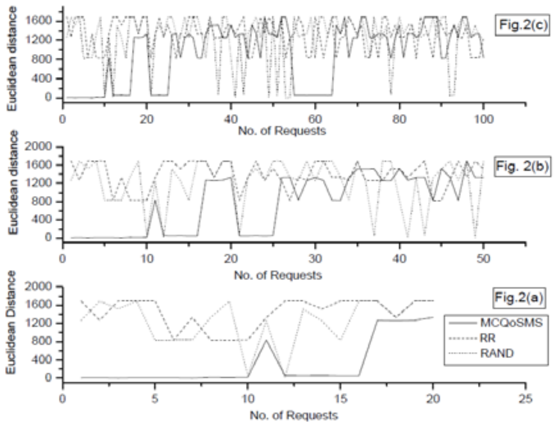

Fig. 2. Performance of MC-QoSMS, Round Robin and Random algorithms (a)for 20 request (b)for 50 requests and (c) for 100 requests

values of ED presented by Random and Round Robin algorithms unfortunately occurred randomly in an inefficient and unbalanced manner. On the other hand, the stable and managed behavior of the present MC-QoSMS framework ensures a higher level of Main memory, Storage, Processor Frequency and availability. Figs. 2(b) and 2(c) show for 50 and 100 requests respectively. These figures show MC-QoSMS framework was still functioning properly, even after increasing the number of provider requests in contrast to the other systems. In the present study it was also noticed that Random algorithms showed better performance than Round Robin method even after increasing the number of requests. This observation is most likely due to its nature of requesting from each provider in a Random manner. However, the MC-QoSMS framework still performed better than both the other systems. Fig.2 also shows that MC-QoSMS framework is more stable than other systems and tends to move closer to the model ideal value (the Zero line). The Average values of Euclidean Distance (AED) for different number of requests obtained in the present frame work are given in Table 1.

Table 1. AED and η(%) for different no. of requests

System/Framework	No. of requests	AED	η (%)
MC-QoSMS	20	318.08	-
	50	766.92	-
	100	928.79	-
Random	20	1151.00	72.36
	50	1223.36	37.31
	100	1248.07	25.58
Round Robin	20	1248.08	77.84
	50	1390.91	44.86
	100	1338.13	30.59

In order to demonstrate further, the Efficiency(η) of the MC-QoSMS framework over other systems is given by eq(3).

$$\eta = \frac{AVOS - AVUS}{AVOS} \times 100 \qquad (3)$$

In eq(3), AVOS represents Average value of the Euclidean distance of the other systems and AVUS stands for Average value of Euclidean distance of the underlying system. Using the AED, the value of η has been calculated for different number of requests and is included in Table 1. It can be seen that the efficiency of MC-QoSMS framework over random is 25.58% and round robin is 30.59% for 100 requests that seems to be closer to the MC-QoSMS framework than the round robin algorithm in the present study.

5 Conclusions

To meet the requirements of both cloud users and service providers, effective and efficient resource broker is proposed with new MC-QoSMS framework. Rough Set Theory is used to minimize number of attributes and minimize searching space by selecting a cluster as required by the user. The proposed framework is more stable than RR and RAND algorithms. The Euclidean distance tends to move to the model ideal value (zero line) indicating enhanced system performance. New QoS parameters namely Main Memory, Processor Frequency, Secondary Storage, Availability, Security and RTT were considered in the present system for a database application in IaaS. The MC-QoSMS framework can be of much benefit to other cloud services that require selection of service provider in cloud environment, utilizing less time and demonstrating high quality of performance.

References

1. Ganghishetti, P., Wankar, R.: Quality of Service Design in Clouds. CSI Communications 35(2), 12–15 (2011)
2. Salehi, M.A., Buyya, R.: Adapting Market-Oriented Scheduling Policies for Cloud Computing. In: Hsu, C.-H., Yang, L.T., Park, J.H., Yeo, S.-S. (eds.) ICA3PP 2010. LNCS, vol. 6081, pp. 351–362. Springer, Heidelberg (2010)
3. Yang, C., Ge, H., Yao, G., Ma, L.: Quick Complete Attribute Reduction Algorithm. In: 6th Inter. Conf. on Fuzzy Systems and Knowledge Discovery, vol. 04. IEEE Computer Society, Washington, USA (2009)
4. Thangavel, K., Shen, Q., Pethalakshmi, A.: Application of Clustering for Feature Selection Based on Rough Set Theory Approach. The International Journal of AIML 6(1) (2006)
5. Pawlak, Z.: Rough Sets. IJCIS 11, 341–356 (1982)
6. Nurmi, D., Wolski, R., Grzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., Zagorodnov, D.: The Eucalyptus Open-Source Cloud-Computing System. In: Proceedings CCGRID 2009, pp. 124–131. IEEE Computer Society, Washington (2009)
7. Amazon EC2 Service, `http://aws.amazon.com/ec2/`