International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 6 (2017), pp. 1441-1453

http://www.ripublication.com

Flexible Cost based Cloud Resource Provisioning using Enhanced PSO

K.Mani[1] and R.Mohana Krishnan[2]

Associate Professor, Department of Computer Science, Nehru Memorial College, Puthanampatti-621 007, India[1]

Research Scholar, Department of Computer Science, Nehru Memorial College, Puthanampatti-621 007, India[2]

Abstract

Cloud centers operate based on virtualization and distribution of resources. Effective distribution and automatic scaling are the major advantages provided by cloud environments. This paper proposes an effective provisioning of resources with it major significance on minimizing the cost, maximizing the quality parameters and improving resource utilization. Resources are provided as packages and not independent entities. This mandates the need for optimal QoS matching rather than accurate matching. Identifying and matching QoS requirements is the major operation to be carried out in the proposed architecture. This work proposes a modified PSO technique known as EPSO. The local optimal issue in PSO is overcome by incorporating Simulated Annealing as the global solution identifier component. The proposed EPSO not only exhibits higher efficiency, but also faster and better results on comparison with existing techniques incorporating a meta-heuristic optimizer. Experiments show that the proposed EPSO exhibits low time requirements and prediction levels leading to high performance.

Keywords: Resource Provisioning; QoS; Cost; PSO; EPSO; MCDM; MIP

1. INTRODUCTION

Cloud computing has emerged as a popular computing model due to its flexible and pay-as-you-go nature. It provides access to computational resources through virtualization. The major advantage of using cloud based resources is that it eliminates the need for dedicated hardware. Hence organizations with varying requirements finds utilizing cloud resources, as an elegant and cost effective solution. The advantages cloud providers gain is that they virtualize the cloud environments and provide it to users on a shared basis. As customers are multiplexed on the same physical hardware and software architecture, it becomes more profitable for the cloud providers.

Cloud environments usually offers utilities such as Software as a Service (SaaS), Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and several other such services. However, SaaS, PaaS and IaaS offers to be the most utilized services. Resource allocation in virtual environments are the major requirements for any cloud architecture. Efficiently performing the allocation process proves to be advantageous for both the consumer and the provider. However, the resource heterogeneity and unpredictable workload levels pose major complications in the allocation process. High scalability levels offered by cloud architectures depicts theoretically unlimited computational resources to the consumers. Hence they are offered automatic resource upscaling, which acts as the major advantage of a cloud system. However, high adaptation levels of cloud based systems and high utilization levels of resources has brought more challenges for the resource allocation systems.

Resource provisioning in the virtual cloud environment is defined as the process of allocating services to the customer. However, in real time, consumers are not provided such flexible environments to define their resource requirements. Instead, the requirements are already grouped in terms of processing, memory and network configurations and are available as packages. The consumer is required to select any one of these configurations. However, the requirement and available QoS values do not exactly match, hence the selection process becomes an optimization problem. This paper presents a swarm based optimizer that effectively identifies the appropriate packages for user's requirements such that the QoS values closely match with low of QoS difference levels.

2. RELATED WORKS

Cloud resource provisioning is usually carried out based on several major requirements such as quality of service, energy, resource utility, load balancing, cost etc. This section discusses recent contributions in provisioning grouped based on the requirements.

An agent based resource allocation technique for cloud resource provisioning is presented by Bajo et al. in [1]. This technique leverages the advantages of the virtualization environment by upscaling the resources directly rather than moving nodes. This results in effective load balancing and reduced computational time. A similar load balancing scheme was proposed by Shaw et al. in [2].

A modelling scheme for allocating cloud resources is presented by Vakilinia et al. in [3]. It presents a view of the type and number of resources required to create cloud centers based on the velocity of the requirement requests. A similar experimental based approach is proposed by Wolke et al. in [4].

An energy and QoS aware resource allocation technique is presented by Peng et al. in [5]. This technique has its major focus on energy conservation, hence proposes to reduce energy losses occurring due to resource heterogeneity. Other energy aware techniques include heuristic based resource allocation by Beloglazov et al. in [6] and Hasan et al. in [7]. A Genetic Algorithm (GA) based energy aware scheme is proposed, which utilizes the power efficiency based on *powerMark*, a novel metric introduced by Peng. A task scheduling and resource allocation technique for cloud is proposed by Yi et al. in [8]. This technique operates on disassembly. An ACO based resource management scheme for cloud is presented by Tiwari et al. in [9]. This is a resource management based optimization scheme applied on virtual machines in cloud. Another similar nature inspired resource allocation technique is proposed by Jason et al. in [10], GA based technique proposed by Hallawi et al. in [11]. Several allocation algorithm based techniques were also proposed such as best first based VM allocation scheme for cloud resource allocation is proposed by Shrivastava et al in [12], modified round robin algorithm for resource allocation in cloud by Pradhan et al. in [13] and bin packing based dynamic resource allocation scheme proposed by Wolke et al. in [14]. Other popular resource allocation schemes include dynamic resource allocation by Saraswathi et al. in [15], Hasan et al. in [7] and Patel et al. in [19].

3. EPSO BASED CLOUD PROVISIONING

Cloud provisioning is the process of assigning appropriate resources for the consumer such that overutilization and under-utilization are avoided. Requirements from cloud include CPU, network, node, storage and VM. Each resource will also be available at varied configuration levels in the virtual machines. Identifying the accurate virtualization requirements for these resources will avoid underutilization and overutilization states. However, these factors are unavoidable, as cloud providers assign resources to consumers in terms of packages and not as individual entities. Hence accurate matching of resources is not possible, however, the QoS difference between requirements and the provisioning packages can be minimized to the maximum possible extent.

Resource provisioning in cloud can be categorized as a Multi-Criteria Decision Making (MCDM) problem. Several optimization parameters such as cost, Quality of Service parameters, energy, resource utility levels, load balancing levels, etc. can be used for optimization. This work considers cost, effective resource utility and QoS as the major parameters of importance. Depending on customer’s applications, several combinations of parameters are used to identify the QoS sufficiency. Prominently used quality parameters are listed in table 1.

Table 1. QoS Parameters Considered for Evaluation.

QoS Parameters Considered for Evaluation	*Description*
Bandwidth (Bw)	*Maximum data transmission level supported by the cloud system.* $BW = \sum_{resource^i} \left(\frac{Size}{Capicity}\right)$
Computation Capability (CC)	*It is defined as the maximum amount of processing that can be handled in a given time period.* $CC = \frac{max_{task}(Exe_{time}) - min_{task}(Exe_{time})}{Avg_{task}(Exe_{time})}$
Availability (Av)	*It refers to a usable resource at a point in the cloud.* $Av = \sum_{resource^i} \frac{MTBM}{MTBM+MTTR}$ *MTBM represents the Mean Time Between Maintenance and MTTR represents the Mean Time to Repair a resource*
Reliability	*It is the ability of a task presented to cloud within a stipulated time.* $R = \frac{\sum_{task} Exe_{time}}{Total_{time}}$
Throughput	*It is defined as the number of tasks to be completed in a defined time interval.* $T = \sum_{task} Exe_{time}$

Though there are several other QoS parameters available for cloud provisioning, these are considered to be the most significant entities required for all consumer application irrespective of the type of the application.

Cost is defined as the monetary value of the package determined by the cloud provider. The cost to be paid by the consumer is defined by

$$Cost_{Total} = \sum_{resource^i} (C_i * T_i)$$

where Ci represents the cost of resource i per unit time and Ti represents the time of utilization of $resource_i$

User requirements are provided to the cloud system in terms of these parameters. The input requirements are analyzed and their corresponding QoS levels are identified. This formulates the first phase of the proposed resource provisioning system.

The next phase deals with identifying the best package to be assigned for the consumer's requirements. This is a MCDM problem, requiring maximization and minimization of parameters using Mixed Integer Programming (MIP). The core formulation of the fitness function is as follows

Minimize:

$$z = \sum_i c_i + \sum_i Pqos_i$$

Subject to

$$\sum_i c_i \leq c_p \; \forall \, i$$

$$\sum_i Pqos_i \geq Rqos_p \; \forall \, qos$$

Where c_i is the cost of the i^{th} resource, $Pqos_i$ is the sum of qos properties of resource i and $Rqos_i$ is the sum of qos properties of the customer's requirement. QoS of a resource is calculated using the below equation

$$Xqos = Ns_{bw} + Ns_{cc} + Ns_{Av} + Ns_R + Ns_T + (-Ns_L)$$

Where Ns_x is the normalized value of the resource parameter.

The optimization process is performed using the proposed Enhanced Particle Swarm Optimization (EPSO). Particle Swarm Optimization (PSO) [16, 17] is a meta-heuristic technique designed to formulate optimal solutions for problems by iteratively refining candidate solutions. However, the major downside of PSO is that it has high probability of getting struck in local optima. Hence this work proposes Enhanced PSO (EPSO). PSO is hybridized with Simulated Annealing to reduce the issue of local optima. The formulation defined in eq is considered as the fitness function for EPSO.

PSO operates based on particles moving in the search space. The search space is created by including the parameters of the packages and the requirements. Particle distribution is carried out on the nodes containing the requirements data. The particle best (*pbest*) and global best (*gbest*) are initialized to default values and particle movement is initiated. The best solution obtained until now by the particle under analysis is contained in *pbest*, and the best solution obtained by the swarm is contained in *gbest.* Movement of particles is defined by its velocity component. The initial velocity component is defined by

$$V_i \sim U(-|b_{up} - b_{lo}|, |b_{up} - b_{lo}|)$$

where b_{up} and b_{lo} are the upper and lower bounds of the search space.
This initial velocity component aids in the initial movement of particles. PSO operates on continuous space, while the problem domain requiring solutions is discrete. Hence the particle positions are discretized to the nearest occurring node.

The native PSO updates its *pbest* and *gbest* components using the new locations of the particles. However, it was identified that this operational mode leads to the particles getting stuck in local optima. Hence PSO is hybridized with Simulated Annealing to obtain EPSO. According to EPSO, only the *pbest* component is updated in this stage. After the *pbest* updates of all the particles is complete, the *gbest* is updated using all the available *pbest* components and the *gbest* component.

Simulated Annealing is yet another metaheuristic technique, aids in optimization by approximating the global optimum of the given function. The working of simulated annealing is described in the algorithm below.

Simulated Annealing(gbest,pbest,p)

1. *Let s = gbest*
2. *For k = 1 through p :*
   a. *T ← pbest*$_k$
   b. *Pick a random pbest (pb),* s_{new} *← pb*
   c. *If P(E(s), E(*s_{new}*), T) ≥ random(0, 1), move to the new state:*
      - $s \leftarrow s_{new}$
3. *Output: the final state s*

Where P(e,e',T) was defined as 1 if e' < e and exp(-(e'-e)/T) otherwise.

The final state *s* gives the *gbest* component. This is followed by further particle movement directed by the new velocity component. The updated velocity component is given by

$$V_{i,d} \leftarrow \omega V_{i,d} + \varphi_p r_p (P_{i,d} - X_{i,d}) + \varphi_g r_g (g_d - X_{i,d})$$

where r_p and r_g are the random numbers, $P_{i,d}$ and g_d are the parameter best and the global best values, $x_{i,d}$ is the value current particle position, and the parameters ω, φ_p, and φ_g provides the significance of current velocity, *pbest* and *gbest* vales respectively and are selected by the practitioner. This process is continued until solution convergence. The *gbest* value is considered as the optimal package for the requirement node from which the process was initiated.

The corresponding virtualized resources are assigned for the customer requirement. Resource assignment is followed by utility monitoring, which forms the final phase of the virtual resource provisioning architecture. As discussed earlier, resource allocation in cloud environments is not a single process. The extended scalability levels offered by cloud servers require continual monitoring of the resource utility levels in-order to automatically upscale and downscale resources. However, during the process of upscaling and downscaling, it is mandatory to identify the appropriate virtual resource levels such that multiple consecutive scaling is avoided. Although the identification procedure is similar to the previously discussed process, it is mandatory for the resultant package to be determined swiftly such that none of the user's process exhibit hiccups during the transition. It was observed that EPSO based package selection satisfies the real-time constraints of the problem, hence providing faster selection time. Monitoring of resource utilization levels is continued until the resources are submitted back to the cloud provider.

4. RESULTS AND DISCUSSION

Experiments were conducted by implementing EPSO in C#.NET. Package and requirements dataset used in [1] were used for analysis of the proposed architecture and result comparison. Ten defined packages and 100 distinct requirements were provided to the processing module and the results were obtained. Results were analyzed in terms of time taken for processing and in terms of QoS values.

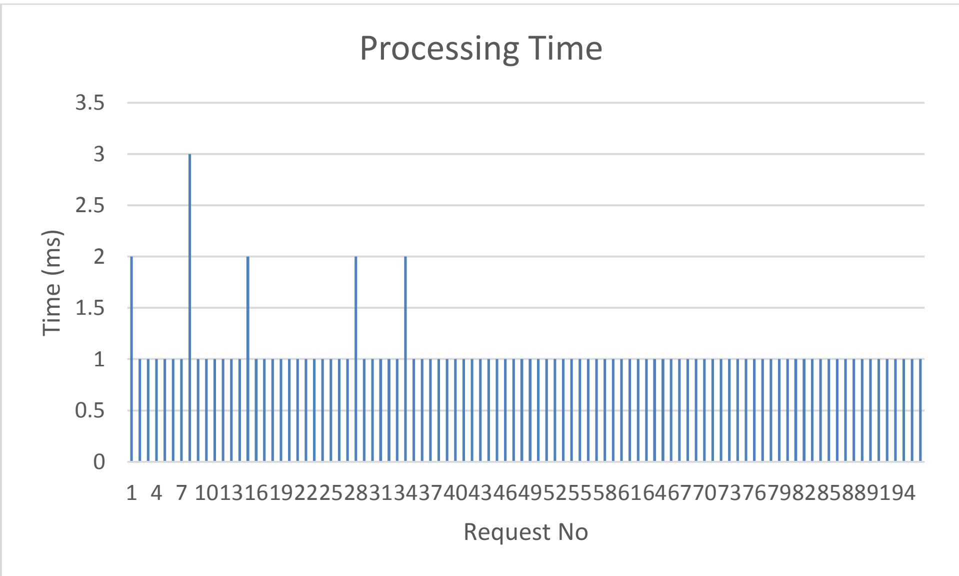

Figure 1: Processing Time

Time taken for processing the requirements is measured from the time of request to the time of identification of the package (figure 1). It was observed that most requests took ~1ms for processing. A very few requests exhibited time requirements of 2 ms and 3 ms. However, such occurrences were found to be rare events. An average identification time of 1ms exhibits that the proposed architecture exhibits high correlation to real-time processing speeds.

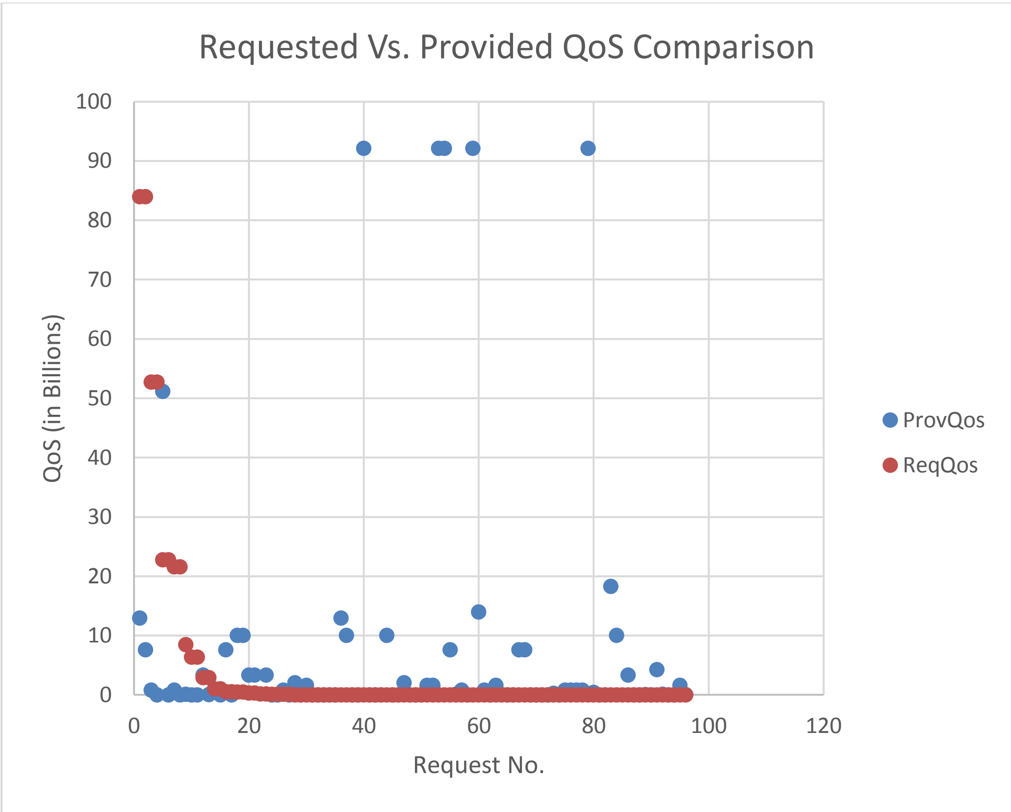

Figure 2: Comparison between Requested and Provided QoS

A comparison between the requested and provided QoS levels is shown in figure 2. It could be observed that in most requirements, the provided QoS is higher or almost equal to the required QoS. The resource utilization levels are shown in figure 3. It could be observed that underutilization levels of 73% and overutilization levels of 26% is shown, indicating that most of the assigned packages correspond to higher configurations, while 26% of the assigned packages correspond to lower configurations compared to the requirements. Perfect package matching is not possible due to the defined package configurations. Hence overutilization and underutilization cannot be avoided. However, it is of paramount importance to assign packages with low QoS differences such that too much of resource wastage is avoided.

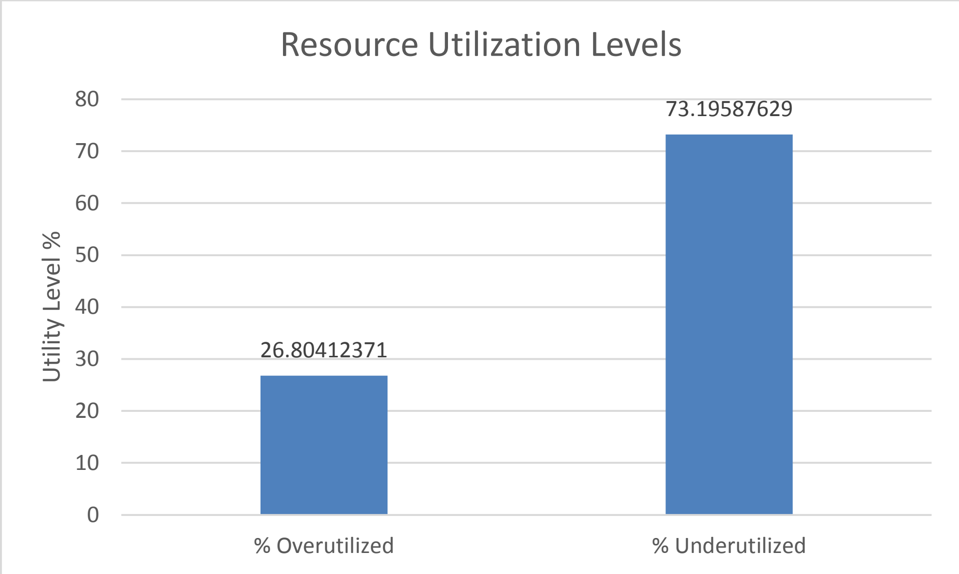

Figure 3: Resource Utilization Levels

QoS difference between the provided and the required levels is shown in figure 4. It could be observed that except for a few requests, most of the requests exhibit very low QoS difference, hence depicting effective allocations. Further, the negative allocation levels are also maintained very low, hence exhibiting low upscale requirements.

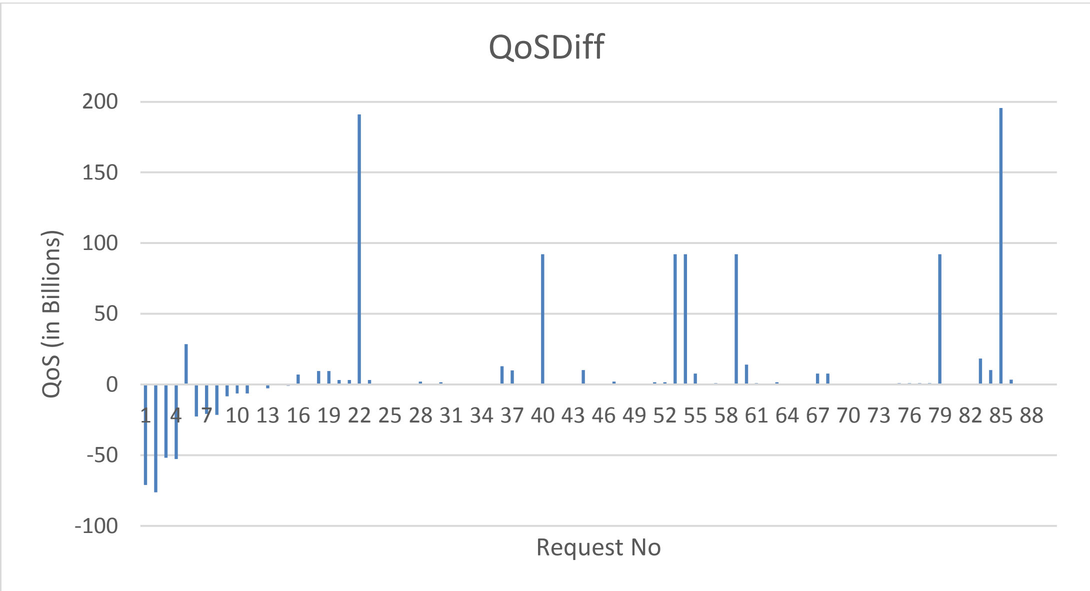

Figure 4: Observed QoS Difference Levels

A comparison of the proposed EPSO with the technique proposed by Madhumathi et al. [18] is shown in figures 5 and 6. Madhumathi et al. proposed a cloud provisioning technique based on modified ACO as the base metaheuristic algorithm. Comparison is carried out in terms of time taken for the optimization process and the average QoS difference levels exhibited by the techniques.

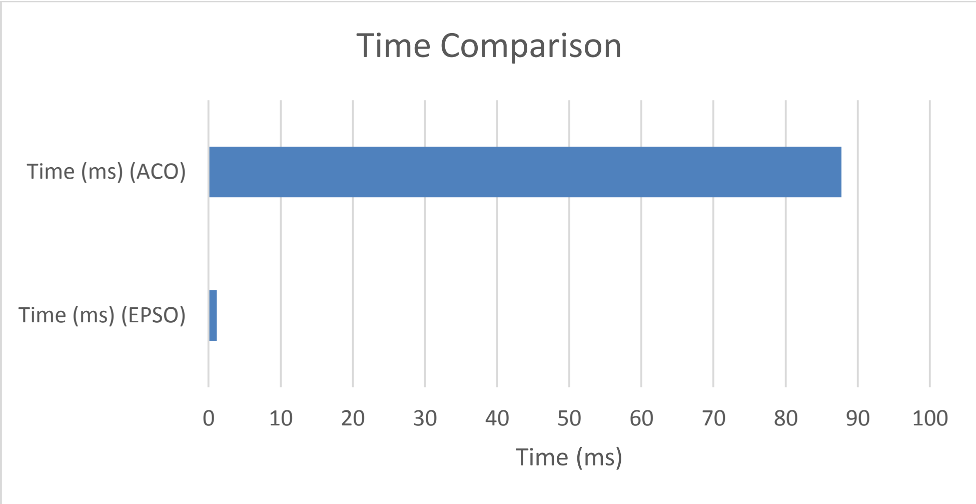

Figure 5: Time Comparison

A time comparison of ACO and EPSO in figure shows that ACO requires ~87ms for the optimization process, while it is completed in ~1ms in EPSO, exhibiting a huge scale of efficiency. A comparison of the average QoS shows that the proposed EPSO exhibits positive QoS difference levels depicting that the assignment is mostly higher than the requirement. However, the QoS difference levels of ACO shows negative levels, depicting that most of the assigned packages have QoS levels lower than the requirements.

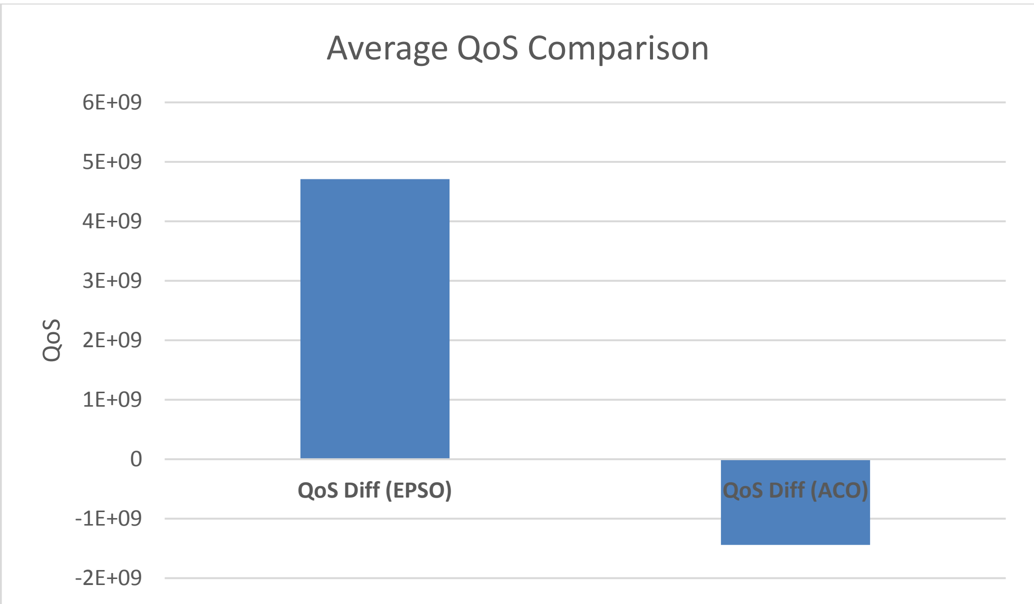

Figure 6: Average QoS Comparison

5. CONCLUSION

Virtualized resources provide major advantages to cloud systems. However, appropriately allocating them is one of the major challenges facing the system. The resources provided by cloud providers are in the form of packages and not individual entities. Hence a perfect match with user requirements is not possible. The package selection models should identify the most appropriate package that exhibits least difference from the required QoS and cost. Further, identifying the appropriate resource levels during an upscale condition proves to be challenging. This paper proposes a fast and effective cloud resource provisioning system that operates to identify optimal packages both during the initial package selection phase and during the upscale condition. Upscale condition requires real time resource identification, as users should not experience hiccups in processing during the selection and transition periods. The proposed EPSO exhibits high selection levels enabling its usage in both the scenarios. Experiments conducted using EPSO exhibits high performances, and the comparisons indicates high efficiency levels both in terms of time and in terms of resource selection.

REFERENCES

[1] Bajo, J., De la Prieta, F., Corchado, J.M. and Rodríguez, S., 2016. A low-level resource allocation in an agent-based Cloud Computing platform. Applied Soft Computing, 48, pp.716-728.

[2] Subhadra Bose Shaw. 2017. Balancing Load of Cloud Data Center using Efficient Task Scheduling Algorithm. International Journal of Computer Applications 159(5), pp. 1-5

[3] Vakilinia, S., Ali, M.M. and Qiu, D., 2015. Modeling of the resource allocation in cloud computing centers. Computer Networks, 91, pp.453-470.

[4] Wolke, A., Bichler, M., Chirigati, F. and Steeves, V., 2016. Reproducible experiments on dynamic resource allocation in cloud data centers. Information Systems, 59, pp.98-101.

[5] Peng, Y., Kang, D.K., Al-Hazemi, F. and Youn, C.H., 2017. Energy and QoS aware resource allocation for heterogeneous sustainable cloud datacenters. Optical Switching and Networking, 23, pp.225-240.

[6] Beloglazov, A., Abawajy, J. and Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 28(5), pp.755-768.

[7] Hasan, M.S. and Huh, E.N., 2013. Heuristic based Energy-aware Resource Allocation by Dynamic Consolidation of Virtual Machines in Cloud Data Center. TIIS, 7(8), pp.1825-1842.

[8] Jiang, H., Yi, J., Chen, S. and Zhu, X., 2016. A multi-objective algorithm for task scheduling and resource allocation in cloud-based disassembly. Journal of Manufacturing Systems, 41, pp.239-255.

[9] Anupama Tiwari, Pankaj Richhariya and Satyaranjan Patra. 2017. Ant Colony

based Cloud VM Allocation and Placement Approach for Resource Management in Cloud. International Journal of Computer Applications 158(4). Pp.8-12

[10] Sebagenzi Jason and Suchithra R..2017. Nature Inspired Algorithms for Load Balancing in Cloud Computing. International Journal of Computer Applications 160(4), pp. 7-14

[11] Hallawi, H., Mehnen, J. and He, H., 2017. Multi-Capacity Combinatorial Ordering GA in Application to Cloud resources allocation and efficient virtual machines consolidation. Future Generation Computer Systems, 69, pp.1-10.

[12] Shirvastava, S., Dubey, R. and Shrivastava, M., 2017. Best Fit based VM Allocation for Cloud Resource Allocation. International Journal of Computer Applications, 158(9).

[13] Pradhan, P., Behera, P.K. and Ray, B.N.B., 2016. Modified Round Robin Algorithm for Resource Allocation in Cloud Computing. Procedia Computer Science, 85, pp.878-890.

[14] Wolke, A., Tsend-Ayush, B., Pfeiffer, C. and Bichler, M., 2015. More than bin packing: Dynamic resource allocation strategies in cloud data centers. Information Systems, 52, pp.83-95.

[15] Saraswathi, A.T., Kalaashri, Y.R.A. and Padmavathi, S., 2015. Dynamic resource allocation scheme in cloud computing. Procedia Computer Science, 47, pp.30-36.

[16] Kennedy, J. and Eberhart, R., 1942. Ieee (1995). Particle swarm optimization. In 1995 IEEE International Conference on Neural Networks (ICNN 95). Perth, Australia (pp. 1942-1948).

[17] Shi, Y. and Eberhart, R., 1998, May. A modified particle swarm optimizer. In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence, The 1998 IEEE International Conference on (pp. 69-73). IEEE.

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

[19] Patel, A. Patel, H. Patel, N. 2017. Power-Aware Scheduling For Urgent Tasks in Cloud Environment. International Journal of Computational Intelligence Research, 13, pp. 45-53

BIOGRAPHY

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 21 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.