IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-7, Issue-3, March 2019 E-ISSN: 2347-2693

Reinforced Dynamic Clustering (RDC) for Optimal Selection of Cloud Packages

K Mani[1*], R Mohana Krishnan[2]

[1,2]Dept of Comp Science, Nehru Memorial College, Tiruchirappalli, Puthanampatti-621 007, India

Corresponding Author: mk554387@gmail.com, Tel.: +91-9840840455

DOI: https://doi.org/10.26438/ijcse/v7i3.166172 | Available online at: www.ijcseonline.org

Accepted: 11/Mar/2019, Published: 31/Mar/2019

***Abstract*—**This paper presents the Reinforced Dynamic Clustering (RDC) for optimal selection of cloud packages which enable effective package allocation for users. This model operates on four major phases. The initial phase identifies the QoS requirements of customers and clusters them effectively. The second phase identifies the average QoS requirements based on each of the clusters. Decision Tree model is used to train on the data from the clusters and to predict packages that are most suitable for each of the clusters. The next phase handles the real-time resource requirements from the users and allocates packages. The final phase aggregates the user requirements, which are then used in the clustering phase to incorporate the latest user requirements. Experiments were performed with the access log data and comparisons were performed with state-of-the-art models. Results indicate highly effective performances of the proposed model.

***Keywords*—**Resource provisioning, Cloud resource allocation, Clustering, Package Selection, Reinforcement

I. INTRODUCTION

Increasing online based services and increase in processing requirements. Using dedicated systems to handle these issues is not a feasible solution. This has led to the increase in usage of cloud-based services. One major advantage of using cloud is its elasticity. This feature enables users to access resources with higher capabilities even without requesting for dedicated usage [1]. The process of assigning resources to cloud users is called resource provisioning.

Resource provisioning is usually performed by optimal allocation of resources to the users. However, in most cases, the allocation is usually performed based on the average user requirements [2]. Hence there is a huge possibility that the user might require resources with higher QoS requirements at some point during the usage. Elastic nature of clouds come in handy during such situations.

The elastic nature of clouds ensures that the user's operations do not fail due to lack of resources. However, usage of resources with higher QoS levels often results in the users paying higher than dedicated resources [3, 4]. Hence resource provisioning should consider the varying requirements of the users to ensure that the user does not over utilize or underutilize the resources.

Further, another challenge is cloud environments is that cloud resources are always available in packages [5]. Hence users cannot request for fine-tuned requirements. This results in a scenario, where the provided packages always contain higher or lower QoS requirements compared to the user's requirements. The major requirement of a resource provisioning system is to reduce the difference in QoS requirements as low as possible [6]. This work presents an effective reinforcement-based model for package selection in cloud environments.

Rest of the paper is organized as follows, Section I contains the introduction elaborating on resource provisioning and its importance in cloud environments, Section II contains the related work, Section III contains a detailed explanation of the proposed Reinforced Dynamic Clustering (RDC) model, Section IV shows the results and discusses them, and section V concludes research work with future directions.

II. RELATED WORK

In this section, the author describes the previous state of the art and recent research works in the domain of resource provisioning.

A horizontal scaling model to enhance the elasticity of cloud was developed by Kirthica et al. [7]. A cloud system might become overwhelmed with requests such that users' requests might get denied at certain instances. The work proposed by Kirthica et al. [7] aids in avoiding these denial requests by enabling a horizontal scaling mechanism and also by providing an aggregation mechanism that can operate upon heterogeneous clouds. Other similar such models include RESERVOIR [8] and mOSAIC API [9].

A novel resource provisioning model was developed by Kirthica et al. [10]. This work elaborates on a novel residue-based resource provisioning technique to perform horizontal scaling in a cloud. The user's request is split dynamically and is passed to the available resources in multiple clouds to ensure scalability. The InterCloud model proposed by Buyya et al. [11, 12] provides a resource organization method in cloud environments to enable resource delivery to users. A similar model was proposed by Kecskemeti et al. [13, 14, 15].

A framework for resource provisioning was proposed by Calzarossa et al. [16]. This model concentrates on data parallel applications to perform resource provisioning under an uncertain environment with performance variability. This model incorporates an estimation mechanism that estimates the resource requirements prior to the actual execution of the application. This enables the cloud provider to effectively handle the variability associated with the environment. A hard deadline-based model was developed by Bossche et al. [17]. Parallel environments are also used widely for cloud provisioning problems. Some studies related to using parallelized resource provisioning includes Map reduce based models by Alvarez et al. [18], Hwang et al. [19] and Chen et al. [20]. Models proposed by Xu et al. [21] is an enhanced model that also aims to guarantee the QoS requirements of the user when performing provisioning. A dynamic model to perform cloud provisioning was presented by Ralha et al. [22]. This model proposes a multi-layered architecture incorporated with horizontal elasticity capabilities. A cloud model to support automotive applications was proposed by Li et al. [23]. This work performs resource provisioning for automotive vehicles.

III. REINFORCED DYNAMIC CLUSTERING (RDC) FOR OPTIMAL SELECTION OF CLOUD PACKAGES

Optimal selection of packages in a cloud environment is mandatory, as it can avoid both over-utility and under-utility of resources. This work presents an effective method that performs package selection by initially grouping customers in terms of their QoS requirements. The packages available with the service providers are then categorized such that packages are labelled depending on the clusters for which they are best suited. Reinforcement is provided in the architecture by incorporating a cluster updating mechanism that ensures the requirements are always up-to-date. The proposed architecture for Reinforcement Dynamic Clustering (RDC) is performed in three major phases; QoS based clustering, Cluster based QoS determination, real-time request processing and reinforcement-based cluster updating. The proposed architecture is shown in Figure 1.

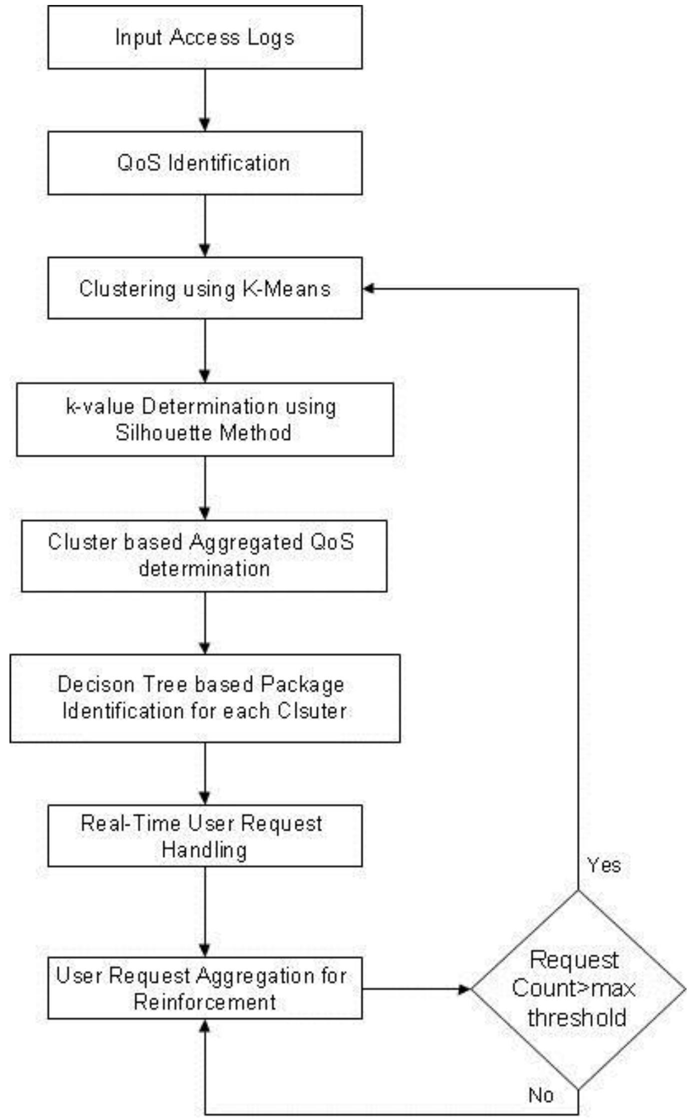

Figure 1. Reinforced Dynamic Clustering (RDC) Architecture

A. *QoS based Clustering of Customers*

This is the initial phase that performs customer grouping based on their QoS requirements. Access log of users is considered as the base data for the grouping process. Data from the user logs is transformed into their corresponding QoS requirements and the base data for clustering is obtained. A sample log file is shown in figure 2.


```
10.223.157.186 - - [15/Jul/2009:14:58:59 -0700] "GET / HTTP/1.1" 403 202
10.223.157.186 - - [15/Jul/2009:14:58:59 -0700] "GET /favicon.ico HTTP/1.1" 404 209
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET / HTTP/1.1" 200 9157
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/js/lowpro.js HTTP/1.1" 200 10469
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/css/reset.css HTTP/1.1" 200 1014
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/css/960.css HTTP/1.1" 200 6206
```

Figure 2. Sample Access Log File

The QoS parameters used for this work includes Bandwidth, Computation capability, Availability, Correctness, Usability, Reliability, Variable Computation load, Serviceability, Latency, Security, Portability, Reliable Storage, Data Backup and Customization. Further, the overall QoS is identified by using the weighted sum method. Every QoS parameter is associated with a weight. The weight signifies its importance pertaining to a particular user. The overall QoS is determined by

$$Overall\ QoS = \sum_{i=1}^{n} w_i QoS_i \qquad (1)$$

Where w_i is the weight of the i^{th} property and QoS_i is the value for the i^{th} QoS property. Significance of the overall QoS is higher than the individual QoS properties. Hence adding it to the data will make the grouping mechanism much more efficient. The final data is composed of 15 dimensions.

The data preparation is followed by the actual grouping/ clustering process. K-Means Clustering algorithm is used to perform clustering. The process of clustering requires the user to provide the number of clusters. The number of clusters create a significant impact on the quality of the clusters obtained. This is performed using the Silhouette method.

Silhouette method is the process of interpreting and validating the consistency of clusters that have been created using a clustering mechanism. It is a value that measures how similar an instance is to its own cluster and how dissimilar it is to all the other clusters. Silhouette values range between -1 and 1. A value of 1 indicates that the instance has been categorized appropriately, while a low value indicates that the instance has been wrongly categorized. The silhouette value for an instance is identified by

$$Silhouette\ Score\ (i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \qquad (2)$$

Where $a(i)$ is the average distance between the instance i and all the instances in the same cluster as i and $b(i)$ is the smallest average distance of i with all the points in other clusters.

The silhouette score provides the optimal number of clusters that can be created using the current dataset. The optimal number of clusters in the current dataset is set to 11 depending on the silhouette score. A sample silhouette chart is shown in figure 3.

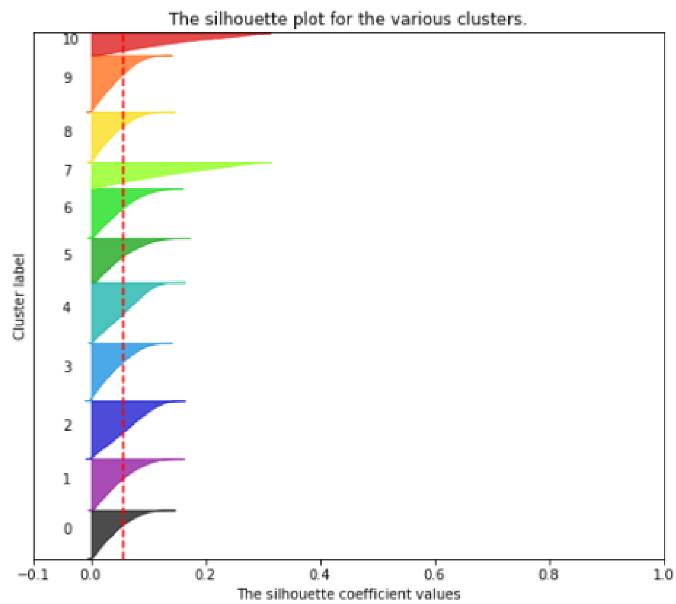

Figure 3. Silhoutte Chart for 11 clusters

It could be observed from figure 3 that all the blocks are almost equal in size with low negative predictions. Hence the k value of 11 was chosen for this work.

B. *Cluster based Aggregated QoS Determination and Package Identification*

The input data is clustered based on the identified number of clusters. Every cluster groups details about multiple users. User group table is created, which contains details about the user and the cluster under which the user belongs to. The structure of user group table is shown in figure 4.

Figure 4. Structure of User Group Table

User_ID	Cluster_ID

Figure 5. Structure of User Group Table

The next step is to identify the aggregated QoS requirements of the clusters. This helps us generalize the user requirements, while still maintaining the individuality of the requirements in a broader level. The average requirements of each QoS parameter is identified.
The next phase is to identify optimal packages that can be assigned to the users contained in the clusters. Decision Tree algorithm is used to perform the package assignment process. Decision Tree model creates tree-like structures from the given input data. Each node is composed of a decision and each leaf node represents the final result. Data from clusters is considered as the input data for the decision tree model. The cluster number indicates the final class column. The problem is considered as a multi-class classification problem. After the training phase, the package details are presented to the trained model and the predictions for each package is identified. This forms the base preparation phase for the proposed architecture.

C. *Real-Time User Request Processing*

When a prediction request arises from the user, their cluster containment is checked and all the packages that are suitable for the given cluster are shortlisted. The package with lowest QoS difference is provided to the user. Since this phase requires selection only from the shortlisted packages, the time requirement reduces to a considerable extent. Hence the waiting time of users is observed to be considerably very low.

D. *Reinforcement based Cluster Updating*

It should however be noted that user requirements are never the same. It tends to vary over time. This property of changes in the requirements of the user over time is referred to as concept drift. Concept drift is a common occurrence in domains that involve decision making based on the user's behaviour.
The major issue in domains experiencing concept drift is that the models developed based on data at a particular time period becomes obsolete in due course of time. Hence every user request is recorded and after a defined period or after the aggregated records reach a particular level, the newly generated records are used to perform clustering and for creating the machine learning model.
This periodic updating of clusters not only aids in handling concept drift but can also be useful in incorporating new users into the model without any need for additional processing.

IV. RESULTS AND DISCUSSION

The proposed RDC model has been implemented using Python, and comparisons were performed with the package selection model proposed by Madhumathi et al. [24]. The proposed model also uses the access log data set used in this model [24].

The processing time required for the proposed model is shown in figure 5. It could be observed that the proposed RDC model exhibits a maximum computational time of ~0.32 ms. This shows the fast processing nature of the proposed model.

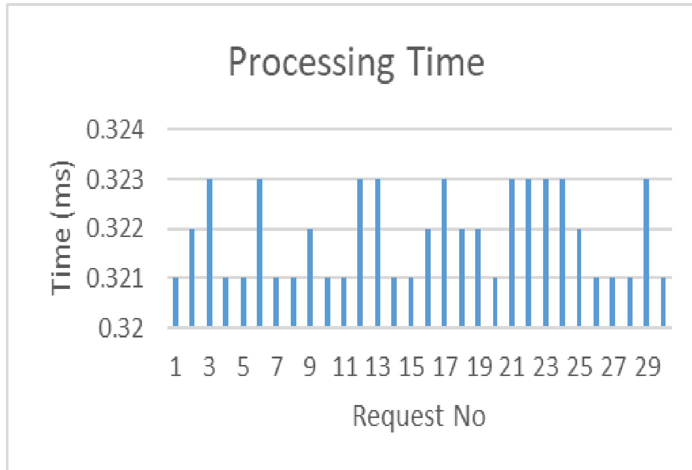

Figure 6. Time Taken for Processing Requests

The difference in QoS values obtained from the proposed model is shown in figure 6. It could be observed that the difference between the required QoS and the proposed QoS is very low, with most of the requests exhibiting QoS difference of less than 10. This exhibits the high precision exhibited by the proposed RDC model.

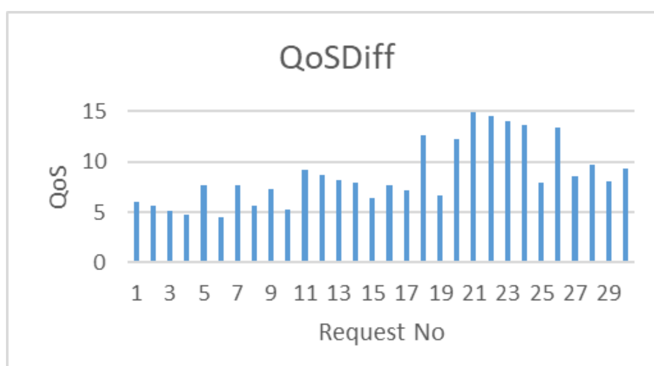

Figure 7. QoS Difference Identified in Requests

A graph depicting the required and the proposed QoS values is shown in figure 7. The solid line represents the requested

QoS and the dotted line represents the provided QoS. It could be observed from the graph that the provided QoS is always observed to be higher than the requested QoS. Further, the difference in the QoS values are also maintained at a minimum level so as to provide the most optimal predictions.

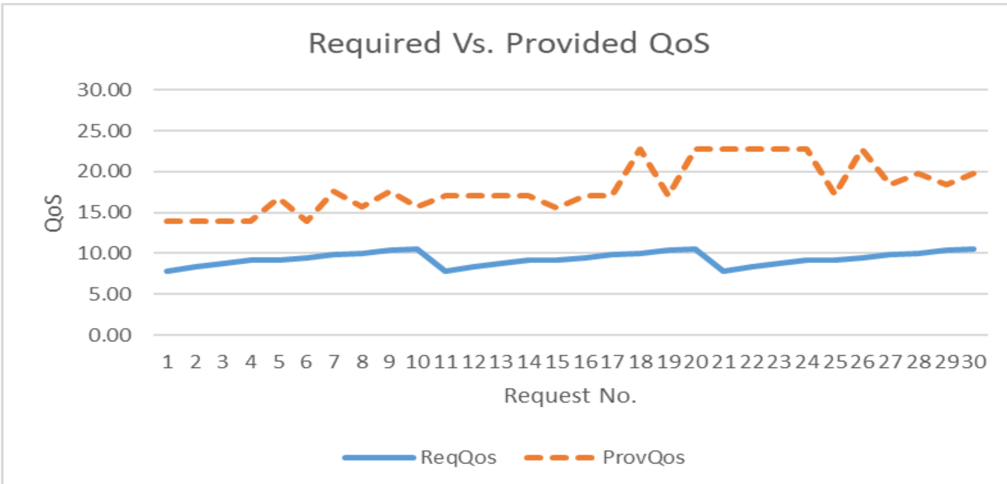

Figure 8. Comparison of Required QoS and Provided QoS

A comparison of time requirements of the ACO based allocation model is shown in figure 8. It could be observed that the proposed RDC model exhibits a time requirement of 0.32 ms, while the ACO based model requires a processing time of 85.6 ms. This shows the reduced computational complexity of the proposed RDC model.

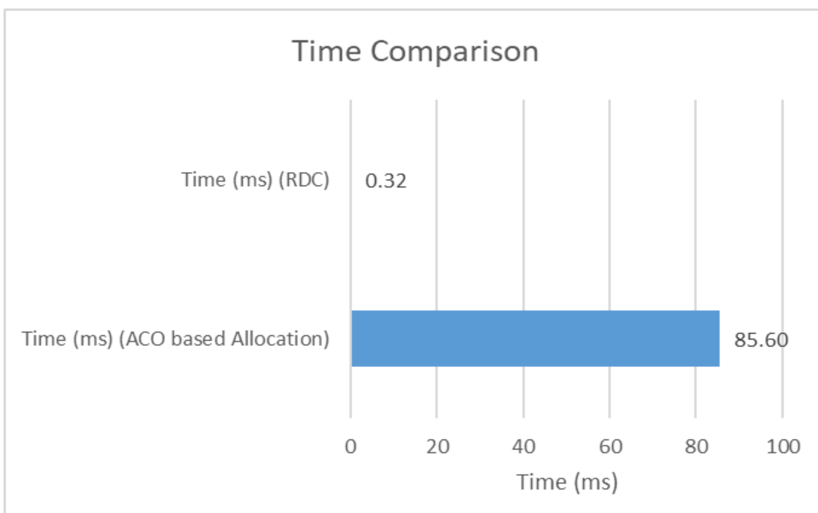

Figure 9. Time Comparison with ACO based Model

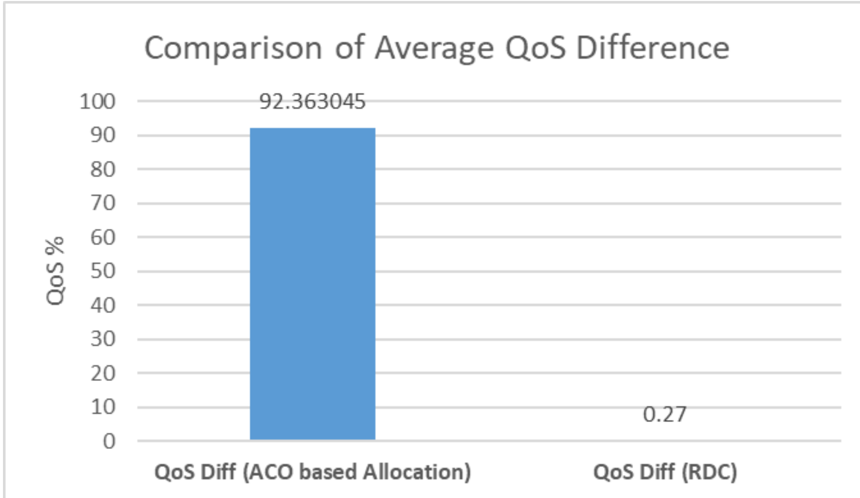

Figure 10. Comparison of Difference in Average QoS

A comparison of the average difference in QoS levels is shown in figure 9. The percent of QoS difference levels was found to be 92% on using ACO based model, while the difference in QoS level has been reduced to 0.27% in the proposed RDC model. This shows that the proposed model is not only time effective, but also provides effective performances.

V. CONCLUSION AND FUTURE SCOPE

Effective package selection for resource provisioning is one of the major requirements in cloud environments. Optimality in the selection process is the most sought-after scenario in such applications. This work proposes the RDC model that can provide optimal resource allocations in cloud. The model uses a clustering mechanism and a package grouping mechanism to initially identify the package grouping. Hence when a user request arises, the package selection process becomes faster. This helps in effectively reducing the waiting time for the end user. The major advantage of this model is the reinforcement mechanism, which collects the user requirements and uses the new requirements for the clustering process. This enables even addition of new users and also enables the model to handle concept drift. Experimental results and comparisons indicate effective performances of the proposed RDC model.

REFERENCES

[1] Mustafa, S., Nazir, B., Hayat, A., & Madani, S. A., "*Resource management in cloud computing: Taxonomy, prospects, and challenges*". Computers & Electrical Engineering, Vol. 47, pp. 186-203, 2015.

[2] Kirthica, S., & Sridhar, R., "*Securely Communicating with an Optimal Cloud for Intelligently Enhancing a Cloud's Elasticity*". International Journal of Intelligent Information Technologies (IJIIT), Vol. 14(2), pp.43-58, 2018.

[3] Kirthica, S., & Sridhar, R., "*CIT: A cloud inter-operation toolkit to enhance elasticity and tolerate shut down of external clouds*". Journal of Network and Computer Applications, Vol. 85, pp. 32-46, 2017.

[4] Grozev, N., & Buyya, R., "*Inter‐ Cloud architectures and application brokering: taxonomy and survey*". Software: Practice and Experience, Vol. 44(3), pp. 369-390, 2014.

[5] Xiao, Z., Song, W., & Chen, Q., "*Dynamic resource allocation using virtual machines for cloud computing environment*". IEEE transactions on parallel and distributed systems, Vol. 24(6), pp. 1107-1117, 2013.

[6] Kumar, M. R. V., & Raghunathan, S., "*Heterogeneity and thermal aware adaptive heuristics for energy efficient consolidation of virtual machines in infrastructure clouds*". Journal of Computer and System Sciences, Vol. 82(2), pp.191-212, 2016.

[7] Kirthica, S., & Sridhar, R., "*Horizontal scaling and aggregation across heterogeneous clouds for resource provisioning*". Computers & Electrical Engineering, Vol. 69, pp. 301-316, 2018.

[8] Rochwerger, B., Breitgand, D., Levy, E., Galis, A., Nagin, K., Llorente, I. M.,& Ben-Yehuda, M., "*The reservoir model and architecture for open federated cloud computing*". IBM Journal of Research and Development, Vol. 53(4), pp. 4-1, 2009.

[9] Petcu, D., Di Martino, B., Venticinque, S., Rak, M., Máhr, T., Lopez, G. E. & Stankovski, V., "*Experiences in building a mOSAIC of clouds*". Journal of Cloud Computing: Advances, Systems and Applications, Vol. 2(1), 2013.

[10] Kirthica, S., & Sridhar, R., "*A residue-based approach for resource provisioning by horizontal scaling across heterogeneous clouds*". International Journal of Approximate Reasoning, Vol. 101, pp. 88-106, 2018.

[11] 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, Springer, pp.13–31, 2010.

[12] R.N. Calheiros, A.N. Toosi, C. Vecchiola, R. Buyya, "*A coordinator for scaling elastic applications across multiple clouds*", Future Gener. Comput. Syst. Vol. 28(8) pp. 1350–1362, 2012.

[13] A.C. Marosi, G. Kecskemeti, A. Kertesz, P. Kacsuk, "*FCM: an architecture for integrating IaaS cloud systems*", in: 2nd

International Conference on Cloud Computing, GRIDs, and Virtualization, IARIA, pp.7–12, 2011.

[14] G. Kecskemeti, M. Maurer, I. Brandic, A. Kertesz, Z. Nemeth, S. Dustdar, "*Facilitating self-adaptable inter-cloud management"*, in: 20th Euromicro International Conference on Parallel, Distributed and Network-Based Processing, IEEE, pp.575–582, 2012.

[15] A. Kertesz, G. Kecskemeti, M. Oriol, P. Kotcauer, S. Acs, M. Rodríguez, O. Mercè, A.C. Marosi, J. Marco, X. Franch, "*Enhancing federated cloud management with an integrated service monitoring approach*", J. Grid Comput. Vol. 11(4) pp. 699–720, 2013.

[16] Calzarossa, M. C., Della Vedova, M. L., & Tessera, D., "*A methodological framework for cloud resource provisioning and scheduling of data parallel applications under uncertainty*". Future Generation Computer Systems, Vol. 93, pp.212-223, 2019.

[17] R. Van den Bossche, K. Vanmechelen, J. Broeckhove, "*Cost-optimal scheduling in hybrid IaaS clouds for deadline constrained workloads*", in: Proc. 2010 IEEE Int. Conf. on Cloud Computing (CLOUD), IEEE, pp. 228–235, 2010.

[18] A. Ruiz-Alvarez, I.K. Kim, M. Humphrey, "*Toward optimal resource provisioning for cloud MapReduce and hybrid cloud applications*", in: Proc. of the IEEE 8th Int. Conf. on Cloud Computing, pp. 669–677, 2015.

[19] Hwang, E., & Kim, K. H., "*Minimizing cost of virtual machines for deadline-constrained mapreduce applications in the cloud".* In Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing(pp. 130-138). IEEE Computer Society, 2012.

[20] K. Chen, J. Powers, S. Guo, F. Tian, "*CRESP: Towards optimal resource provisioning for MapReduce computing in public clouds*", IEEE Trans. Parallel Distrib. Syst. Vol. 25 (6) pp. 1403–1412, 2014.

[21] Xu, X., Tang, M., & Tian, Y. C., "*QoS-guaranteed resource provisioning for cloud-based MapReduce in dynamical environments*". Future Generation Computer Systems, Vol. 78, pp.18-30, 2018.

[22] Ralha, C. G., Mendes, A. H., Laranjeira, L. A., Araújo, A. P., & Melo, A. C., "*Multiagent system for dynamic resource provisioning in cloud computing platforms*". Future Generation Computer Systems, Vol. 94, pp.80-96, 2019.

[23] Li, Z., Chu, T., Kolmanovsky, I. V., Yin, X., & Yin, X., "*Cloud resource allocation for cloud-based automotive applications*". Mechatronics, Vol. 50, pp.356-365, 2018.

[24] C., Madhumathi, and Gopinath Ganapathy. "*Cloud Package Selection for Academic Requirements using Multi Criteria Decision Making based Modified Ant Colony Optimization Technique*." International Journal of Engineering and Technology (IJET) Vol. 8 pp. 1205-1211, 2016.

Authors Profile

Mani. K received his MCA and M.Tech from the Bharathidasan University, Trichy, India in Computer Applications and Advanced Information Technology respectively. Since 1989, he has been with the Department of Computer Science at the Nehru Memorial College, affiliated to Bharathidasan University where he is currently working as an Associate Professor. He completed his PhD in Cryptography with primary emphasis on evolution of framework for enhancing the security and optimizing the run time in cryptographic algorithms. He published and presented round 15 research papers at international journals and conferences

R.Mohana Krishnan received his MCA from the Bharathidasan University, Trichy India in Computer Applications and Advanced Information Technology respectively and MBA from Great Lakes Institute of Management with Association with Stuart School of Business, illinois institute of technology, has 22 years of IT & Strategic Management experience at Various Corporates in India and abroad IBM, CSC, GEMINI, U.S.A and HCR,USA in the field of IT Delivery and Complex Solutions Provider, Client Engagement, Pre Sales and Technology solutions for various domains. Expertise in delivering enterprise level solutions with prime focus in creating maximum value through strategic planning, Leadership in Solutions. Currently associated with HCL India as AGM, and leading the Application and Solutions.

