Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

Location and Temporal based Multi-Cloud Package Selection for Enterprise

K.S. Guruprakash[1], M. Jaiganesh[2], V. Vijey nathan[3], R. Sathya[4]

[1] Assistant Professor, Department of Computer Science and Engineerimg, K.Ramakrishnan College of Engineering, Trichy.

[2,3,4] Assistant Professor, Department of Computer Science and Engineerimg, SRM TRP Engineering College, Trichy.

ABSTRACT

Increase in requirements for resources has led to a huge increase in organizations opting for cloud-based services rather than dedicated hardware. This has in-turn led to a huge increase in the number of cloud providers. Multi-cloud options appear to be luring aspects compared to single cloud resource usage due to the fine-tuned resource requirements that can be satisfied by opting for multiple cloud providers. This paper proposes a metaheuristic-based package selection model for enterprise level organizations to enable multi-cloud package selection. The proposed architecture is divided into two broad sections; the temporal and geolocation-based grouping mechanism and the provider and package selection mechanisms. Both provider and package selection are performed using Firefly based optimization models. Experimental results and comparisons indicate high efficiency in both grouping and package identification process when compared to state-of-the-art models..

Keywords:
Multi-cloud package selection; Location based grouping; Temporal grouping; Firefly Optimization; Enterprise resource provisioning

1. Introduction

Performance is a critical component of cloud environments, as that is the major reason for users to migrate from on-premise dedicated environments to cloud. However, ensuring performance is a major challenge in distributed environments, especially in cloud environments, where multiple users require resources in multiple dynamic and unprecedented time [1, 2]. Utilizing multi-clouds, rather than single cloud environments can ensure performance to maximum extent by providing a fault tolerant environment to the enterprises.

Multi-cloud is the process of utilizing multiple independent cloud providers to obtain varied and specialized resources from each of the providers in accordance with the user requirements. The major issues in single cloud environments is that they are failure prone and the optimality of selections is constrained [3]. The entire organization is completely reliant on a single cloud provider, hence when the provider experiences down-time, the entire organization is forced to experience down-time. Further, a single provider may not be able to satisfy all the QoS requirements accurately. Hence the organizations are usually forced to use whatever packages are available with the provider, either at the expense of quality or at the expense of cost. All these issues can be effectively countered while using cloud environments [4]. The major advantage of moving towards multi-cloud is that, it eliminates the reliance on a single provider. Hence even in case a provider experiences down-time, only the specific functionalities are down, while other modules can be utilized. It tends to provide increased flexibility towards choice and can effectively enable mitigation against disasters.

The remainder of this paper is structured as follows; section II provides the related works, section III presents a detailed view of the package selection models, section IV presents the results and discussions and section V concludes the work.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

2. Related works

Effective resource selection models for cloud environments have been researched since cloud computing has gained prominence. Utilization of multi-cloud resources has been recently adopted due to the increase in usage levels of cloud computing resources. This section discusses some of the recent and most prominent works in this domain[24].

A scheduling algorithm for multi-cloud environment was presented by Panda et al. in [5]. This work presents three task partitioning scheduling algorithms for cloud task partitioning. The model is developed as an online scheduling algorithm for usage in multiple cloud environments. Other similar models include scheduling algorithms by Panda et al. in [6,7]. A cloud brokerage model aimed at solving resource management issues in cloud environments was proposed by Heilig et al. in [8]. This model has its [23] major concentration on reducing the monetary cost and execution time of the consumer, hence providing a cost effective Infrastructure as a service to the user. This model uses a biased random key genetic algorithm to automate the resource management issues in the multi-cloud environments. An open service model for multi-cloud platform was proposed by Paraiso et al. in [9]. This work also proposes a generic kernel infrastructure to enable effective deployment of software-as-a-service applications in multi-cloud. A heterogeneous configuration enabled framework for multi-cloud environments was presented by Quinton et al. in [10]. The major downside of this approach is that it only considers the feasibility and not the optimization of infrastructure-application mappings. Other similar models include works by Heilig et al. [11] and Coutinho et al. [12]. [22]A multi-resource task scheduling algorithm for green clouds was proposed by Mao et al. in [13]. This model incorporates two major algorithm models; one to incorporate time-awareness into the allocation model and the other to incorporate energy awareness into the computing model. Other similar models include challenges in green computing by Wu et al. [14] and power management models by Harchol et al. [15] and Liu et al. [16]. An optimal virtual machine allocation model for multi-cloud environments was proposed by Diaz et al. in [17]. This work presents a Load Level based OptmizatiOn for VIrtual machine Allocation (LLOOVIA). This is an optimization model, proposed with two price schemas reserved and on-demand for cost effective allocations. Other similar models include optimal VM allocation models by Chaisiri et al. [18, 19] and evolutionary VM placement by Mark et al. [20][26].

3. Location and Temporal based Multi-Cloud Package Selection for Enterprise

Package selection in cloud environments is usually based on QoS requirements of the user. In general, package selection models select packages from cloud providers based on the package QoS from a single cloud provider. This suits for an organization with a single base. However, when considering enterprises, workforce is usually distributed in various geographical locations called branches. Every branch in the enterprise will have their own independent requirements and have high probability of being located at different regions. Hence obtaining resources from the same cloud provider is not an optimal solution for this scenario. This claim has been made due to two major reasons; a single provider may not be optimally located when considered in terms of all the branches, and a single provider might not be able to entirely satisfy the requirements of all the existing branches in the enterprise. [25] Even if both the criterions were satisfied, the provider might become a single point of failure leading to fatal issues. However, identifying best suited cloud provider from a list of providers and identifying optimal packages from these multiple

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

providers is a tedious task. This work proposes a location and temporal based package selection model for multi-cloud environments. Architecture of the proposed model is shown in figure 1.

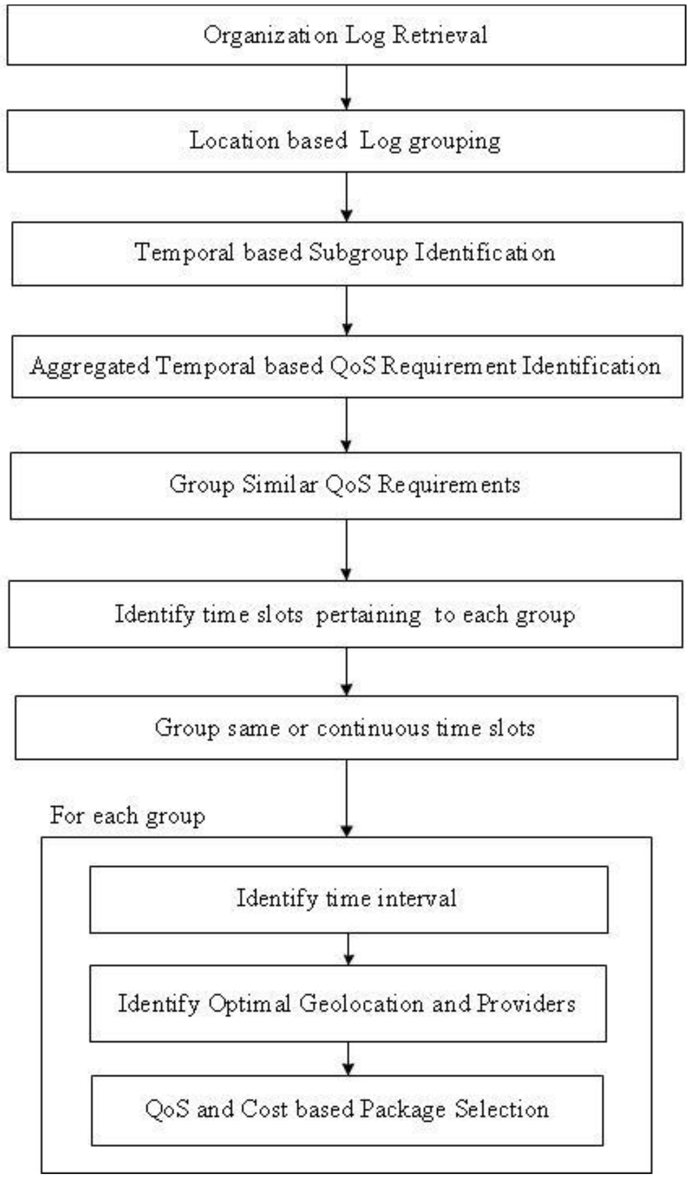

Figure 1: Location and Temporal based Multi-Cloud Package Selection Architecture

Location based Log Grouping
Cloud requirements are usually provided in terms of access logs. Considering an enterprise, each branch in the enterprise will have its own access log details. All these logs are integrated to provide a single unified log for evaluation. The basic format of an access log is given in figure 2.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

```
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
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/css/the-associates.css HTTP/
1.1" 200 15779
10.223.157.186 - - [15/Jul/2009:15:50:35 -0700] "GET /assets/js/the-associates.js HTTP/
1.1" 200 4492
```

Figure 2: Access log Format

Access logs usually contain details about the IP addresses of the client, ID of the client or system name, User ID, timestamp, zone, request, path, protocol used for transfer, status received as acknowledgement and the size of the data being sent.

The initial phase is to perform location based grouping of logs. Locations are usually identified by analyzing the IP address pertaining to the transmission. QoS pertaining to each IP group is identified and an average QoS pertaining to each IP in group is formulated.

Temporal Sub-Group Identification and Aggregation

The next phase is to identify the temporal sub-grouping of the location based grouped data. For each identified location group, temporal sub-groupings are performed. Every group is analysed and temporally segregated using the timestamp available in the access log. Equal and continually occurring time slots are grouped into separate sections.QoS pertaining to each of the time groups is identified, and continuous time-slots with similar QoS requirements are aggregated to form a single requirement unit.

Group based Provider and Package Selection

The initial phases prepares the data in order to easily identify the requirements. This phase performs the process of actually identifying the optimal providers and optimal packages for the requirements. This phase has two major tasks; identifying the optimal providers from a multi-cloud environment and identifying optimal packages from the selected providers.

Geolocation based Provider Selection

This phase identifies optimal providers for the requirements by performing geolocation based selections. The location grouped data obtained in the initial processing phase is utilized in this model. The major issue in moving towards multi-cloud environments is that multi-clouds tend to increase the available package options to a very large extent. This phase aims to reduce the providers and packages to provide a computable list for the package selection process.

The requirement for this phase is to identify the providers at optimal locations who offers packages that satisfy the requirements of the users at the required or lower cost. This work uses Firefly based optimal provider selection mechanism. Geolocation of the providers and the enterprise branches is used as the input to the Firefly based optimization model, and is used to build the search-space. Geolocation pertaining to each of the groups is used to identify the optimally located providers for each of the location groups. Location of the providers is the only criterion used to determine the fitness of the solutions. Multiple providers are identified for each location group. The major reason is that, a single provider might not effectively satisfy the requirements. Multiple providers might provide the user with better options at better cost.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

QoS based Package Selection

This phase identifies the effective packages for each of the location based temporal requirements identified in the previous phases. Package selection is performed using Firefly algorithm [21]. Package details and their overall QoS levels is used to build the search-space. This section formulates Firefly algorithm as a multi-criterion decision making problem, by incorporating package requirements and the cost. Although multiple providers are considered for analysis, all the package details are added to the same search-space for analysis. The process of Firefly based optimization is performed in three major phases; Firefly initialization phase, Firefly movement and Firefly convergence.

Firefly Initialization

The search space is populated with package details pertaining to all the selected providers, along with the QoS requirement details from the organization.Fitness is determined by identifying the QoS pertaining to each of the package. The fireflies are initially distributed in-random in the search-space. The intensity levels of the Fireflies are identified using their QoS values and their corresponding weights pertaining to each of the packages. The requirement node is considered to be test, and the corresponding intensity values of each of the fireflies corresponding to the requirement data is defined by

$$Intensity_i = 1 \Big/ \sqrt{\sum_{j=1}^{attr} (X_{test,j} - X_{i,j})^2}$$

Where Xtest,j refers to the jth attribute of the requirement data and Xi,jrefers to the jthattribute of the firefly i. This process marks the beginning of the firefly based classifier model.

Modified Local Search and Firefly Movement

Optimal solution identification is performed iteratively, and begins from the initial firefly distribution and continues till the convergence of the model on the optimal solution. It is the process, where every firefly is compared with every other firefly to identify the intensity levels. Fitness of the solution determines the intensity levels. Higher fitness corresponds to higher intensity. The firefly with lower intensity is moved towards the firefly with higher intensity.

Firefly Convergence

Convergence is achieved when all the Fireflies or most of the fireflies are grouped in a single solution, or when the swarm reaches a pre-defined maximum number of iterations. In general, convergence is usually determined with iteration levels, as actual convergence might require huge amounts of time. The maximum number of iterations should be an optimal value that allows for sufficient convergence time. The swarm would not be expected to have completely converged when termination is based on a maximum number of iterations. However, the swarm is expected to have begun its convergence process and is moving towards stagnation. Hence the optimal solution is identified by

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

$$optima = argmax_{j\in(0,n)}(posVector_j)$$

$$posVector_i = \sum_{j=0}^{n} count_{ij} \quad \forall i = 0,1 \dots n$$

$$count_{ij} = \begin{cases} 1 & if\ pos_i = pos_j \\ 0 & Otherwise \end{cases}$$

Where posi and posj are the coordinate positions of fireflies i and j respectively and n is the number of fireflies.
The node where maximum number of fireflies have converged corresponds to the optimal solution for the test data under consideration.
This process is repeated for each temporal and location based group identified in the previous phase. The optimal package for each group is identified and obtained from the corresponding providers. This enable the usage of multi-cloud for cost and computation effective package selection.

4. Results and discussion

The proposed optimal package selection model is implemented using C#.NET, and the temporal geolocation based grouping mechanism has been implemented using Python. Search space for the proposed Firefly model is constructed using the location based details for the first phase and package details corresponding to 20 distinct requirements for the second phase.
The proposed model has been operated upon 50 distinct requirements and the QoS based results are shown in figure 3. It could be observed from the figure that in most of the requirements, the provided QoS almost matches the required QoS. However, a small difference level is observed, which can be attributed to the impossible perfect matching scenario due to standardized packages available from the cloud providers.

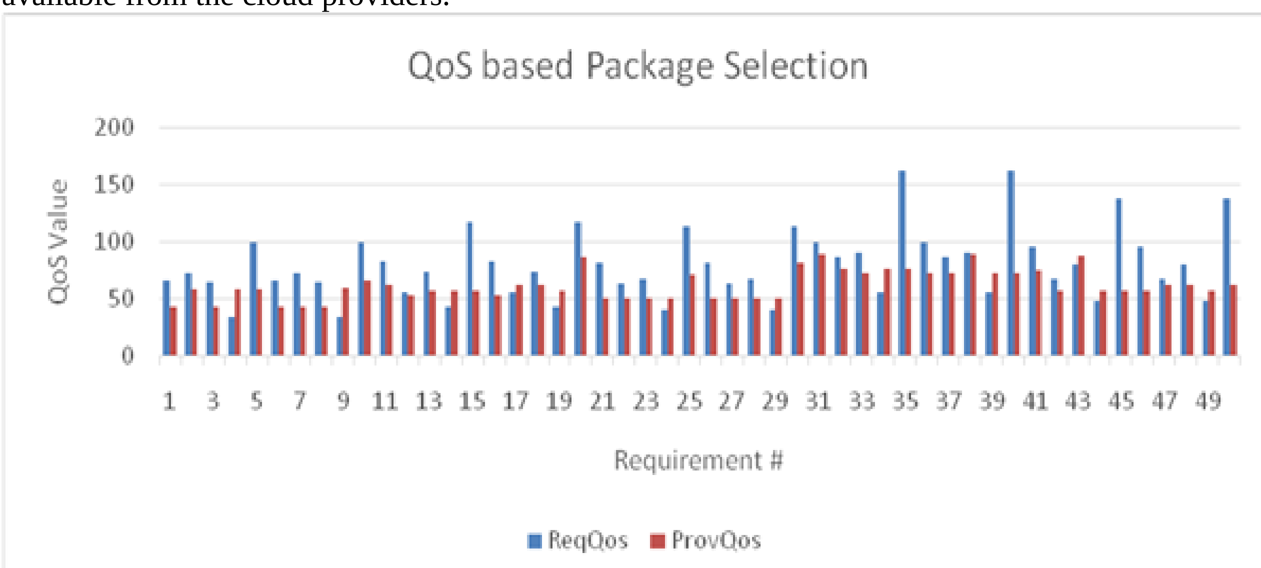

Figure 3: QoS based Package Selection

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

Difference between the provided QoS and the required QoS are shown in figure 4. Positive values depict that the provided QoS is higher than the required QoS, while negative values depict that the provided QoS is lower than the required QoS. It could be observed that most of the transactions exhibit lowered QoS assignments. These are attributed to the incorporated cost constraints provided during the optimization process.

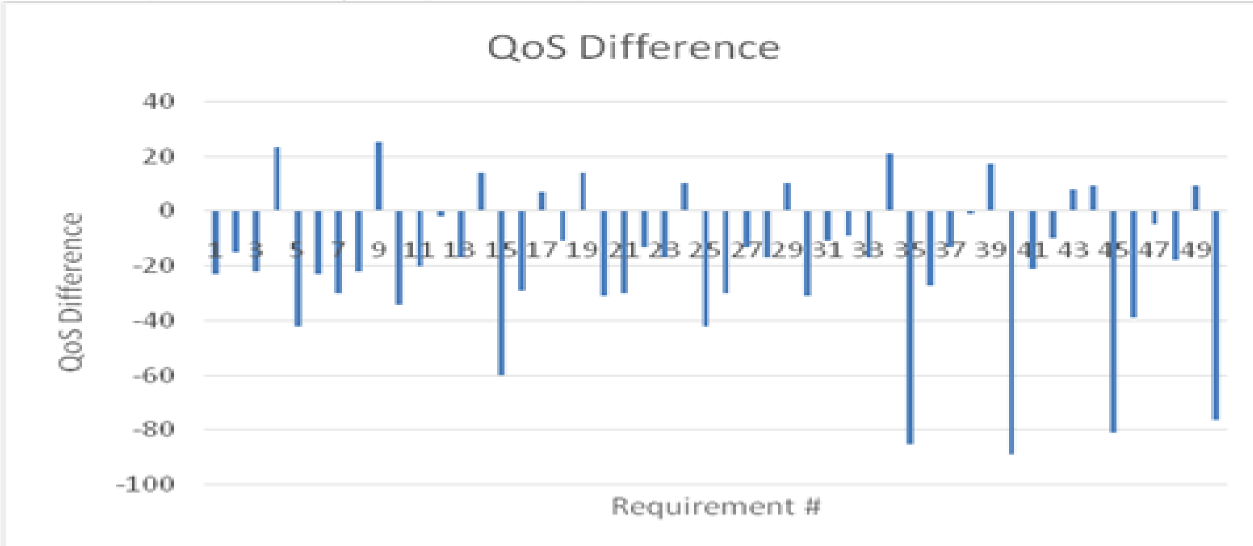

Figure 4: QoS Difference

A comparison of the proposed model with PSO based package selection model in [22] is performed and is shown in figures 5 and 6. A QoS based comparison exhibiting the average QoS of the proposed model and [22] is shown in figure 5. It could be observed that the average QoS provided by the proposed Firefly based model exhibits a low variation of -18.8, while the package selection model proposed in [22] exhibits a high variation level of -419, exhibiting the efficiency of the proposed model.

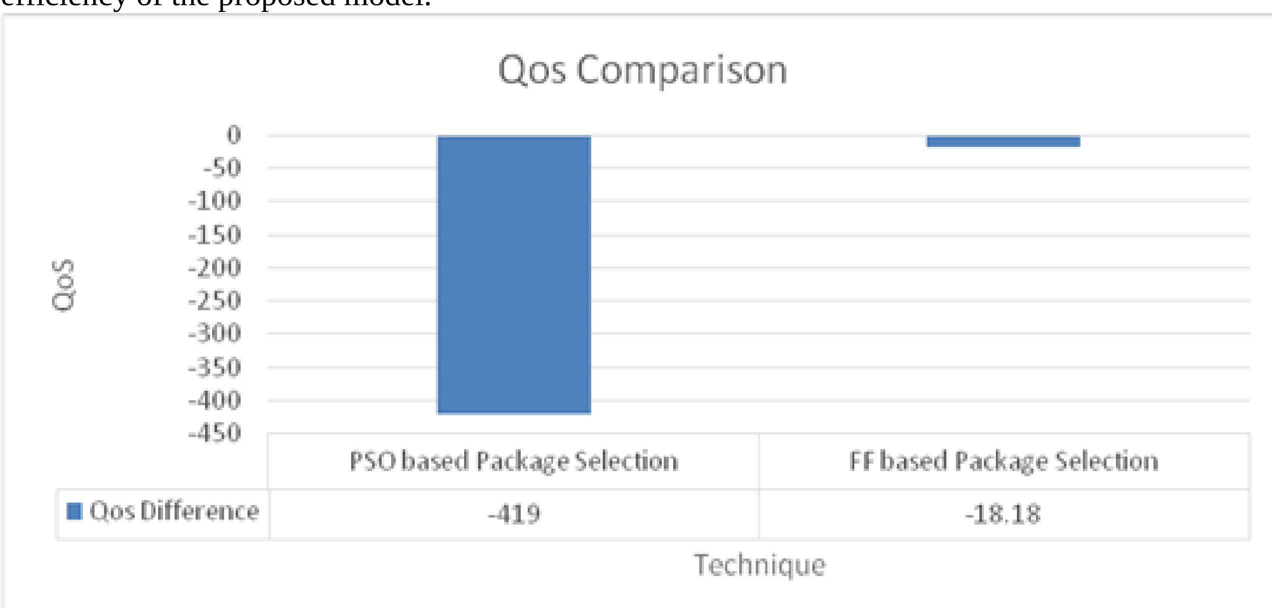

Figure 5: QoS Comparison

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

A time comparison between the proposed Firefly based package selection model and the model proposed in [22] is shown in figure 6. It could be observed from the figure that the proposed Firefly model exhibits a slightly increased time requirement when compared to the package selection model proposed in [22]. This could be attributed to the additional geolocation based processing contained in the proposed Firefly based model. However, the time difference is very less at 0.0045 seconds, hence is negligible.

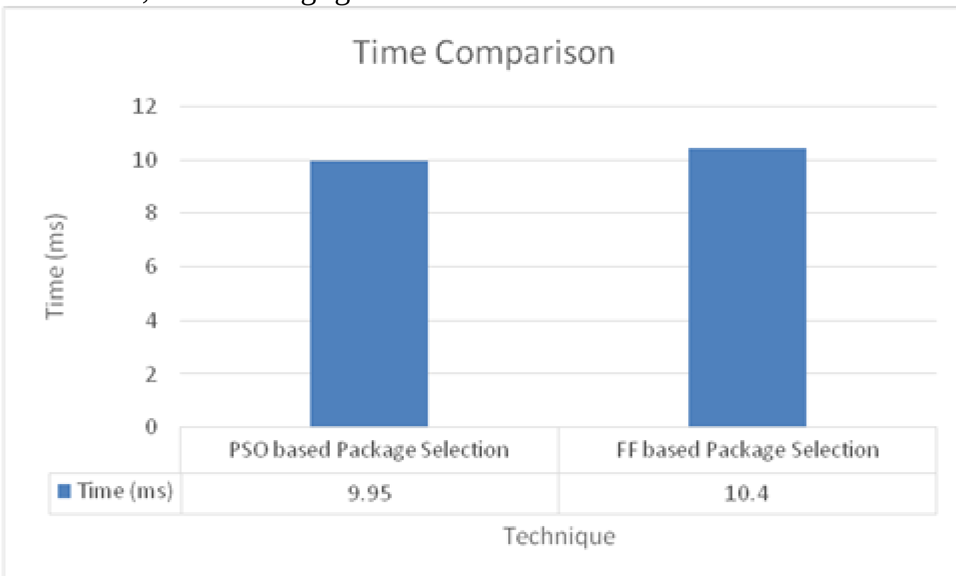

Figure 6: Time Comparison

5. Conclusion

Effective provisioning of resources in cloud environments is one of the major requirements of the current cloud user scenario. However, from a customer perspective, it could be observed that they can avail cost-effective resources when opting for multiple cloud-providers, rather than enabling resource utilization from a single provider. This is a complex task, due to the existence of a large number of providers and a higher number of resource plans.This paper proposes an effective model for resource provisioning in multi-cloud environments. The initial phase groups the log records as temporal and geographical groups and the next phase determines the optimal resource provider and optimal package for the given requirements. Experiments and comparisons indicate effective performances both in terms of providing QoS and in terms of time. Future enhancements to this model are aimed at improving the selection process by hybridizing the Firefly model for improved efficiency

References

[1] Pandey, S., Karunamoorthy, D., & Buyya, R. (2011). Workflow engine for clouds. Cloud computing: principles and paradigms, 321-344..

[2] Manvi, S. S., & Shyam, G. K. (2014). Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications, 41, 424-440.

[3] Jamshidi, P., Pahl, C., Chinenyeze, S., & Liu, X. (2015). Cloud migration patterns: a

multi-cloud service architecture perspective. In Service-Oriented Computing-ICSOC 2014 Workshops (pp. 6-19). Springer, Cham.

[4] da Cruz Ismael, M. A., da Silva, C. A., Silva, G. C., & Ré, R. (2015, November). An Empirical Study for Evaluating the Performance of jclouds. In Cloud Computing Technology and Science (CloudCom), 2015 IEEE 7th International Conference on (pp. 115-122). IEEE.

[5] Panda, S. K., Pande, S. K., & Das, S. (2018). Task Partitioning Scheduling Algorithms for Heterogeneous Multi-Cloud Environment. Arabian Journal for Science and Engineering, 43(2), 913-933.

[6] Panda, S. K., & Jana, P. K. (2015). Efficient task scheduling algorithms for heterogeneous multi-cloud environment. The Journal of Supercomputing, 71(4), 1505-1533.

[7] Panda, S. K., & Jana, P. K. (2018). Normalization-based task scheduling algorithms for heterogeneous multi-cloud environment. Information Systems Frontiers, 20(2), 373-399.

[8] Heilig, L., Lalla-Ruiz, E., & Voß, S. (2016). A cloud brokerage approach for solving the resource management problem in multi-cloud environments. Computers & Industrial Engineering, 95, 16-26.

[9] Paraiso, F., Haderer, N., Merle, P., Rouvoy, R., & Seinturier, L. (2012, June). A federated multi-cloud PaaS infrastructure. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on (pp. 392-399). IEEE.

[10] Quinton, C., Haderer, N., Rouvoy, R., & Duchien, L. (2013, April). Towards multi-cloud configurations using feature models and ontologies. In Proceedings of the 2013 international workshop on Multi-cloud applications and federated clouds (pp. 21-26). ACM.

[11] Heilig, L., Voß, S., & Wulfken, L. (2015). Building Clouds: An Integrative Approach for an Automated. Delivery and Adoption of Cloud Computing Services in Contemporary Organizations, 269.

[12] Coutinho, R. D. C., Drummond, L. M., Frota, Y., & de Oliveira, D. (2015). Optimizing virtual machine allocation for parallel scientific workflows in federated clouds. Future Generation Computer Systems, 46, 51-68.

[13] Mao, L., Li, Y., Peng, G., Xu, X., & Lin, W. (2018). A Multi-Resource Task Scheduling Algorithm for Energy-Performance Trade-offs in Green Clouds. Sustainable Computing: Informatics and Systems.

[14] Wu, J., Guo, S., Li, J., & Zeng, D. (2016). Big data meet green challenges: big data toward green applications. IEEE Systems Journal, 10(3), 888-900..

[15] M. Harchol-Balter, A. Gandhi, Systems and Methods for Managing Power Consumption and Performance of a Processor: US, US8589709, (2013).

[16] Liu, Z., Wierman, A., Chen, Y., Razon, B., & Chen, N. (2013). Data center demand response: Avoiding the coincident peak via workload shifting and local generation. Performance Evaluation, 70(10), 770-791.

Annals of R.S.C.B., ISSN:1583-6258, Vol. 25, Issue 2, 2021, Pages. 1066 - 1075
Received 20 January 2021; Accepted 08 February 2021.

[17] Díaz, J. L., Entrialgo, J., García, M., García, J., & García, D. F. (2017). Optimal allocation of virtual machines in multi-cloud environments with reserved and on-demand pricing. Future Generation Computer Systems, 71, 129-144.

[18] Chaisiri, S., Lee, B. S., & Niyato, D. (2009, December). Optimal virtual machine placement across multiple cloud providers. In Services Computing Conference, 2009. APSCC 2009. IEEE Asia-Pacific (pp. 103-110). IEEE.

[19] Chaisiri, S., Lee, B. S., & Niyato, D. (2012). Optimization of resource provisioning cost in cloud computing. IEEE transactions on services Computing, 5(2), 164-177.

[20] Mark, C. C. T., Niyato, D., & Chen-Khong, T. (2011, March). Evolutionary optimal virtual machine placement and demand forecaster for cloud computing. In Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on (pp. 348-355). IEEE.

[21] Yang, X. S. (2008). Nature-Inspired Metaheuristic Algorithms. Luniver Press. ISBN 1-905986-10-6.

[22] T.M.Nithya, J. Ramya, L. Amudha,"Scope Prediction Utilizing Support Vector Machine for Career Opportunities", International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249- 8958, Volume-8 Issue-5, June 2019, pp.2759-2762.

[23] L. Amudha, Dr.R.PushpaLakshmi, "Scalable and Reliable Deep Learning Model to Handle Real-Time Streaming Data", International Journal of Engineering and Advanced Technology, ISSN: 2249 – 8958, Volume-9 Issue-3, February, DOI: 10.35940/ijeat.C6272.029320, 2020, Retrieval Number: C6272029320/2020©BEIESP, pp. 3840 – 3844

[24] T.M.Nithya, K.S.Guruprakash, L.Amudha. (2020). DEEP LEARNING BASED PREDICTION MODEL FOR COURSE REGISTRATION SYSTEM. International Journal of Advanced Science and Technology, 29(7s), 2178-2184

[25] Nithya, T.M., Chitra, S.. (2020). Soft computing-based semi-automated test case selection using gradient-based techniques. Soft Computing. 24. 12981–12987 (2020)

[26] K.S.Guruprakash, R.Ramesh, Abinaya K, Libereta A, Lisa Evanjiline L, Madhumitha B. (2020). Optimized Workload Assigning System Using Particle Swarm Optimization. International Journal of Advanced Science and Technology, 29(7), 2707-2714