A Novel Approach for Resource Allocation using Swarm on Clouds

Harvinder Singh and Anshu Bhasin
I.K. Gujral Punjab Technical University (IKGPTU), Kapurthala, Punjab, India

Key words: Cloud computing, resource allocation, Swarm intelligence, utilization, execution cost

Corresponding Author:
Harvinder Singh
I.K. Gujral Punjab Technical University (IKGPTU), Kapurthala, Punjab, India

Page No.: 212-218
Volume: 19, Issue 10, 2020
ISSN: 1682-3915
Asian Journal of Information Technology

Abstract: Cloud resource allocation if done efficiently reduce execution cost and improve resource utilization. Resource management policies follow different criteria to discover most turntable resource to fulfill user expectations of minimum cost. Task resource negotiation process improved the probabilities for allocation based on demand by estimation of reasonable cost for current allocation. Such optimization process improves resource availability and facilitates the allocation done based on customer requirements. In this research paper, we present resource allocation problem which further delegates to task resource requirement, resource availability and negotiation process. We proposed efficient resource allocation policy using swarm heuristic approach on simulated cloud environment. The comparison of results with traditional algorithms extemporaneously suggests that proposed algorithm performs better and efficiently allocate resources in terms of QoS.

INTRODUCTION

Cloud computing provides services such as Infrastructure as Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) to users' instant demand[1]. Users access cloud resources using network services and expect Quality of Service (QoS) based results. Therefore, service providers continue doing efforts to improve services quality based on Software Level Agreement (SLA). Tasks heterogeneity, uncertainty and dynamic demand make the resource allocation problem more complex. Provider's facing the major challenges on clouds in current scenario is to allocate resources according to expectation by the users and QoS parameters (execution cost). On the other side resource utilization depends on processing cost using limited number of resources and consider as energy consumption which affects return from investment on cloud resources. Existing literature survey showing that cloud resource management problem is becoming complex day-by-day due to increasing resource demand in different context. Existing resource allocation techniques are unable to find best match from available resources. Resource allocation policy should process user demand on estimated cost to improve reliability and minimize user budget and save data center energy to maximize resource availability. Therefore, an efficient resource allocation policy should be developed to minimize execution cost and maximize resource utilization.

We investigated various research work related to QoS and resource management for resource allocation on clouds. To deliver most desirable resource for task execution there should be resource demand identification framework for efficient resource allocation. To address the resource allocation problem, we have developed a QoS based resource allocation scheme which allocate resource to tasks based on task requirements. A Resource Allocation Scheme (QRAS) has been proposed based on swarm technique in which resources are allocated based on optimal solution. Swarm is population based optimization technique where intelligent agents work collectively and communicate with each other to find most desirable match which minimize cost between demand and supply. So, this technique have the capability to evaluate initial created feasible solutions based on QoS and select optimal solution for resource allocation based on task demand.

The motivation of this research work comes from Virtual Machine (VM) technology in cloud which improved the capability of single physical machine to handle heterogeneous demand. The research challenge is to create and allocate VMs for tasks which should be based on Estimated Execution Cost (EEC) from Available Execution Unit (AEU) where dynamic tasks fired by users with different context. Our proposed technique object is to achieve QoS (reduce cost and maximize resource utilization) by optimization EEC for user demand for resource allocation.

Literature review: Existing literature survey [2-5] reported that resource allocation to tasks in the perspective of QoS is challenging due to: user demand is uncertain resource availability impact on resource allocation resource allocation technique effect on QoS. User demand always comes with different context and user’s expected QoS based results therefore suitable resource allocation based on particular demand is optimization process which impact on cost and resource utilization. In this section brief review presented on resource allocation particular based on QoS.

Singh and Chana[5] presented pattern based resource provisioning technique in terms of QoS (execution cost and time) by analyzes and categories workloads based on requirements[6]. Nosrati and Karimi[7] presented energy efficient method for resource allocation to improve performance and minimize energy consumption by optimizing communication latencies and consolidate allocation decision for appropriate resource to best applicant[7]. Vakilinia *et al*.[8] presented resource allocation model based on poison process where jobs (multiple tasks) submitted by users to access resources. In this model, different cloud resources configuration and tasks (constant and dynamic) used for allocation process to reduce cost and improve utilization[8]. Koch *et al*.[9] presented optimization method for resource allocation particular for educational purpose to reduce cost by workload aware dynamic allocation. Singh *et al*.[10] presented agent based automated service composition model for resource provisioning on cloud to reduce cost based user requirement by appropriate resource assignment. Warneke and Kao[11] presented dynamic resource allocation method (Nephele) to reduce execution cost and explicitly exploit the cloud resources and overrides the limitations of Hadoop. Beloglazov *et al*.[12] presented energy aware resource allocation dynamic provisioning algorithm to minimize energy consumption and improve QoS as customer reliability; better than static allocation. Xiao *et al*.[13] proposed dynamic resource allocation to allocate resources based on demand and improved server utilization and achieve good performance. Skewness is used to measure resource utilization.Krishna[14] proposed scheduling algorithm for balancing load in system using Honey bee behavior with focus on task waiting time and well balance load for tasks on VMs and comparison show better performance in terms of makespan. Pandey *et al*.[15] proposed workflow scheduling algorithm for resource allocation to tasks using Particle Swarm Optimization (PSO) to minimize computation and communication cost; it is limited to workflow applications. Zuo *et al*.[16] presented PSO based resource allocation using self-adaptive PSO to provide the QoS based services. Tsai *et al*.[17] proposed resources allocation algorithm using Differential Evolution (DE) for load balancing and exploitation of resources to reduce makespan and cost. Pacini *et al*.[18] proposed custom VMs generation based on ant pool system by searching dynamic load information concern to host. The algorithm compared with random assignment and genetic algorithm performed well dealing with balancing the hosts that improved performance. Sharkh *et al*.[19] presented dynamic resource allocation to provide runtime reallocation for QoS but execution time is not considered. Netjinda *et al*.[20] proposed optimization method for resource allocation to tasks with optimal solution using PSO to minimize computation cost and completion time using workflow application. Xue *et al*.[21] proposed resource allocation using DE to minimize makespan and energy consumption but not consider execution cost. Zhang *et al*.[22] presented Map reduced based fine grain resource aware scheduler to improve resource utilization and minimize make span, outperforms Hadoop.

MATERIALS AND METHODS

Designed resource allocation algorithm: In this study, we have presented QoS based resource allocation policy using swarm (QRAS) in which user tasks negotiated with cloud resources. The QRAS policy analyzed user tasks and identifies computation cost before resource allocation to improve resource utilization. The QRAS approach is: identify user tasks resource requirement; tasks execution

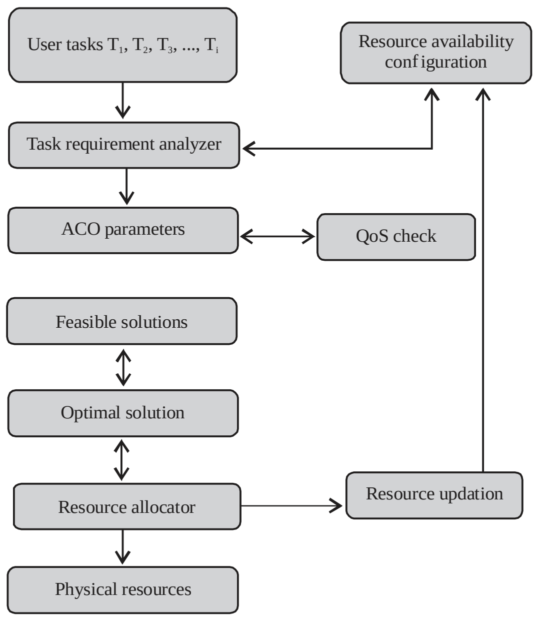

Fig. 1: Efficient resource allocation in cloud computing

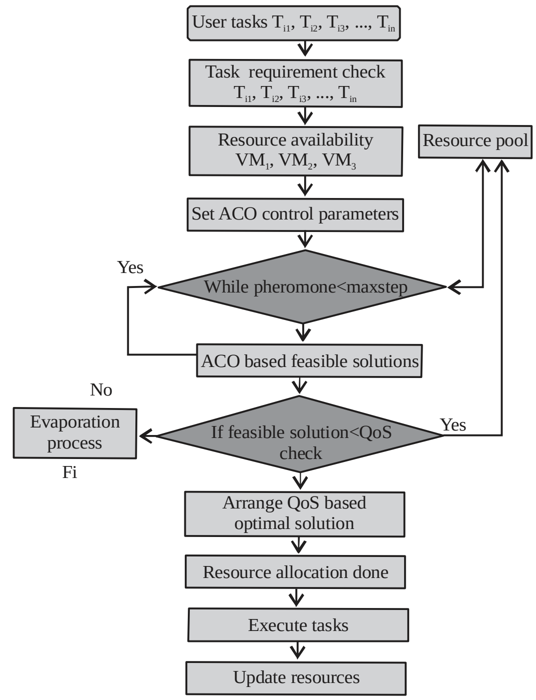

Fig. 2: Resource allocation flow chart

requirements identification using Ant Colony Optimization (ACO) on real time; feasible solutions are negotiated with each other based on QoS for optimal solution; resources allocated based QoS to minimize cost. The QRAS process for resource allocation represented with Fig. 1 which consists of ACO generated feasible solution Optimal solution based on QoS parameters.

Cloud user submitted number of tasks with different context to access resources. The user demand interacts with task requirement analyzer in form of tasks. Resource Availability (RA) contains all the detailed information about number of available VMs, Pes, MIPS, memory, bandwidth, etc. RA information is updated with cloud physical resources. ACO parameters contains the control parameter information that how to execute search process. Feasible solutions generated based on ACO parameters. All the generated feasible solutions stored in the temporary solution list. After generating feasible solutions, optimal solution generated among solutions using intercommunication based QoS check. The QoS check contains information to evaluate feasible solutions for selection of optimal solution. After successful selection of optimal solution resource allocator has taken appropriate action for resource allocation to tasks. Resource Allocator (RAC) performed prominent task for resource allocation based on optimal solution. After successful resource allocation RAC has sent the relevant information to resource updation. Resource updation follows the activity as allocation action performed by RAC and result sent back for resource availability. So, as per cloud user requirements QRAS system efficiently allocate resources based on QoS which minimize execution cost and maximize resource utilization in state of user satisfaction. The main purpose of our designed objective function is to reduce the execution cost of user tasks which is based on QoS.

D QRAS allocated resources to users by optimization process as represented with Fig. 2. Cloud users submitted number of tasks with different requirements and expected results on reasonable cost. User tasks pass through the requirement checker which identifies tasks length to validate demand. Users demands identified based on resource requirements. Resource availability contains the resources to be allocated for current demand and synchronized with resource pool. ACO control contains the parameters to generate feasible solutions. Feasible solutions generated based on pheromone value while termination criteria not meet (maxStep). ACO generated feasible solutions evaluated based on QoS check parameters. QoS check contains the criteria to generate optimal solution and picked only solutions which fulfill QoS based conditions, otherwise solution sent evaporation process. Generated optimal solution further sent to resource allocator. On the basis of optimal solution resource allocator processes the resource allocation decision which minimized cost and improved resource utilization. As resources allocation process generated update resources immediately informed to resource pool. In this research work, we have used ACO technique in cloud computing for resource allocation[23]. Suitable resource allocation is optimization process and has taken time due to number of feasible solution generation process which contains probable solutions for resource allocation. Feasible solutions mark as computation cost as pheromone value lay down on ant route. Pheromone is working as means of communication between feasible solutions to select optimal solution. As max step reached collective communication generated optimal solution for task and resource match which provide customer and provider satisfactions (reduce cost and improve utilization). The evaporation processes remove the unnecessary feasible solutions to save unnecessary usage. ACO is well suited for solving optimization problems to find the efficient task resource matching after a digastric search.

In our presented resource allocation technique allocation action has performed based on demand and resource availability.

User submitted Tasks (T_{i1}, T_{i2}, T_{i3}, T_{in}) to access cloud resources; negotiation process created feasible solutions based on ACO parameters (for VM_{j1} € VM_{jm}/T_{i1} € T_{in}) and QoS based optimal solution selected as cost effective solution ($T_{i1}VM_{j1}$, $T_{i2}VM_{j2}$, $T_{i3}VM_{j3}$, $T_{in}VM_{jm}$) which has minimize cost and improve, resource utilization.

Resource utilization calculation: Utilization calculation for each task:

$$U_{tot} = \sum_{i=1}^{n} t_i \quad (1)$$

Utilization calculation for tasks execution on each VM:

$$U_{vm} = \sum_{i=1}^{q} t_i \quad (2)$$

Total utilization calculation for tasks execution on all VMs:

$$U_{totvm} = \sum_{i=1}^{m} Tvm_i \quad (3)$$

We have been used equations 1-3 to measure the utilization. Resource utilization is calculated based on execution of tasks on VMs as task length and VMs MIPS is heterogeneous. To define equationVM_i j = 1, 2, m is the set of VMs for execution tasks as T_i i = 1, 2, n.; U_{tot} denoted the utilization for each task $T_i$on on VM where n is representing as number of tasks. U_{vm} is denoted with utilization for tasks T_i on each VM where q is representing task size. U_{totvm} is calculating the utilization for tasks T_i on total number of VMs within Data center where m is maximum number of VMs; our objective is to maximize resource utilization, which save energy spent on cloud resources allocation and hence maximize benefits.

Table 1: Experimental data configurations

Parameters	Values
Data center specifications	
Number of Datacenter	01
Number of Hosts	01
VMs specifications	
Number of VMs	5-50
MIPS	1000
PEs per VM	1
VM Memory	512
Bandwidth	100
Users demand specifications	
Number of users	10
Data Set	Montage
Number of Cloudlets	1000
Number of iterations	100

Simulation based experimental work: Simulation tools provide creation of cloud resource, configuration, virtualization, allocation, and implementation of policies to evaluate the performance of algorithm[24].

Workflowsim is used which supports cloud environment customization, configuration such as Data centers, VMs, users, and bandwidth, etc. QRAS policy has been implemented on workflowsim cloud environment and further results are compared using different set of VMs based on QoS parameters execution cost and resource utilization. Our presented resource allocation algorithm has executed Montage dataset as execution of user cloudlets. It is suitable application to execute on distributed nature as cloud environment.

The presented QRAS algorithm experimental work has been executed with the VMs specification and user demand specifications as Table 1. Performance has been measured based on cost calculation and resource utilization which considered as QoS parameters. The experimental performance and analysis compared with three existing algorithms MINMIN, MCT and ACO to evaluate execution cost calculation and resource utilization. Our presented QRAS algorithm has been processed as flow diagram represented with Fig. 2 for resource allocation. To solve user demand cloud resources require for executing user task. Cloud resource capability depends of Data center, Host capacity as CPUs, RAM, Bw, MIPS, VMs.

RESULTS AND DISCUSSION

In this study, comparison of four algorithms has been given for the problem resource allocation to minimize execution cost and improved resource utilization. The results have been generated as statistical analysis to compare the performance of presented QRAS with existing MINMIN, MCT and ACO algorithms. Resource utilization has been measured based on Eq. 1-3 and cost

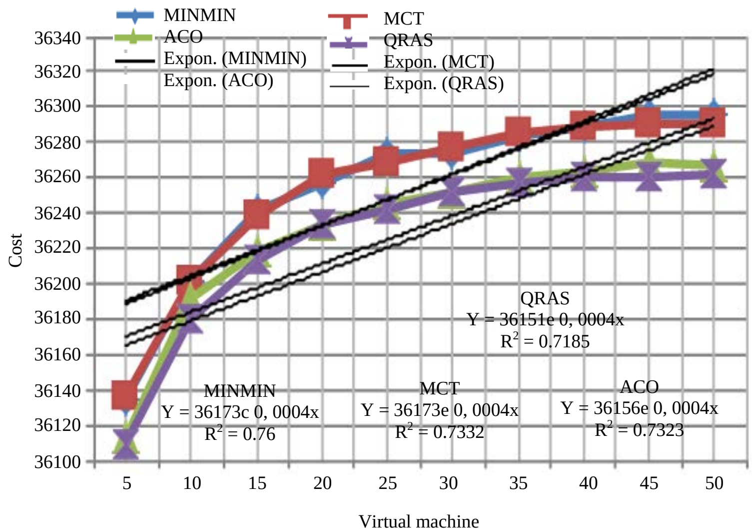

Fig. 3: Impact of change in cost with respect to number of VMs (100 cloudlets)

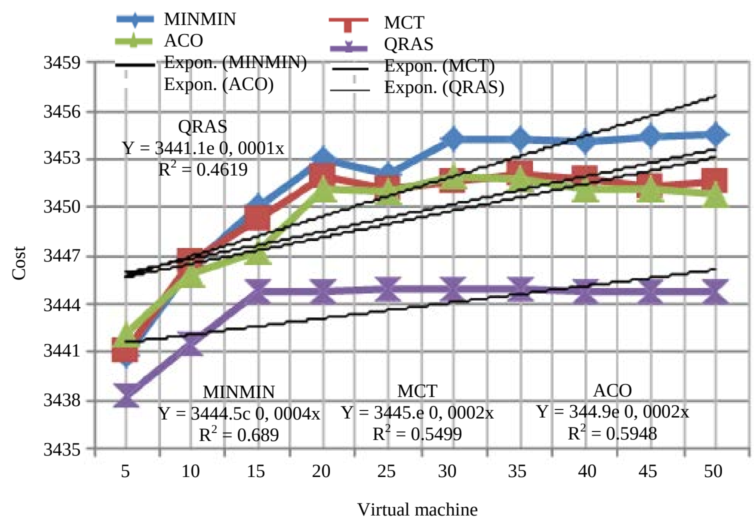

Fig. 4: Impact of change in cost with respect to number of VMs (1000 cloudlets)

measured based on resources spent for task processing. Graphical results represented by Fig. 3 (execution cost for 100 tasks), Fig. 4 (execution cost for 1000 tasks) and Fig. 5 (resource utilization for 1000 tasks) has been shown the different values of execution cost and resource utilization. QRAS compared with others in terms of execution cost and resource utilization for resource allocation and has been taken less cost to finish. Resource allocation algorithms MINMIN, MCT and ACO have been taken more execution cost and resource utilization to finish task. QRAS have been efficiently solve resource management problem, and taken optimal solution for resource allocation which provide user satisfactory and save cloud resources cost. The proposed algorithm comparative with other algorithms efficiently distribute and manage resources. The results are generated in the perspective of cost and time; the R^2 value is calculated using exponential curve. The R^2 value which near to 1 shows exponential growth in all value and other hand value near to 0 shows exponential decrease. The graphical results based on execution of four different resource

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

R^2	Algorithm
MinMin	0.689
MCT	0.549
ACO	0.594
QRAS	0.461

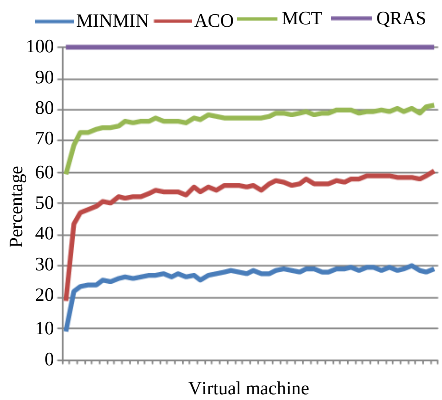

Fig. 5: Impact of change in cost with respect to number of VMs (1000 cloudlets)

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

R^2	Algorithm
MinMin	0.760
MCT	0.733
ACO	0.732
QRAS	0.718

allocation policies and compared in terms of R^2 value. Table 2 show the R^2 for execution cost of execution 100 tasks and Table 3 shows the R^2 for execution cost of execution 1000 tasks. To prove the reliability and stability of our presented QRAS policy we have been generated R^2 value for all the policies and overall comparison of all policies shown that QRAS method take less cost to execute tasks.

The results are generated in the perspective of cost and time; the R^2 value is calculated using exponential curve. The R^2 value which near to 1 shows exponential growth in all value and other hand value near to 0 shows exponential decrease. The graphical results based on execution of four different resource allocation policies and compared in terms of R^2 value. Table 2 show the R^2 for execution cost of execution 100 tasks and Table 3 shows the R^2 for execution cost of execution 1000 tasks. To prove the reliability and stability of our presented QRAS policy we have been generated R^2 value for all the policies and overall comparison of all policies shown that QRAS method take less cost to execute tasks.

Montage application dataset was used to test present method with all the resource allocation methods. Although, this application doesn't consist of large number of vertical tasks per level, it was important to test system stability by testing the cost and utilization used by different sets of virtual machines.

Observations for cost for 100 tasks: The execution cost for 100 tasks reduced by proposed algorithm based on comparison as: 33.09, 16.03, 22.39% of improvement in cost compared w.r.t MinMin, MCT and ACO.

Observations for cost for 1000 tasks: The execution cost for 1000 tasks reduced by proposed algorithm and changed as number of VMs increased which is represented with Fig. 4. The 5.53, 2.05, 1.91% of improvement in cost compared with MinMin, MCT, ACO.

Figure 5 is designed in terms for utilization of computing resources, it was observed for a set of 50 Virtual Machines. The utilization is been evaluated with respect to time per virtual machine. The following observation are derived from by evaluating the utilization parameters.

Observations: MaxMin is observed to perform the resource utilization as 10-30% w.r.t proposed QRAS. MCT is observed to perform the resource utilization as 20-58% w.r.t. proposed QRAS. ACO is observed to perform the resource utilization as 60-62% w.r.t. proposed QRAS. Resource utilization is a major issue and challenge for cloud computing environment, the basic factors affecting cloud services involves resources utilization.

Cloud being a total server oriented architecture, if every computing resource plays important role in terms of cost. The users are paying for several sets of services in real time. Users expecting total utilization of services what they are paying for in cloud environment. Resources utilization was a major objective of the designed work, based on the above observation it was found that QRAS algorithm take the maximum benefit of resources and provide maximum utilization compared to existing algorithm. The evaluation of the system is scaled by considering the results of QRAS algorithm as 100% compared to other existing algorithms. Based on implementation better utilize resources is observed with designed resource allocation policy with better performance is observed

CONCLUSION

Resource allocation in cloud environment based on user requirement is a complex process which impact on cost and resource utilization. In this study we have presented resource allocation technique based on QoS. ACO based approach was used to generate optimal solution from feasible solutions to minimize cost and improve resource utilization. Our presented algorithm efficiently identified exact task requirement before allocation. Experimental results show that proposed algorithm performed better compared to traditional algorithms for resource allocation and improved resource availability in a significant machine.

REFERENCES

01. Jula, A., E. Sundararajan and Z. Othman, 2014. Cloud computing service composition: A systematic literature review. Expert Syst. Appl., 41: 3809-3824.
02. Ma, T., Y. Chu, L. Zhao and O. Ankhbayar, 2014. Resource allocation and scheduling in cloud computing: policy and algorithm. IETE Tech. Rev., 31: 4-16.
03. Zhang, J., H. Huang and X. Wang, 2016. Resource provision algorithms in cloud computing: A survey. J. Netw. Comput. Appl., 64: 23-42.
04. Singh, H. and A. Bhasin, 2017. Efficient resource management technique for performance improvement in cloud computing. Indian J. Comput. Sci. Eng., 8: 33-39.
05. Singh, S. and I. Chana, 2016. Resource provisioning and scheduling in clouds: QoS perspective. J. Supercomputing, 72: 926-960.
06. Singh, S. and I. Chana, 2015. Q-aware: Quality of service based cloud resource provisioning. Comput. Electr. Eng., 47: 138-160.
07. Nosrati, M. and R. Karimi, 2016. Energy efficient and latency optimized media resource allocation. Intl. J. Web Inf. Syst., 12: 2-17.
08. Vakilinia, S., M.M. Ali and D. Qiu, 2015. Modeling of the resource allocation in cloud computing centers. Comput. Netw., 91: 453-470.
09. Koch, F., M.D. Assuncao, C. Cardonha and M.A. Netto, 2016. Optimising resource costs of cloud computing for education. Future Generation Comput. Syst., 55: 473-479.
10. Singh, A., D. Juneja and M. Malhotra, 2017. A novel agent based autonomous and service composition framework for cost optimization of resource provisioning in cloud computing. J. King Saud Univ. Comput. Inf. Sci., 29: 19-28.
11. Warneke, D. and O. Kao, 2011. Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. IEEE Trans. Parallel Distrib. Syst., 22: 985-997.
12. Beloglazov, A., J. Abawajy and R. Buyya, 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gen. Comput. Syst., 28: 755-768.
13. Xiao, Z., W. Song and Q. Chen, 2013. Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Trans. Parallel Distrib. Syst., 24: 1107-1117.
14. Krishna, P.V., 2013. Honey bee behavior inspired load balancing of tasks in cloud computing environments. Appl. Soft Comput., 13: 2292-2303.
15. Pandey, S., L. Wu, S.M. Guru and R. Buyya, 2010. A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications (AINA'10), April 20-23, 2010, IEEE, Perth, Western Australia, ISBN:978-1-4244-6695-5, pp: 400-407.
16. Zuo, X., G. Zhang and W. Tan, 2014. Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IAAS cloud. IEEE. Trans. Autom. Sci. Eng., 11: 564-573.
17. Tsai, J.T., J.C. Fang and J.H. Chou, 2013. Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput. Oper. Res., 40: 3045-3055.
18. Pacini, E., C. Mateos and C.G. Garino, 2015. Balancing throughput and response time in online scientific clouds via ant colony optimization (SP2013/2013/00006). Adv. Eng. Software, 84: 31-47.
19. Sharkh, M.A., M. Jammal, A. Shami and A. Ouda, 2013. Resource allocation in a network-based cloud computing environment: Design challenges. IEEE. Commun. Mag., 51: 46-52.
20. Netjinda, N., B. Sirinaovakul and T. Achalakul, 2012. Cost optimization in cloud provisioning using particle swarm optimization. Proceedings of the 9th International Conference on Electrical Engineering-Electronics, Computer, Telecommunications and Information Technology (ECTI-CON'12), May 16-18, 2012, IEEE, Phetchaburi, Thailand, ISBN:978-1-4673-2026-9, pp: 1-4.
21. Xue, J., L. Li, S. Zhao and L. Jiao, 2014. A study of task scheduling based on differential evolution algorithm in cloud computing. Proceedings of the 2014 International Conference on Computational Intelligence and Communication Networks (CICN'14), November 14-16, 2014, IEEE, Bhopal, India, ISBN:978-1-4799-6930-2, pp: 637-640.
22. Zhang, Q., M.F. Zhani, Y. Yang, R. Boutaba and B. Wong, 2015. PRISM: Fine-grained resource-aware scheduling for MapReduce. IEEE. Trans. Cloud Comput., 3: 182-194.
23. Pacini, E., C. Mateos and C.G. Garino, 2014. Distributed job scheduling based on swarm intelligence: A survey. Comput. Electr. Eng., 40: 252-269.
24. Chen, W. and E. Deelman, 2012. Workflowsim: A toolkit for simulating scientific workflows in distributed environments. Proceedings of the 2012 IEEE 8th International Conference on E-Science (E-Science'12), October 8-12, 2012, IEEE, Chicago, Illinois, ISBN: 978-1-4673-4467-8, pp: 1-8.