CrossMark

Competent resource provisioning and distribution techniques for cloud computing environment

A. Suresh[1] · **R. Varatharajan[2]**

Abstract The principal intension of the investigation is to provisioning the resources and effectively allocates them. Initially, the resource are identified and analyzed and then clustered. Cluster the resource using the kernel fuzzy c-means clustering algorithm. After the clustering algorithm, the resources are allocated using resource provider. In the proposed method resource allocation is done with the help of modified cloud resource provisioning algorithm. With the help of optimization technique, the traditional OCRP algorithm is improved. Modified cloud resource provisioning algorithm is selecting the resource with minimum cost using optimization. Here particle swarm optimization algorithm is used to select the optimal resource with minimum cost. Finally the resource provisioner allocates the resource in an effective manner. The enactment of the proposed method is evaluated by means of cost value. The proposed technique is performed with the mighty assistance of the Cloud simulator in the working platform of Java software.

Keywords Kernel fuzzy c-means · Cloud resource provisioning · OCRP · Particle swarm optimization

✉ A. Suresh
    prisu6esh@yahoo.com

    R. Varatharajan
    varathu21@yahoo.com

1   Department of Computer Science and Engineering, Nehru Institute of Engineering and Technology, "Nehru Gardens", T.M. Palayam, Coimbatore 641 105, India

2   Department of Electronics and Communication Engineering, Sri Lakshmi Ammaal Engineering College, Chennai, India

1 Introduction

Resource allocation is an arrangement for utilizing accessible resources, for instance human resources, particularly in the close term, to accomplish objectives for the future. Cloud computing is internet-based computing in which extensive groups of remote servers are networked to permit the centralized data storage, and online access to computer services or resources. Cloud framework should be under a payment model, in order to maintain a strategic distance from users' over—interest of their resources against their actual needs [1]. Distributed computing suppliers pass on application through the Internet, which are gotten to from web program, while the business programming and data are secured on servers at a remote territory [2]. Applications can be utilized as a part of Cloud computing with no installation of software. The users can access the Internet and send messages anyplace in the world [3]. Google gives an efficient and scalable answer for working extensive scale data [4]. Centralized storage, memory, processing and bandwidth are more efficient computing that are permitted in Cloud computing. The accomplishment of clouds has been driven to some degree by the utilization of virtualization as their hidden innovation [5].

Cloud helps consumers to outsource computation, storage and different tasks to third party cloud providers and pay just for the resources utilized [3,6]. Furthermore, find out about how a cloud service provider is best multiplexing its virtual resources. Accordingly a cloud model is required to have a scale all over in order to deal with the load variation [7]. At the point when a job is submitted to the clouds, it is generally partitioned into a few tasks. Taking after two inquiries are have to consider when applying parallel processing in executing these tasks: (1) How to distribute resources to tasks; (2) Task are executed in what order in cloud; and (3) How to

schedule overheads when VMs prepare, terminate or switch tasks. Task planning and resource allocation can take care of these three issues [2].

Present day internets applications are perplexing software actualized on multi-level designs [8]. The task booking application is not appropriate for use in a cloud computing environment because of the little particles [9]. Ren et al. [10] algorithm which adjusts the load among the resources furthermore builds the reliability of the grid environment; this algorithm used to diminish the Communication overhead and response time of the individual job and expand the throughput of the whole framework. This algorithm comprises of two-phases. In the principal phase, the resources are chosen in view of the user deadline and "Fault Tolerant Factor" (FTF) of the resources [11–13]. In the second phase, the load balancing algorithm is connected to determine the status of the chose resources and afterward the jobs are scheduled to the resources if and just if the load is adjusted [14].

2 Literature survey

Cloud computing contributors stipulate their sources into dissimilar kind of virtual machine (VM) examples which are implemented to the consumers for specific period. Zaman and Grosu [15] have anticipated the difficulty of enthusiastically stipulating VM examples in clouds to create advanced turnover, whereas formative the VM distribution among a combinatorial auction-based system. They also include intended a system known as CA-PROVISION to explain the difficulty. They carried out widespread model research by actual workloads to estimate their system. The outcome demonstrates that CAPROVISION could successfully confine the marketplace requirement, stipulation the computing resources to counterpart the requirement, and produce advanced profits than CA-GREEDY, mainly in elevated requirement situations. A few of the short requirement situations, CAGREEDY execute enhanced than CA-PROVISION of turnover but not in consumption and fraction of provide consumers. They have fulfilled that an extremely effectual VM example stipulation and distribution scheme could be intended merging these two combinatorial auction-based systems. They seem advance to locating a confidential cloud and executing such a scheme in the upcoming.

Headed for this novel design where every contributor may separately operate as together source purchaser and contributor, Di and Wang [16] have projected an abundant circulated, VM-multiplexing source distribution system to handle decentralized sources. Their method not only accomplishes exploited source consumption by the proportional share model (PSM), but also conveys provably and adaptively finest effecting effectiveness. They also contain intend a fresh multi quality choice uncertainty procedure

for establishing experienced nodes. Dissimilar to obtainable clarification which frequently produce immense communication per demand, their procedure construct only one inconsequential uncertainty communication per assignment on the Content Addressable Network (CAN). It carried out successfully to discover for every assignment its competent sources underneath a randomized strategy that moderate the conflict between clients. They exposed the SOC by their optimized algorithms could formulate an enhancement by 15-60 percent in method throughput than a P2P Grid representation. Their result also demonstrates moderately elevated compliance in an active node-churning.

In recent times, fusion cloud computing model has been broadly promoted as a hopeful result for software as a service (SaaS) contributors to successfully take place the active client requirements. Li et al. [17] have offered on augmenting the functional charge for the fusion cloud model by hypothetically investigate the difficulty among a Lyapunov optimization outline. That permits us to intend an online active stipulation algorithm. In this manner, their method could tackle the real-world confront where no a priori detail of communal cloud hiring cost was obtainable and the upcoming possibility supply of client requirements was mysterious. They also contain accomplished widespread investigational research depend on a group of real-world data, and the outcome verify that their algorithm could act efficiently in dropping the functional cost.

Zhu and Agrawal [18] have presented the employ of cloud sources for a class of adaptive functions, where there could be function exact flexibility in the calculation that may be preferred. Additionally, there may be a preset time-limit and also a source account. In these limitation, such adaptive function necessitate exploiting their quality of service (QoS), extra accurately, the value of an application-specific promote task, by enthusiastically altering adaptive limitation. They include existing the plan, execution, and estimation of an outline that could maintain such active variation for function in a cloud computing atmosphere [19,20]. The important module of their outline was a multi-input-multi-output response control model-based active source stipulating algorithm which accepts support knowledge to regulate adaptive limitation to promise the finest function profit contained by the time limitation. After that a qualified source representation modifies source distribution consequently to gratify the account. They include estimate their outline among two real-worlds adaptive function, and contain established that their method was effectual and origin a very short operating cost.

Workflow technologies are turn into a key vehicle for simple and competent expansion of systematic functions. Byun et al. [21] have recommended design for performing workflow function on expandable computing sources.

Moreover, they encompass projected the foundation algorithm such as PBTS for computing the least source competence to accomplish a workflow in a specified target. The foremost part of our study was that PBTS bridges the opening among workflow management scheme and the state-of-the-art source stipulating atmosphere. By this outline, scientists were talented to utilize cloud on discount extra functional cost by efficacy source demanding for compensation strategy. PBTS calculates source competence per time-partition therefore that it diminish the source cost, fulfilling the target. Additionally, PBTS performs several workflows of DAG organization that contain single, data comparable and even MPI-like responsibilities. At last, PBTS resolve not only the source competence but also the time program and the host necessities of responsibilities. The trial outcome by an amount of artificial workflows and numerous actual science workflows display that PBTS calculates the source competence near to the hypothetical small bound.

A service-level agreement (SLA) present warranty for precise worth attributes to the clients of services [8]. Though, existing SLAs obtainable by cloud transportation contributors do not tackle reaction time, which, from the client's point of view, is the significant worth feature for Web applications. Iqbal et al. [22] have projected a method and explained a prototype scheme for mechanical detection and declaration of blockage and mechanical detection and declaration of in excess of stipulation in multi-tier function congregated on a cloud. It also recommends a technique for recognizing and withdrawing over-provisioned sources in multi-tier cloud-hosted Web applications. They verified the possibility of the method in trial estimation by a check bed EUCALYPTUS based cloud and a fake workload. Mechanical blockage recognition and declaration in active source organization has the probable to permit cloud transportation contributors to supply SLAs for Web applications that promise exact reaction time necessities though diminishing source consumption. Their investigational outcome have revealed that whereas they evidently cannot offer a SLA promising a precise reaction time by an approximate weight stage for a multi-tier Web application by fixed source distribution, their adaptive source provisioning technique could permit us to propose such SLAs.

3 Problem identification

Resource provisioning is essential issue in cloud computing and in the earth of heterogeneous clouds. The principle target of resource provisioning system is to supply cloud consumers an arrangement of computing resources for processing the jobs and putting away the data. The general issue of existing resource provisioning procedure is given beneath,

- Dynamic resource provisioning does not work for testing on genuine cloud–based framework and over a few spaces.
- A basic test in coordinating work process advancements with resource provisioning advances is to determine the perfect measure of resources required for the execution of work processes in order to minimize the money related expense from the viewpoint of users and to boost the resource use from the point of view of resource providers.
- The real issue of resource provisioning is load balancing and the execution time.
- If the measure of resources is sufficiently vast, the make range can be lessened by executing free tasks all the while. Be that as it may, an excessive number of resources can prompt low resource use, high booking overheads, and above all high cost.
- Medium to large issue is not relevant in existing resource provisioning algorithm.

These are the main drawbacks of various existing work, which motivate us to do the research on resource provisioning.

4 Proposed method

Cloud computing is a striking model as it allows for the providing of resources on-demand. In cloud computing environment effective resource provisioning is a major issue. The main objective of this research is to provisioning the resources and allocates them in an effective manner. At first, the resource are identified and analyzed and then clustered. Cluster the resource using the kernel fuzzy c-means (KFCM) clustering algorithm. After the clustering algorithm, we allocate the resource using resource provisioner. At first resource owner is inform the available resources to resource provisioner. Then resource provisioner creates the list for the available resources. If cloud consumer is need the available resources for the executed jobs. It creates the VMs for the executed jobs using the VM repository. After that it asks for the resource to resource provisioner. Then resource provisioner checks the available list and then allocates the resource for the executed jobs. If the resources are not available it will ask for the resource owner. Resource owner check the availability based on the cost factor. The overhead of using computing resources provisioned by reservation arrangement is less expensive than that provisioned by on-demand arrangement. With the reservation arrange, the consumer can lessen the aggregate resource provisioning cost. In any case, the best propel reservation of resources is hard to be accomplished because of vulnerability of consumer's future interest and providers' resource costs. To address this issue, an ideal cloud

Fig. 1 Architecture diagram of
proposed method
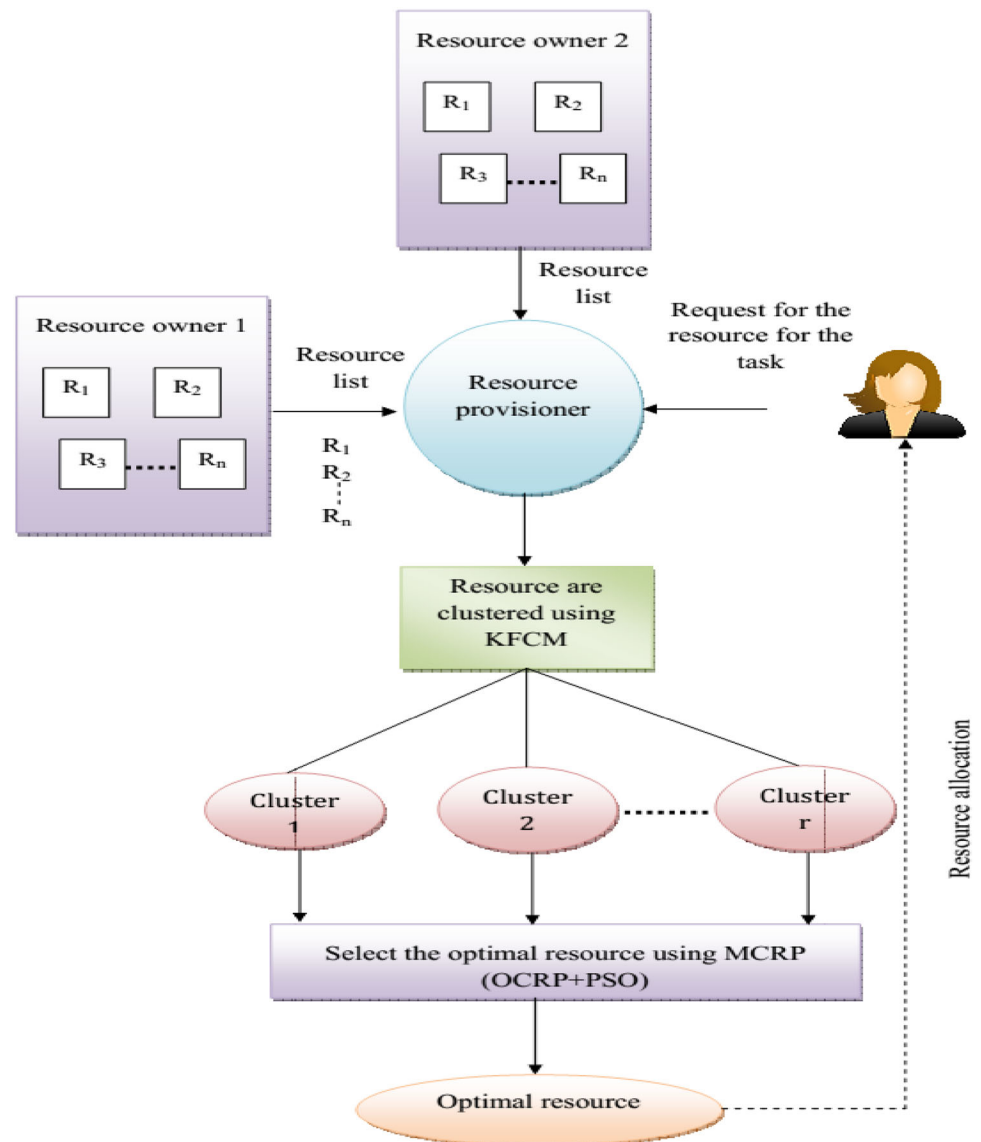
resource provisioning algorithm is proposed by planning
a stochastic programming model. The proposed method is
selecting the resource with minimum cost using optimization
technique. Here PSO is used to select the optimal resource
with minimum cost. Finally the resource provisioner allo-
cates the resource in an effective manner. The block diagram
of the proposed method is shown in Fig. 1 and the detail
explanation of the proposed method is described in further
section.

4.1 Initial setup

In the initial setup user reveal their information and register
their details in the cloud server. If the registration is success
the user or owner gets their corresponding secret key. If the
user was already registered their details in the cloud server

means, the user authentication is verified. If the authentica-
tion is success means the user allowed to access the cloud
server otherwise the user request is rejected. After the verifi-
cation, first the available resources in the cloud are grouped
to allocate the task or jobs in the corresponding resource type.
So the proposed method use kernel fuzzy c means clustering
algorithm to group the resource based on the resource type.
Clear idea of kernel fuzzy c means clustering algorithm is
illustrated in beneath,

4.2 Resource clustering using kernel based fuzzy c
means

Clustering techniques are for the most part unsupervised
techniques that can be utilized to sort out information into
gatherings in light of likenesses among the individual data

items. Most clustering algorithms don't depend on presumptions regular to conventional statistical methods, for example, the basic measurable conveyance of information, and in this way they are valuable in circumstances where little prior knowledge exists. In most existing k-modes clustering methods, the number of initialized cluster modes is the number of target clusters, resulting in potential miss of a true cluster for certain classes of large data sets due to the random selection of initial cluster modes. Avoid the problem of K mode clustering and increasing the precision rate in document clustering KFCM is used.

4.2.1 Kernel based fuzzy c means (KFCM)

Fuzzy c-means (FCM) is a method of clustering that allows information focuses to group in view of closeness and fundamentally utilized in pattern recognition. The change of independent function by minimizing as takes after:

$$OB_m = \sum_{i=1}^{n} \sum_{j=1}^{c} A_{ij}^m \| R_i - C_j \|^2, \tag{1}$$

where A_{ij}^m, m is the real number >1, is the unit of affiliation of in the cluster, R_j is the d-dimensional measured information, C_j is the d-dimension centre of the cluster.

Fuzzy classification is finished by objective function iteration as appeared above, with the update of membership a and the cluster centres C_j.

Reiteration of this process will be ended when there is end condition somewhere around 0 and 1, whereas K is the iteration steps and N is the total number of clusters.

This KFCM process considers the input as different resource and output as clustered document based on the resource type.

4.2.1.1 Procedure

1. KFCM algorithm and its objective function are given bellow:

$$OB_m = \sum_{i=1}^{n} \sum_{j=1}^{c} A_{ij}^m (1 - K(R_j - C_i)) \tag{2}$$

2. Compute the fuzzy centers C_j

$$C_j = \frac{\sum_{j=1}^{n} A_{ij}^m K(R_j, C_i) R_j}{\sum_{j=1}^{n} A_{ij}^m K(R_j, C_i)}; \quad i = 1, 2, \ldots, C \tag{3}$$

3. Calculate the fuzzy membership function A_{ij} using

$$A_{ij} = \frac{(1 - K(R_j, C_i))^{-1/(m-1)}}{\sum_{j=1}^{n} (1 - K(R_j, C_i))^{-1/(m-1)}}; \quad i = 1, 2, \ldots, C \tag{4}$$

4. Repeat steps 2 and 3 until the maximum of OB_m value is achieved

Identify the essential conditions for minimizing OB_m is revise equations (3) and (4) only when the kernel function K is selected to be the Gaussian function with

$$K(R_j, C_i) = \exp\left(-\|R_j - C_i\|^2/\sigma^2\right)$$

By replacing the Euclidean distance for dissimilar conditions various kernels can be selected. Nevertheless, a Gaussian kernel is suitable for clustering in which it can principally mark the essential conditions. Finally all the resources are grouped with their resource type. Then we allocate the resource for the executed jobs or task. Here the tasks are allocated to the corresponding resource with minimum cost; so the proposed method is to select the optimal resource with minimum cost using optimal cloud resource provisioning algorithm using particle swarm optimization algorithm. The detail explanation of modified cloud resource provisioning algorithm using PSO algorithm is illustrated in the below section.

4.3 Modified cloud resource provisioning using PSO

The proposed technique use changed cloud resource provisioning algorithm to choose the ideal resource with least cost. To choose the ideal resource with least cost the proposed cloud resource provisioning is utilized molecule swarm enhancement algorithm. Here the wellness capacities as OCRP [23]. The target capacity of the proposed strategy is selecting the ideal resource utilizing ideal cloud resource provisioning algorithm. Molecule swarm enhancement is a populace based improvement algorithm modeled. The algorithm of PSO is instated with a group of discretionary particles and next hunt down optima by overhauling eras. Each of the particles are flown through the hunt space encasing its position adjusted taking into account its separation from its very own best position and the separation from the best molecule of the swarm. The presentation of every molecule, i.e. how shut the particles is from the global optimum, is ascertained by method for a wellness capacity which depends on the enhancement issue. As indicated by PSO, there are two different sorts of forms are utilized. The first is "singular best" and the second is "global best". It is the individual best determination algorithm by assessing every individual position of the molecule to its own particular best position. It is the global best choice algorithm, which gets the global information by making the development of the particles contains the position of the best molecule from the entire swarm. Every molecule i, flies in a n-dimensional inquiry space.
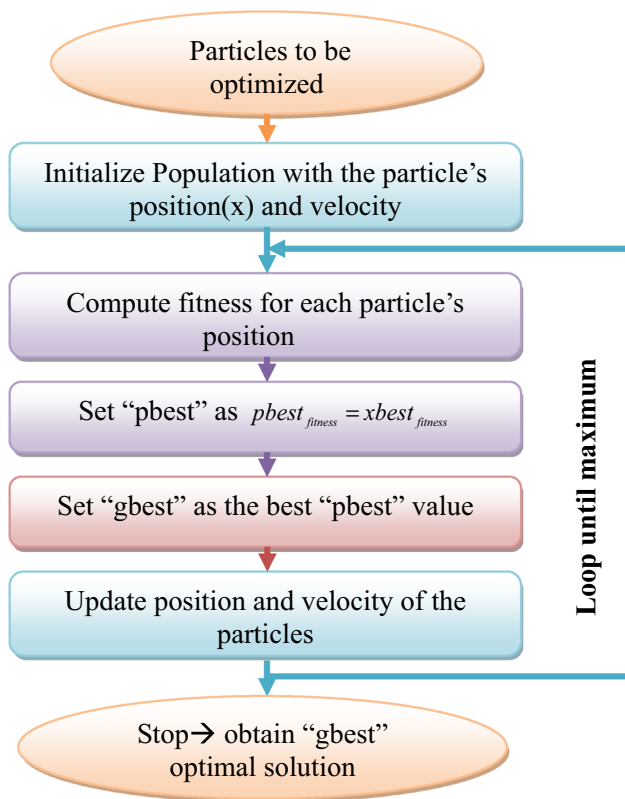
Fig. 2 Steps for PSO

In Fig. 2, steps to obtain optimal solution—cp_i vector: It symbolizes the current position of the i th particle in the search space, lb_i vector: It points out the location of the best solution of the i th particle in the search space, and cv_i vector: It points out the direction for which the particle i will travel.

The personal best position of a particle is indicated as *pbest*. The particles in the search space have its co-ordinates that are associated with the fitness as the best solution. The value of fitness is accumulated in it. It is the *pbest* value. The *pbest* value of a particle i is the best position that the particle has perceived so far. If *fitness fitness* indicates the fitness function means, then the *pbest* value of the particle i is revises as follows.

$$pbest = \begin{cases} lb_i = cp_i \\ lb_{fitness} = cp_{fitness} \end{cases} \quad if\ cp_{fitness} > lb_{fitness} \quad (5)$$

where the fitness value is calculated as follows,

$$fitness = OCRP \quad (6)$$

One more location attained by any of the particle in the population is the *gbest* value, which is the universal version of the PSO and is the complete best value. The global best position of the whole swarm is represented as.

$$gbest = best(pbest) \quad (7)$$

4.3.1 Steps to find optimal solution

Step 1 Initialize a populace of i particles with each speck's position cp_i and velocity cv_i on a problematic space of dimension n.
Step 2 Fitness function to be calculated for each speck.
Step 3 Make assessment among the particle's fitness value, $cp_{fitness}$ and speck's *pbest* fitness value $lb_{fitness}$. If the current fitness value of speck is better than the speck's *pbest* fitness value, then set the *pbest* value into current position.
Step 4 Inspect all of the speck's *pbest* fitness value, $lb_{fitness}$ with value of *gbest*. If the current value, *pbest* is better than the *gbest* value means, then set the *gbest* value into current speck's array index and value.
Step 5 Revise the velocity and position of the specks specified as in Eqs. (8) and (9)

$$cv_i^{new} = cv_i + \varphi_1 \cdot r_1 \cdot (pbest_i - cp_i)$$
$$+ \varphi_2 \cdot r_2 \cdot (gbest_i - cp_i) \quad (8)$$
$$cp_i = cp_i + cv_i^{new}, \quad (9)$$

where i is the speck. φ_1, φ_2 is the learning rates governing the speck towards to its best position. r_1, r_2 is the random numbers that are uniformly distributed in the range [0, 1]. cv_i indicates the current velocity.
Step 6 Repeat step 2, until a better fitness or maximum number of iterations are met.

The optimized resource with minimum cost attained from this PSO algorithm in modified cloud resource provisioning algorithm. As a result, we can allocate the jobs to the corresponding resource type with minimum cost using optimal cloud resource provisioning method. The performance of the proposed cloud resource provisioning algorithm is evaluated and compared with various existing method. It is described in further Sect. 5.

5 Results and discussion

The result analysis in resource provisioning and allocation process based on KFCM and MCRP is performed in the working platform of JAVA with CloudSim. Different performance evaluation measures are considered to evaluate the performance of the proposed technique in resource provisioning and allocation process.

5.1 Performance analysis

The performance of the proposed method is evaluated based on execution time, cost and memory taken for execution process by varying the number of tasks in proposed approaches in contrasted with that of the current method. The proposed MCRP technique is compared with existing OCRP [19] in terms of execution time, cost value and memory usage. The graphical representation of comparative analysis is shown in Figs. 3, 4 and 5. In Fig. 3 the execution time of MCRP is less than OCRP by varying the number of tasks. Figure 4 shows the cost value of MCRP and OCRP . It is observed from Fig. 4 that MCRP achieves minimum cost when com-
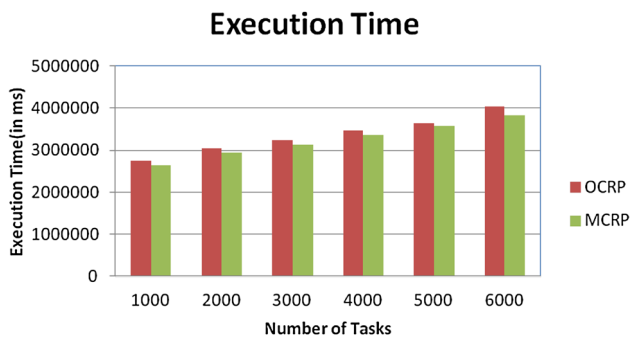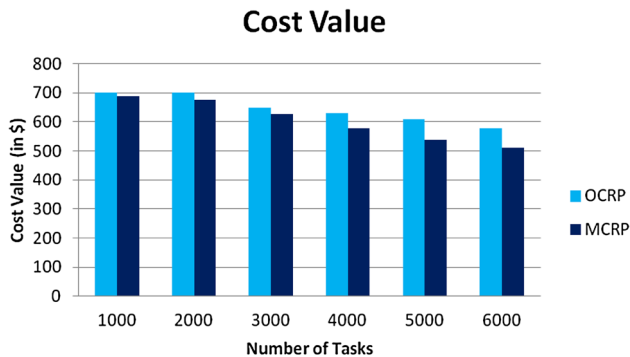

Fig. 3 Execution time evaluation

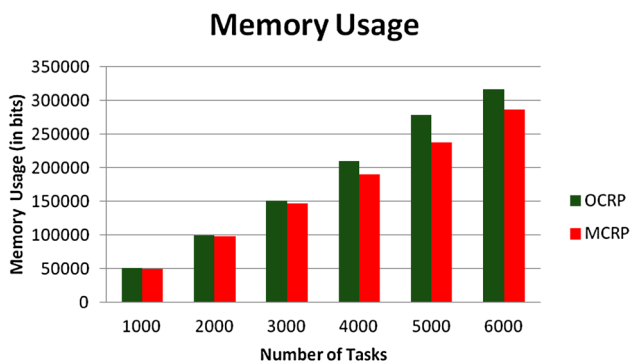**Fig. 4** Cost value evaluation

Fig. 5 Memory usage evaluation

pared with OCRP. Figure 5 represents the memory usage of number of tasks for both MCRP and OCRP. The MCRP technique utilizes least memory for varying number of tasks when compared with OCRP.

From Figs. 3, 4 and 5 it is observed that for all the system loads in terms of number of tasks given, MCRP gives a lesser execution time, cost value and memory usage. The MCRP has an average improvement of 20.62, 40.14 and 39.32% over OCRP algorithm in standings of execution time, cost value and memory usage respectively.

6 Conclusion

In this paper, resource provisioning and resource allocation techniques were proposed. Here KFCM algorithm was used to cluster the available resource. Then the tasks were allocated to the available resource using modified cloud resource provisioning algorithm (MCRP). In MCRP algorithm PSO algorithm were used to select the optimal cost. Based on the optimal result we allocate the resource. The performance of the proposed work was evaluated by cost, execution time and memory usage. From the results it is established that the proposed MCRP method achieve minimum execution time, minimum cost value and low memory usage than the existing technique.

References

1. Jewarg, P.B., Patil, J.: Payment minimization and error-tolerant resource allocation for cloud system using equally spread current execution load. Int. J. Adv. Res. Comput. Eng. Technol. (IJARCET) **3**(8), 2669–2676 (2014)
2. Pawar, C.S., Wagh, R.B.: Priority based dynamic resource allocation in cloud computing. In: Proceedings of International Symposium on Cloud and Services Computing, pp. 1–6 (2012)
3. Saranya, S., Saranya, N.: An efficient resource allocation for improving resource utilization in self organizing clouds. Int. J. Innov. Res. Comput. Commun. Eng. **2**(1), 1648–1651 (2014)
4. Sivaranjani, V., Jayamala, R.: Optimization of workload prediction based on map Reduce frame work in a cloud system. Int. J. Res. Eng. Technol. **3**(3), 264–266 (2014)
5. Pandya, P.P., Bheda, H.A.: Dynamic resource allocation techniques in cloud computing. Int. J. Adv. Res. Comput. Sci. Manag. Stud. **2**(1), 559–563 (2014)
6. Al-Sharif, Z.A., Jararweh, Y.: Al-Dahoud, A: ACCRS: autonomic based cloud computing resource scaling. Clust. Comput. **20**(3), 2479–2488 (2017)
7. Dhivya, L., Padmaveni, MsK: Dynamic resource allocation using virtual machines for cloud computing environment. IJREAT Int. J. Res. Eng. Adv. Technol. **2**(1), 1–4 (2014)
8. Goudarzi, H., Pedram, M.: Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems. In: Proceeding of IEEE 4th International Conference on Cloud Computing, pp. 324–331 (2011)
9. He, C., Lu, Y., Swanson, D.: Matchmaking: a new MapReduce scheduling technique. In: Proceeding of Third IEEE International

Conference on Cloud Computing Technology and Science, pp. 40–47 (2011)

10. Ren, H., Lan, Y., Yin, C.: The load balancing algorithm in cloud computing environment. In: 2nd International Conference on Computer Science and Network Technology, pp. 925–928 (2012)

11. Zhou, Z., Liu, F., Xu, Y., Zou, R., Xu, H., Lui, J.C.S., Jin, H.: Carbon-aware load balancing for geo-distributed cloud services. In: Proceeding of IEEE 21st International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems (2013)

12. Dou, H., Qi, Y., Wang, P.: Hybrid power control and electricity cost management for distributed internet data centers in cloud computing. In: Proceeding of 10th Web Information System and Application Conference, pp. 394–399 (2013)

13. Aljohani, A.M., Holton, D.R.W., Awan, I., Alanazi, J.S.: Performance evaluation of local and cloud deployment of web clusters. In: Proceeding of International Conference on Network-Based Information Systems, pp. 244–248 (2011)

14. Toosi, A.N., Calheiros, R.N., Thulasiram, P.K., Buyya, R.: Resource provisioning policies to increase IaaS provider's profit in a federated cloud environment. In: Proceedings of IEEE International Conference on High Performance Computing and Communications, Banff, Canada, pp. 279–287 (2011)

15. Zaman, S., Grosu, D.: A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds. IEEE Trans. Cloud Comput. **1**(2), 129–141 (2013)

16. Di, S., Wang, C.-L.: Dynamic optimization of multi attribute resource allocation in self-organizing clouds. IEEE Trans. Parallel Distrib. Syst. **24**(3), 464–478 (2013)

17. Li, S., Zhou, Y., Jiao, L., Yan, X., Wang, Xin, Lyu, Michael Rung-Tsong: Towards operational cost minimization in hybrid clouds for dynamic resource provisioning with delay-aware optimization. IEEE Trans. Serv. Comput. **8**(3), 398–409 (2015)

18. Zhu, Q., Agrawal, G.: Resource provisioning with budget constraints for adaptive applications in cloud environments. IEEE Trans. Serv. Comput. **5**(4), 497–511 (2012)

19. Chunlin, L., Jianhang, T., Youlong, L.: Distributed QoS-aware scheduling optimization for resource-intensive mobile application in hybrid cloud. Clust. Comput. **20**(1), 1–18 (2017)

20. Li, W., Zhang, Q., Wu, J., Li, J., Hao, H.: Trust-driven and QoS demand clustering analysis based cloud workflow scheduling strategies. Clust. Comput. **17**(3), 1–18 (2014)

21. Byun, E.-K., Kee, Y.-S., Kim, J.-S., Maeng, S.: Cost optimized provisioning of elastic resources for application workflows. Future Gen. Comput. Syst. **27**(8), 1011–1026 (2011)

22. Iqbal, W., Dailey, M.N., Carrera, D., Janecek, P.: Adaptive resource provisioning for read intensive multi-tier applications in the cloud. Future Gen. Comput. Syst. **27**(6), 871–879 (2011)

23. Phani Praveen, S., Tulasi, U., Vishnu, B., Yuvakrishna, A.: A new approach for optimizing resource provisioning in cloud computing using OCRP algorithm. Int. J. Comput. Sci. Technol. **1**(8), 15–21 (2013)

A. Suresh B.E., M.Tech., Ph.D. works as the Professor & Head, Department of the Computer Science and Engineering in Nehru Institute of Engineering & Technology, Coimbatore, TamilNadu, India. He has been nearly two decades of experience in teaching and his areas of specializations are Data Mining, Artificial Intelligence, Image Processing, Multimedia and System Software. He has published 30 papers in International journals. He has published more than 40 papers in National and International Conferences. He has served as a reviewer for Springer, Elsevier, and Inderscience journals. He is a member of ISTE, IACSIT, IAENG, MCSTA, MCSI, and Global Member of Internet Society (ISOC). He has organized several National Workshop, Conferences and Technical Events. He is regularly invited to deliver lectures in various programmes for imparting skills in research methodology to students and research scholars. He has published two books, one in the name of Data structures & Algorithms in DD Publications, Chennai and other one in the name of Computer Programming in Excel Publications, Chennai.

R. Varatharajan received his B.E., M.E. and Ph.D. degrees all in Electronics and Communication Engineering from Anna University and Bharath University, India. His main area of research activity is Medical Image processing, Wireless Networks and VLSI Physical Design. He has served as a reviewer for Springer, Inderscience and Elsevier journals. He has published many research articles in refereed journals. He is a member of IEEE, IACSIT, IAENG, SCIEI and ISTE wireless research group. He has been serving as Organizing Chair and Program Chair of several International conferences and in the Program Committees of several International conferences. Currently he is working as a Associate professor in the Department of Electronics and Communication Engineering at Sri Lakshmi Ammaal Engineering Engineering College, Chennai, India.