Research Article

QRAS: efficient resource allocation for task scheduling in cloud computing

Harvinder Singh[1] · Anshu Bhasin[1] · Parag Ravikant Kaveri[2]

Received: 6 February 2020 / Accepted: 9 March 2021 / Published online: 19 March 2021

Abstract
Cloud resource allocation, a real-time problem can be dealt with efficaciously to reduce execution cost and improve resource utilization. Resource usability can fulfill customers' expectations if the allocation has performed according to demand constraint. Task Scheduling is NP-hard problem where unsuitable matching leads to performance degradation and violation of service level agreement (SLA). In this research paper, the workflow scheduling problem has been conducted with objective of higher exploitation of resources. To overcome scheduling optimization problem, the proposed QoS based resource allocation and scheduling has used swarm-based ant colony optimization provide more predictable results. The experimentation of proposed algorithms has been done in a simulated cloud environment. Further, the results of the proposed algorithm have been compared with other policies, it performed better in terms of Quality of Service parameters.

Keywords Cloud computing · Resource allocation · Scheduling · Utilization

1 Introduction

Cloud computing is a Service Oriented Architecture (SOA), providing service in form of infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) [1]. Users can access cloud resources using network services under the pay-per-use model. Thus, providers are doing efforts to optimize resources according to customers requirements whilst managing the Service Level Agreement (SLA) violation. The growing demand of cloud services make difficult to handle the unexpected workload. Resource allocation accordingly to workload is challenging that impacts on Quality of Service (QoS) parameters. Therefore, it needs an accurate policy that can assist in scalability and elasticity. Thus, appropriate workload distribution to resource will also helps in availability and utilization.

The existing literature survey shows that cloud resource management problem has become difficult due to increase in demand of cloud services [2]. Hence, the study in this research area motivated to develop a novel technique for optimal matching and throughput utilization of resources. Further, the resource allocation policy should able to identify tasks exact requirements before placing on virtual machine (VM). Such a scheme will be managed resource availability, facilitate to avoid SLA violations, and deadline of tasks [3]. Thus, the development of a standard technique can overcome the scheduling problem in cloud computing.

We have investigated various research works related to resource management from the perspective of QoS. To meet suitable resource, the proposal comes with the following systematic order: (1). demand identification criteria (2) generating initial matchings (3) matching criteria meet the QoS requirements (4) selection of suitable solutions (5)

✉ Harvinder Singh, harvinder9815653260@gmail.com; Parag Ravikant Kaveri, parag.kaveri@sicsr.ac.in | [1]IKGPTU Kapurthala, Kapurthala, Punjab, India. [2]Symbiosis Institute of Computer Studies and Research, Symbiosis International (Deemed University), Pune, India.

SN Applied Sciences (2021) 3:474 | https://doi.org/10.1007/s42452-021-04489-5

resource usability by proper distribution (6) performance analysis and SLA management. To address the above-discussed research problem, the proposed QoS based Resource Allocation and Scheduling (QRAS) used ant colony optimization [4, 5] for generating and evaluating the relevant solution. This work is an extention of our existing research papers [6–8]. The allocation has been performed based on optimal solutions generated by swarm agents who become intelligent from the environment.

This research work motivated from resource allocation and scheduling problem in cloud environment to workload automatically. VM technology inspired to work on autonomic resource allocation based on different tasks requirements. Therefore, cost-aware VM scheduling is difficult process whilst dealing with dynamic and heterogeneous environment. Further, optimal usage of resource energy was also challenging work. Consequently, efficient task execution requires an expected execution cost (EEC) with specific VM configurations. The proposed framework aims to reduce the energy consumption and to exploite the cloud resources by optimizating EEC. The research paper has been structured in sub-sections as follows: Sect. 2, presents a literature survey on contemporary research problems in resource allocation and workflow scheduling; Sect. 3, proposed resource allocation and scheduling framework; Sect. 4, presents experimentation setup, Sect. 5, results, and analysis; Sect. 6, presents conclusion and future work.

2 Related work

Masoud et al. [9] presented an energy-efficient method for resource allocation by optimizing communication latencies and consolidate allocation decisions for an appropriate resource to the best applicant, it has improved performance and reduced energy consumption. Fernando et al. [10] presented an optimization method for resource allocation, particularly for educational purposes. It has performed dynamic workload allocation and reduced cost [10]. Singh et al. [11] presented BULLET, a resource provisioing technique whilst adopting optimization mechanism for appropriate mapping on available resources. The algorithm usage K-means clustering algorithm and PSO for workload identification and grouping, it assigns suitable resource, reduced cost, time, and energy [11]. Zhang et al. [12] presented a VM allocation method, it adopted resource reservation-based scheme that efficiently performed VM-to-PM mapping. It performed higher resource utilization and energy saving [12]. Allah et al. [13] presented resource allocation scheme using hybridizing genetic and cuckoo search algorithms to perform cost-effective results. The suitable solution identified through a rigorous process using cuckoo algorithm and genetic operators [13]. Kholidy [14] presented a "Swarm Intelligence Based Prediction Approach" (SIBPA) for managing cloud resources accuracy and response time. It improved throughput utilization of CPU, memory, disk storage, and dynamic behavior of consumer request [14]. Than et al. (2020) proposed an energy-aware resource allocation algorithm that has been efficiently managed power consumption and workload management. It evaluates the energy usage in resource allocation, and reduces power consumption on data centers [15]. Bhattacherjee et al. [16] proposed VM placement and migration policy using a dynamic thresholding approach. Markov chain has been used for predicting future requirements based on history. Its reduces energy consumption in data center resources [16]. Mansouri et al. [17] proposes cost-based job scheduling (CJS) strategy for optimizing resource usage. It identified computation and communication cost, and respond accordingly in reasonable amount of time and cost [17]. Reshmi et al. [18] presented profit and resource availability aware methodd that managed complexity and accuracy. It performed grouping of similar tasks, and assign an optimal resource using a hybrid cuckoo search algorithm [18]. Tripathi et al. [19] proposed the "Modified Dragonfly Algorithm" for optimising virtual machine placement to manage resource distribution problem. Due to dynamic VM placement, it shows better resource utilization [19].

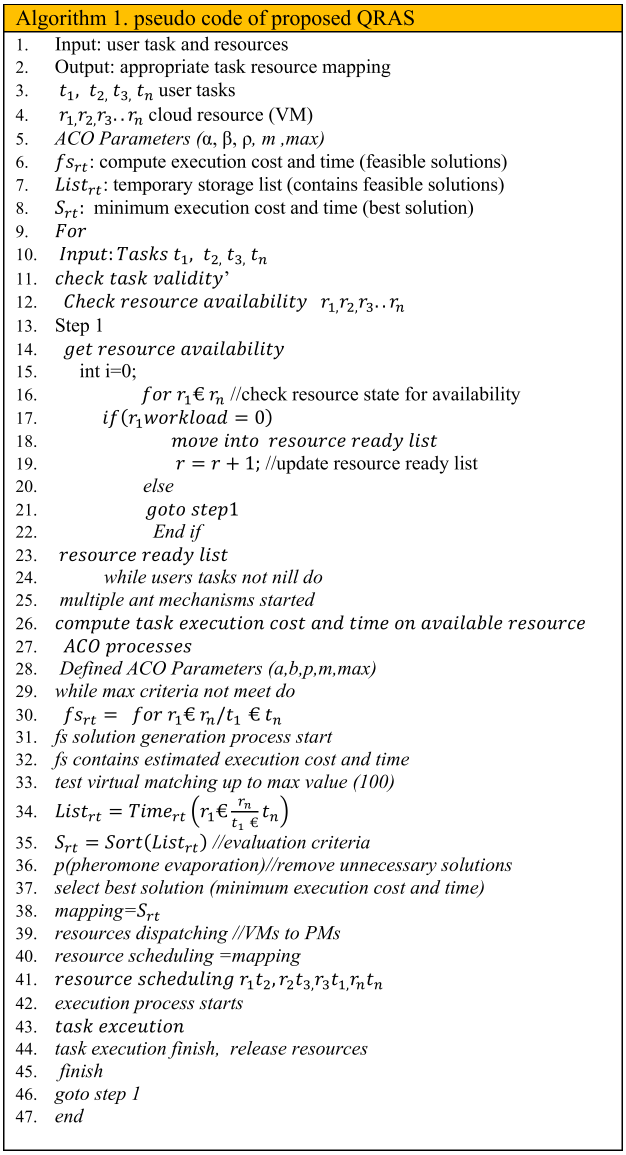
2.1 Contribution of the research work

The major contribution of this research work is as follows: optimal resource allocation and scheduling, maximize the availability of VMs. Further, dynamic workload identification for appropriate allocation of resources according to customer demand. Therefore, dynamic workload allocation and scheduling needs optimal matching. Moreover, for measuring cloud resource utilization, and calculation of cost consumption was a difficult process. Thus, design of objective function targets to evaluate cost consumption and resource utilization. The optimization metrics for cost and utilization calculation are standards [4, 5] which are also useful in calculating makespan and energy consumption. Therefore, defined objective function fulfills the expectations of proposed work by meeting QoS parameters.

3 Proposed resource allocation algorithm

In this section, the proposed QoS based Resource Allocation and Scheduling (QRAS) algorithm presented in methodical manner. Swarm based ant colony optimization (ACO) technique has been used for resource allocation and scheduling problem. The optimal matching has been underlined through negotiation process. The QRAS permed in systematic manner: (1) identify task requirement; (2) multiple ant mechanism starts based on resources availability (3) negotiation process initiated; (4) feasible solutions generated (5) evaluation of solutions based on QoS validation (6) design of mapping based on solutions (7) resource allocation and scheduling followed by mapping (8) task execution and release of resources performed. The pseudo-code of the proposed algorithm has been given in Fig. 1.

The proposed algorithm performed efficient workload scheduling based on optimal solutions. These solutions have been generated by foraging behavior of ant's. The availability of VMs help in generating fast initial solutions. The placement on VMs has been done based on most appropriate result.



```
Algorithm 1. pseudo code of proposed QRAS
Input: user task and resources
Output: appropriate task resource mapping
 $t_1,\ t_{2,}\ t_{3,}\ t_n$ user tasks
 $r_{1,}r_{2,}r_3..r_n$ cloud resource (VM)
ACO Parameters (α, β, ρ, m ,max)
$fs_{rt}$: compute execution cost and time (feasible solutions)
$List_{rt}$: temporary storage list (contains feasible solutions)
$S_{rt}$:  minimum execution cost and time (best solution)
For
 $Input: Tasks\ t_1,\ t_{2,}\ t_{3,}\ t_n$
check task validity'
  $Check\ resource\ availability\ \ r_{1,}r_{2,}r_3..r_n$
Step 1
  get resource availability
     int i=0;
                 $for\ r_1€\ r_n$ //check resource state for availability
         $if(r_1 workload = 0)$
                        move into  resource ready list
                         $r = r + 1$; //update resource ready list
                 else
                  goto step1
                   End if
 resource ready list
          while users tasks not nill do
 multiple ant mechanisms started
compute task execution cost and time on available resource
  ACO processes
 Defined ACO Parameters (a,b,p,m,max)
while max criteria not meet do
  $fs_{rt} =\ \ for\ r_1€\ r_n/t_1\ €\ t_n$
fs solution generation process start
fs contains estimated execution cost and time
test virtual matching up to max value (100)
$List_{rt} = Time_{rt}\left(r_1€\frac{r_n}{t_1\ €}t_n\right)$
$S_{rt} = Sort(List_{rt})$ //evaluation criteria
p(pheromone evaporation)//remove unnecessary solutions
select best solution (minimum execution cost and time)
mapping=$S_{rt}$
resources dispatching //VMs to PMs
resource scheduling =mapping
$resource\ scheduling\ r_1t_2, r_2t_{3,}r_3t_{1,}r_nt_n$
execution process starts
task exceution
task execution finish,  release resources
 finish
goto step 1
end
```

Fig. 1 The pseudo-code of ACO proposed a multi-swarm algorithm

The proposed QRAS code works as follows: The algorithm process starts with the input of users' tasks. Then, the system analyzer validate tasks whilst endorsing them to competency list. Resource availability starts checking of idle/free resources and put them into availability list. As number of resources are available in availability list, multiple ant colony systems initiated negotiation process. It generates initial solution by looking at all possible results. The process continue generating solutions until termination criteria. These solutions are stored in temporary list for evaluation. Further, QoS based assessment identify reasonable solution, others sent to evaporation process.

These standardize solutions have a capacity to perform task execution on affordable cost and time. The mapping performed based on optimal solution, generated by the system. Further, VM placement has performed according to virtual mapping. It defines an optimal scheduling by best result from all solutions. Then, the task execution process has started, optimal workload assigned to resources. As task execution over, resources released to availability. This process will continue until users' tasks, and the process goes at step 1.

3.1 Cost ($execution_{cost}$)

Calculation of cost of VM usage for task execution. The cost of execution measured using Eqs. (1–4) defined as; CS_{tt} denotes the cost of task execution where T_i on VMs represented by Eq. 1. CS_{vm} represents (Eq. 2) execution cost of tasks on each VM. CS_{ttvm} (Eq. 3) calculates the cost of tasks execution on VMs within data center. Equation 4 depicts the optimal cost of tasks execution on available resources. Thus, objective function fulfill the needs of porviders and consumers requirements by reducing cost and energy consumption.

To measure the execution cost of each task

$$CS_{tt} = \sum_{i=1}^{tn} \mathrm{CS_i} \tag{1}$$

where *tn* represent numerous of tasks.

To measure the execution of tasks on each VM

$$CS_s = \sum_{i=1}^{len} \mathrm{CS_i} \tag{2}$$

where len represents task size.

To measure the total execution cost of tasks on all VMs

$$CS_{ttvm} = \sum_{i=1}^{vn} \mathrm{C}_{\mathrm{vm}_i} \tag{3}$$

where *vn* maximum numerous of VMs.

$$\mathrm{execution}_{\mathrm{cost}} = \min\left(\mathrm{cs}\left(\mathrm{r}_1,\ \mathrm{t}_1\right)\right) \quad \text{for} \quad 1 < \mathrm{r} < \mathrm{vn} \quad \text{and} \quad 1 < \mathrm{t} < \mathrm{tn}. \tag{4}$$

3.2 Resource utilization (utilization$_{\mathbf{vm}}$)

Resource utilization is a pivotal and unavoidable factor whilst measuring resource usage for task execution. It also helps in carrying out the QoS requirements of customers and providers. To measure the utilization of VMs, Eqs. 5–7 have been defined as follows; U_{tot} (Eq. 5) represents the utilization calculation of VMs usage for executing each task on VM. U_{vm} (Eq. 6) computes the utilization of each VM for executing all task *tn*. U_{totvm} (Eq. 7) totals the utilization of VMs for executing number of tasks T_n.

Calculation of utilization for each task

$$U_{tot} = \sum_{i=1}^{tn} \mathrm{t}_{\mathrm{i}} \tag{5}$$

tn reprents numbers of tasks submitted by users

$$U_{\mathrm{vm}} = \sum_{i=1}^{len} \mathrm{t}_{\mathrm{i}} \tag{6}$$

len represents size of tasks for executing users' workload on VM

$$U_{totvm} = \sum_{i=1}^{m} \mathrm{Tvm}_{\mathrm{i}} \tag{7}$$

Resource utilization has calculated VMs (MIPS) usage for task execution (length). The presented QRAS has performed allocation according to customer demand and resource availability. User request cloud server for information retreiving in form of tasks $\left(T_{i1}, T_{i2,}T_{i3,}T_{in}\right)$; negotiation process initiated, and the race of optimal resource finding put alternative solutions. ACO parameters $(for\ VM_{j1}€\ VM_{jm}/T_{i1}\ €\ T_{in})$; impact on solutions QoS validation pick accurate matches, a cost-effective matching has found ($T_{i1}VM_{j1,}T_{i2}VM_{j2,}T_{i3}VM_{j3,\ldots}T_{in}VM_{jm}$) and final stage put predicted resource utilization.

The research work has followed swarm intelligence for optimizing scheduling problem. Its agents are very sensible for solving the Travelling Salesman Problem (TSP) whilst exploiting resources and reduction in energy consumption [20]. Resource scheduling in cloud is NP-hard problem, and swarm has the capabilities to drive such complex environment. However, considering a suitable match according to demand is difficult process. The proposed resource allocation algorithm follows an ant mechanism for dealing with the problem. The sense of agents becoming intelligent from environment make capable to manage complex state. To control them and act accordingly ACO parameters considered best tool for operating.

Here, it has measured the estimation of task execution cost and time. VMs availability based simultaneously multiple solution generation processes has adopted. Before forwarding these solutions are evaluation done under specific criteria. At last, merely reasonable answers have remained, known as optimal matching whilst other solutions no need to stay. These shortlisted solutions projected by the objectives of work whereas appropriate scheduling demand such cost-effective matching.

4 Experimentation setup

To prove the accuracy of proposed QRAS, testing of experimentation has been performed. Due to the limitations of accessibility of real cloud environment the alternative choice of simulation has been opted in experimenation. The QRAS algorithm has been implemented on Workflowsim, a cloud environment based simulation tool [21]. It provides the features of real cloud environment such as data centers, VMs, users, cloudlets and bandwidth, etc. for configuring and customizing. To setup the simulation environment Workflowsim requirements NetBeans IDE 8.0, JDK 1.7 and libraries installed on the system.

In this experimentation the montage dataset considered as a users' workload [22]. It was an appropriate problem for simulation environment due to its distributed environment nature. The architecture of this dataset problem is communication oriented. Thus, the control flow of dependent tasks depend on the output of parents solutions. Furthermore, complexity of dataset is also impacting the workload execution. The experimentation specifications like data center, VMs, and users' demand defined in Table1.

The ACO parameters have described as: α indirect control of pheromone value; β indirect control for moving next; ρ denotes evaporation rate of pheromone; *m* represents maximum number of ants in the colony; *max* indicates maximum number of iterations for generating feasible solutions. The parameter's value has an influence on results and performance. Therefore, parameter selection is also as an important factor which accounts important consideration. The results have been generated using

Table 1 Experimentation configurations

Data center setup	
Data center	01
Number of hosts	01
VMs configuration	
Number of VMs	5–50
MIPS	1000
Nuner of PEs per VM	1
VM memory	512
Bandwidth	1000
Workload environment	
Number of users	15
Data Set	Montage
Numbers of cloudlets	1000
ACO parameters setup	
Number of iterations (max)	100
Number of ants (m)	10
α	.3
β	1
ρ	.4

different set of VMs and cloudlets in the perspective of QoS parameters. Further, QRAS results have compared with MINMIN, MCT and ACO basedon varied number of VMs, and cloudlets in terms of QoS parameters.

5 Results and analysis

The results have been generated from the perspective of cost and utilization; the R^2 value calculated by an exponential curve. The R^2 value indicates 1 shows exponential growth and 0 shows an exponential decrease. The graphical results have described execution cost of four different resource allocation policies and compared in terms of R^2 value. The results have been depicted by Fig. 2 (execution cost for 100 tasks), Fig. 3 (execution cost for 1000 tasks), and Fig. 4 (resource utilization for 1000 tasks) show different values of execution cost and resource utilization. Table 2 shows the R^2 for the execution cost of 100 tasks, Table 3 represents the execution cost of R^2 value for 1000 tasks.

The results have been compared using statistical analysis. The performance of presented QRAS compared with existing MINMIN, MCT, and ACO algorithms. The execution cost of tasks calculated using Eqs. (1–4), and resource utilization measured based on Eqs. (5–7). The cost denotes the execution cost, time and cost task remain under execution process. Moreover, utilization shows the VMs actively used in execution process.

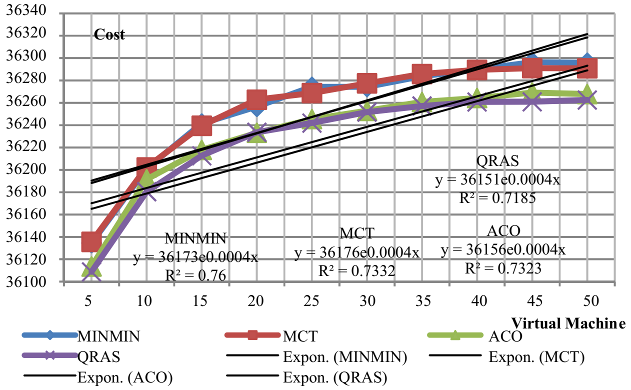


Fig. 2 Execution cost for 1000 cloudlets on VMs 5–50

Table 2 R^2 value for montage cost calculation (100 cloudlets)

	MINMIN	MCT	ACO	QRAS
R^2	0.689	0.549	0.598	0.461

Table 3 R^2 value for montage cost calculation (1000 cloudlets)

	MINMIN	MCT	ACO	QRAS
R^2	0.760	0.733	0.732	0.718

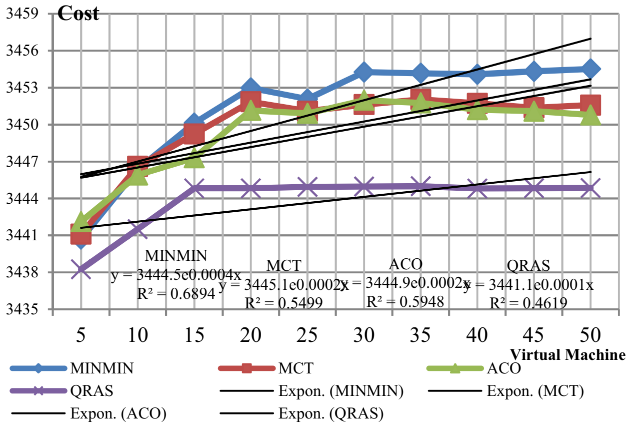


Fig. 3 Execution cost for 1000 cloudlets on VMs 5–50

5.1 Observations for cost for 100 tasks

To evaluate the execution cost for 100 tasks on VMs 5–50 results have been compared with existing algorithms, shows in Fig. 2. The performance in results proved, 33.09%, 16.03%, 22.39% outperformed in terms of cost compared w.r.t MinMin, MCT, and ACO respectively.

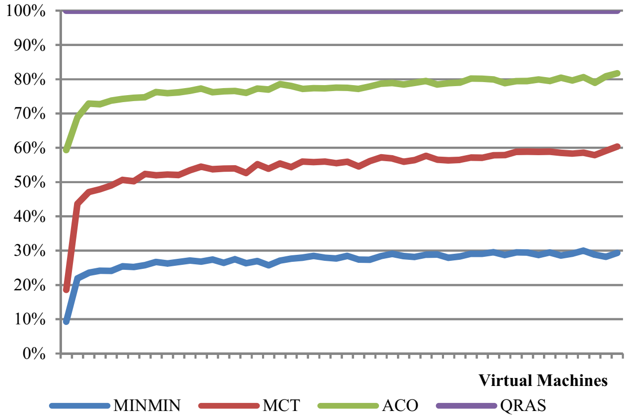


Fig. 4 Comparison of resource utilization (1000 cloudlets on VM = 50)

5.2 Observations for cost for 1000 tasks

To evaluate the execution cost for 1000 tasks on VMs 5–50, comparison of results shows in Fig. 3. The reduction of cost 15%, 2.75%, 2.09% proved the proposed algorithm performed better than MinMin, MCT, ACO respectively.

5.3 Observations for utilization for 1000 tasks on 50 VMs

Figure 4 shows the utilization of VM resources, it has been measured on set of 50 VMs. Resource utilization experimentation performed based on VM usage. The analysis of result shows, the proposed technqiue scaled up resource utilization. Thus, QRAS algorithm considedred as higher utilization of resources compared to other existing algorithms.

The following observations have been concluded by evaluating of VMs utilization:

MinMin performed resource utilization as 10–30% w.r.t. proposed QRAS.

MCT shows to perform resource utilization as 20–58% w.r.t. proposed QRAS. Simple ACO performed resource utilization as 60–62% w.r.t. proposed QRAS.

The results have shown the QRAS algorithm performed better in terms of utilization and energy saving.

6 Conclusion and future work

Resource allocation in cloud environment is challenging due to users' dynamic requirements. It has also complex process and impacts cost and resource utilization. This research paper proposed a novel resource allocation technique for QoS based performance. ACO technique used for identifying an optimal solution to reduce the execution cost and improve resource utilization. Therefore, it proves in identification of task to resource suitability. Furthermore, results and analysis shows proposed algorithm outperformed against traditional algorithms.

In addition, the optimization mechanism can be adopted in optimal usage of advanced cloud computing components such as internet of things. The future work in cloud computing resource management would be more concentrating on renewable energy, automative of resource management and reading consumer behavior which accelerate the cloud services.

Declarations

Conflict of interest The authors have no conflict of interest regarding publication of this manuscript. This submission is original work has and not under review at any other journal/ conference etc.

References

1. Choudhary V, Vithayathil J (2015) The impact of cloud computing: should the IT department be organized as a cost center or a profit center? Emerald J Manag Inf Syst 30(2):67–100. https://doi.org/10.2753/MIS0742-1222300203
2. Harvinder S, Anshu B, Kaveri PR, Vinay C (2020) Cloud resource management: comparative analysis and research issues. Int J Sci Technol Res 9(6):96–113
3. Singh S, Chana I, Buyya R (2017) STAR: SLA-aware autonomic management of cloud resources. IEEE Trans Cloud Comput. https://doi.org/10.1109/TCC.2017.2648788
4. Qiang G (2017) Task scheduling based on ant colony optimization in cloud environment. Proc AIP Conf 1834(1):040039. https://doi.org/10.1063/1.4981635
5. Kun L, Gaochao X, Guangyu Z, Yushuang D, Dan W (2011) Cloud task scheduling based on load balancing ant colony optimization. In: Proceedings of sixth annual chinagrid conference. https://doi.org/10.1109/ChinaGrid.2011.17
6. Harvinder S, Anshu B (2017) Efficient resource management technique for performance improvement in cloud computing. Indian J Comput Sci Eng 8(1):33–39
7. Harvinder S, Anshu B, Ravikant KP (2019) QoS based efficient resource allocation and scheduling in cloud computing. Int J

Technol Hum Interact (IJTHI) 15(4):13–29. https://doi.org/10.4018/IJTHI.2019100102

8. Harvinder S, Anshu B, Ravikant KP (2019) SECURE: efficient resource scheduling by swarm in cloud computing. J Discrete Math Sci Cryptogr 22(2):127–137. https://doi.org/10.1080/09720529.2019
9. Masoud N, Ronak K (2016) Energy-efficient and latency optimized media resource allocation. Emerald Int J Web Inf Syst 12(1):2–17
10. Fernando K, Marcos DA, Carlos C, Marco ASN (2016) Optimising resource costs of cloud computing for education. ELSEVIER Future Gener Comput Syst 55:473–479
11. Sukhpal S, Rajkumar B, Inderveer C, Maninder S, Ajith A (2018) BULLET: particle swarm optimization based scheduling technique for provisioned cloud resources. Springer J Netw Syst Manage 26:361–400. https://doi.org/10.1007/s10922-017-9419-y
12. Zhang Xinqian Wu, Tingming CM, Tongquan W, Zhou Junlong Hu, Shiyan BR (2019) Energy-aware virtual machine allocation for cloud with resource reservation. J Syst Softw 147:147–161. https://doi.org/10.1016/j.jss.2018.09.084
13. Min AN, Bilal QM, Saleh A, Omer FR (2019) Cost-efficient resource allocation for real-time tasks in embedded systems. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2019.101523
14. Kholidy HA (2020) An intelligent swarm based prediction approach for predicting cloud computing user resource needs. Comput Commun 151:133–144. https://doi.org/10.1016/j.comcom.2019.12.028
15. Than MM, Thein T (2020) Energy-saving resource allocation in cloud data centers. In: IEEE conference on computer applications (ICCA), Yangon, Myanmar, pp 1–6. https://doi.org/10.1109/ICCA49400.2020.9022819
16. Srimoyee B, Rituparna D, Sunirmal K, Sarbani R (2020) Energy-efficient migration techniques for cloud environment: a step toward green computing. J Supercomput 76:5192–5220. https://doi.org/10.1007/s11227-019-02801-0
17. Mansouri N, Javidi MM (2020) Cost-based job scheduling strategy in cloud computing environments. Distrib Parallel Databases 38:365–400. https://doi.org/10.1007/s10619-019-07273-y
18. Reshmi B, Poongodi P (2020) Profit and resource availability-constrained optimal handling of high-performance scientific computing tasks. J Supercomput 76:4247–4261. https://doi.org/10.1007/s11227-018-2332-7
19. Tripathi A, Pathak I, Vidyarthi DP (2020) Modified dragonfly algorithm for optimal virtual machine placement in cloud computing. J Netw Syst Manage. https://doi.org/10.1007/s10922-020-09538-9
20. Dorigo Marco and Gambardella Luca Maria (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66. https://doi.org/10.1109/4235.585892
21. Chen W, Deelman E (2012) WorkflowSim: a toolkit for simulating scientific workflow in distributed environments. In: 8th IEEE international conference on science, Chicago
22. http://pegasus.isi.edu/schema/DAX