



Task scheduling in a cloud computing environment using HGPSO algorithm

A. M. Senthil Kumar¹ · M. Venkatesan²

Received: 30 October 2017 / Revised: 29 January 2018 / Accepted: 12 March 2018 / Published online: 26 March 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Cloud computing delivers computing resources like software and hardware as a service to the users through a network. The main idea of cloud computing is to share the tremendous power of storage, computation and information to the scientific applications. In cloud computing, the user tasks are organized and executed with suitable resources to deliver the services effectively. There are plenty of task allocation techniques that are used to accomplish task scheduling. In order to enhance the task scheduling technique, an efficient task scheduling algorithm is proposed in this paper. Optimization techniques are very popular in solving NP-hard problems. In this proposed technique, user tasks are stored in the queue manager. The priority is calculated and suitable resources are allocated for the task if it is a repeated task. New tasks are analyzed and stored in the on-demand queue. The output of the on-demand queue is given to the Hybrid Genetic-Particle Swarm Optimization (HGPSO) algorithm. To implement HGPSO technique, genetic algorithm and particle swarm optimization algorithm are combined and used. HGPSO algorithm evaluates suitable resources for the user tasks which are in the on-demand queue.

Keywords Cloud computing · Optimization techniques · Task scheduling · Genetic algorithm · Particle swarm optimization algorithm

1 Introduction

Cloud computing provides application access to all the users and businesses based on demand from any part of the world. Computing becomes a delivered service in the cloud computing to the users. Resources are abstracted and provided as services to the users based on their demand. Data centers are the major elements which are maintained by the cloud providers. Cloud service providers charge the consumers for accessing the resources over the network. Instead of investment cost towards the resources, enterprises receive the resources as services. So consumers are able to reduce their investment cost towards the resources. Service Level Agreements (SLAs) are established between

consumers and service providers to mention the guarantees of service providers to the consumers [1].

Task scheduling distributes the user tasks on the resources with the aim of maximizing the utilization rate and minimizing the task execution time. Task scheduling can be classified into two types: static scheduling and dynamic scheduling. In static scheduling, task and resource details are known by the scheduler. In dynamic scheduling, task and resource details not known in advance. The Scheduler makes dynamic scheduling plans to distribute user tasks to the suitable resources.

Dynamic scheduling problem is a kind of NP-complete problem in which computation exponentially increases with respect to the problem size [2]. Meta-heuristic algorithms like genetic algorithm and Particle Swarm Optimization algorithm are popular to solve NP-complete problems. These algorithms are biology-inspired algorithms. Genetic algorithm became very popular due to its simplicity and accuracy. The Genetic algorithm will become slow for the problems larger in size. Hence, the

✉ A. M. Senthil Kumar
senthil1185@gmail.com

¹ Department of CSE, Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu, India

² KSR Institute for Engineering and Technology, Tiruchengode, Tamil Nadu, India

Particle Swarm Optimization algorithm is combined with a genetic algorithm for faster convergence.

This paper's main contribution is discussed as follows:

- (1) This paper presents a priority based task scheduler, which sort the user tasks based on the priority. Priority is calculated using task length and memory.
- (2) This paper illustrates the Hybrid Genetic-Particle Swarm Optimization algorithm (HGPSO) which takes the input from the on-demand queue and evaluates the suitable resources for the user tasks.

Finally, the proposed HGPSO algorithm is validated and compared with the other Meta-heuristic algorithms like genetic algorithm and Particle Swarm Optimization algorithm in a cloud simulated environment. The remaining contents of this paper are organized as follows. Section 2 discusses the related works and Sect. 4 explains the proposed methodology. Section 5 shows the results in detail and Sect. 6 presents the conclusion of the paper.

2 Related works

Ghanbari et al. [3] presented a priority based job scheduling method using mathematical statistics. This technique focused on the priority of the jobs. A multiple criteria decision model was used in the algorithm for calculating priority. Consistency and time are the issues related to this model. PSO and GA algorithms are combined and the three various hybrid strategies [4] are presented to find a better solution with faster convergence. In the first strategy, the crossover is applied over gbest particle with the GA chromosome. Here the gbest particle does not change its position. In the second strategy, pbest particles change their positions by the GA mutation operator. In the third strategy, GA's solution is assigned as an initial population to the PSO. Total iterations are shared by GA and PSO. Hybrid algorithm HGAPSO is presented [5] for force based Simultaneous Analysis and Design by combining PSO with GA. Here PSO is used to enhance the elites by introducing cross over into the society.

Sossa and Buyya [6] presented a meta-heuristic optimization method to minimize the execution cost and satisfy the deadline constraints for workflows over IaaS clouds. Mudjihartono et al. [7] proposed Parallel GA-PSO algorithm to solve a job scheduling problem. The result of the proposed technique proved that it is faster than the standard GA algorithm. Meng et al. [8] presented a hybrid PSO algorithm which is fused with genetic algorithm (GA) and simulated annealing (SA). The proposed technique escaped from falling into a local minimum. The Results of the proposed technique are better than other hybrid PSO algorithm as it consumed lesser time for task completion.

Manasrah et al. [9] proposed a new workflow task scheduling in a cloud environment. The GA-PSO_HEFT algorithm selects VMs based on workflow task size and VM execution speed. Selected VMs execute the tasks in the minimum time. This GA_PSO_HEFT algorithm also executes tasks over the VMs with a balanced load distribution over slow and fast VMs. This technique reduces the makespan when compared with GA, PSO, HSGA, and WSGA algorithms. Also GA-PSO_HEFT algorithm improves the load balancing when compared with GA, PSO, and HSGA algorithms.

A hybrid task scheduling method [10] for cloud computing using genetic and PSO algorithm is introduced. In this technique, the cost function is used based on genetic and PSO algorithms. The Cost function evaluates the task execution time on resources in a cloud environment. These new models minimize the communication cost and critical path length. The results of the proposed model outperform other existing algorithms.

Various methods [11–15] and hybrid techniques are available with the combination of GA and PSO algorithms. But they do not optimize various QoS parameters. The proposed HGPSO algorithm focuses on various QoS parameters to be optimized.

3 Problem formulation

QoS parameters used in this work refer to execution time, scalability and availability.

T : total time for execution, vm : virtual machine, R : resource, St : starting time of task execution on resource r , Ft : finishing time of task execution on resource r , $CTir$: completion time of task i on resource r , $SCir$: scalability of resource, $AVir$ = availability of resource.

$$CTir = \sum_{i=1}^M Ft_i - St_i, 1 \leq i \leq M, 1 \leq r \leq k \quad (1)$$

$$SCir = \sum_{i=1}^M vm_i / Rr, 1 \leq i \leq M, 1 \leq r \leq k \quad (2)$$

$$AVir = \sum_{i=1}^M vmi / T, 1 \leq i \leq M, 1 \leq r \leq k \quad (3)$$

4 Proposed task scheduling model

The proposed Task scheduling model is shown in Fig. 1. User tasks are sent to the queue manager for evaluating the type of the tasks.

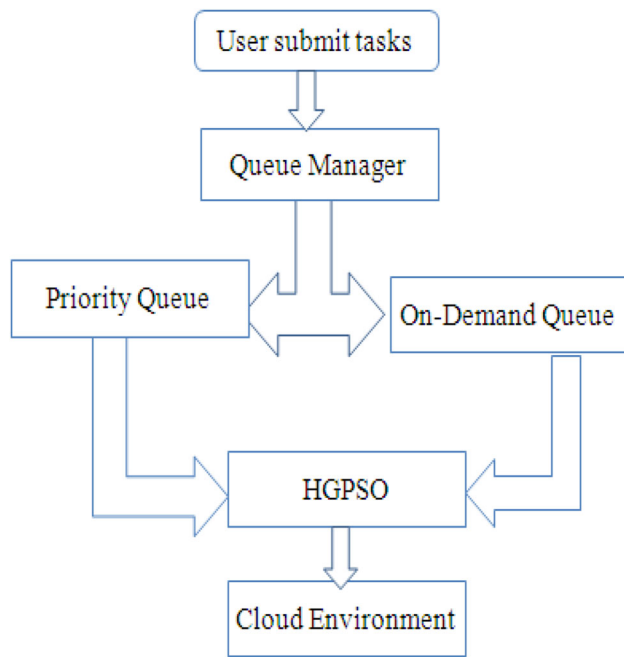


Fig. 1 Task scheduling model

Queue manager will check the tasks that are in the task history. If the task type is already present, that task is sent to priority queue and task priority is assigned using priority based task scheduler and the required resource is allocated to the user task. If tasks are new and not in the history, that type of task is sent to the on-demand queue. This on-demand queue fetches the user tasks and forwards it to the HGPSO algorithm. An HGPSO algorithm accepts the user tasks as input and finds the suitable resources for the user tasks.

Initially, HGPSO calls the genetic algorithm to generate the initial population. Selection operation is used to find the fitted chromosomes from the population set. With the selected individuals from the GA, the PSO's initial population is initiated. Before this, the population of GA is transformed into PSO particles. The PSO algorithm finds the best particles by finding through pbest and gbest methods. Best particle from PSO solution is transformed to GA for a crossover with the chromosome that is selected randomly from the fitted chromosomes. GA applies a mutation operation and suitable resources are assigned for the tasks.

4.1 Priority-based task scheduling

In this scheduling method, the user tasks are analyzed for the priority. Priority queue receives all the user tasks and checks the priority for all the tasks. Queue manager has a

list of task history, which is executed previously. Priority is given to the tasks which are previously executed on the resources in the cloud. Priority is calculated based on task length and the memory size. Average task length and average memory required are calculated. Each task length and memory required is compared with the average task length and average memory. If checked values are less, the priority is set as high otherwise the priority is set as low.

Pseudo code: Priority-based Task scheduling

Input: Set of tasks $T = (T_1, T_2, \dots, T_i)$ containing its length $L = (L_1, L_2, \dots, L_n)$ and memory $M = (M_1, M_2, \dots, M_j)$

Output: Arrange the task based on Priority Queue

Procedure:

Get the lengths of each task using *getlength()* function

Get the memory sizes of each task using *gesize()* function

Calculate average length using *avg(L)*

Calculate average memory size using *avg(M)*

Check each task T_i length with its *avg(L)*

For each $T = 0$ to i do

If $L < avg(L)$

Assign $T_i = High$

Else

Assign $T_i = Low$

End For

Check each task T_i memory size with its

avg(M)

For each $T = 0$ to i do

If $M < avg(M)$

Assign $T_i = High$

Else

Assign $T_i = Low$

End For

Calculate priority using adding the task length and memory.

Repeat the steps until entire task is evaluated.

End

4.2 HGPSO-based task scheduling

Figure 2 shows the task scheduling technique using HGPSO algorithm. The user tasks which are rejected by the

priority queue are considered as input for the on-demand queue. The On-demand queue output is given to the HGPSO algorithm. HGPSO algorithm calls the GA algorithm to initialize the individuals. Chromosomes are encoded indirectly to represent the tasks and resources. Resources are represented by gene value. The fitness function given in Eq. (4) evaluates all the chromosomes in the population and fitted individuals are identified from the population.

The fitness function is represented as follows

$$Fitness = \min \sum_{ir} \frac{1}{CT_{ir}} + \max \sum_{ir} (AV_{ir} + SC_{ir}) \quad (4)$$

In selection operation, ranking methods are applied to select the best individuals for further operation. The solution of the GA is transformed as PSO's initial population by using the following function.

$$T_i^S(0) = \alpha \cdot GA_n \quad (5)$$

α is a constant and GA_n is a solution of GA $T_i^S(0)$ is an initial solution of PSO.

PSO initializes the particles using the random initialization operation of the PSO. PSO evaluates all the tasks against pbest and gbest values and that will be updated once the pbest and gbest values are greater. At every iteration, particles update the position and velocity based on the best values using Eq. (6).

$$V(t) = \omega v_1(t-1) + k_1 n_1 (x_1 \wedge (t-1) - x_1(t-1)) + k_2 n_2 (x^*(t-1) - x_1(t-1)) \quad (6)$$

where $x_1(t)$ is Position vector, i is Particle index, $v_1(t)$ is velocity vector $x_1 \wedge (t-1)$ is Position of particle, $x(t-1)$ is best particle's position vector, n_1 and n_2 is random numbers between 0 and 1, k_1 and k_2 is positive constants, ω is inertia factor.

The best particle value is encoded as GA's chromosomes for crossover. This chromosome is crossed over with the chromosome which is selected randomly from the GA's selection output. The brand new chromosome is generated and flip Mutation is applied to the offspring by swapping 1s to 0s and 0s to 1s that is generated in the crossover step to maintain the genetic diversity. Finally, the suitable resources that are given as output from the HGPSO algorithm are allocated to the user tasks.

Pseudocode: HGPSO based task scheduling

Begin

Step 1: Initialize the parameter setting

- Setting the parameter
- Generate an Initial Population

Step 2: Calculate the fitness function

- Candidate Solution $i, i=1,2,3,\dots,n$
- $fitness_i = f(GA)$

Step 3: Perform selection operation

- for each candidate solution $i, i=1,2,3,\dots,n$

$$p_i = q'(1-q)^{rank-1}$$

Step 4: $S_{ps0} \leftarrow 0$

1. Generating the initial particles
 - a. Parameter are obtain from external HGA method (p_i)
 - b. Initialize the PSO particle
2. While $S_{ps0} \leq p_i$ do
 - a. Compute the objective function
 - b. **If** $fitness(CT_k \leq pbest_k)$ **then**

$$pbest_k \leftarrow CT_k;$$
End if
If $fitness CT_k \leq gbest_k$ **then**

$$gbest_k \leftarrow CT_k;$$
End
 - c. Update the particle velocity & Position
3. End While

For each candidate solution $I, i=1,2,\dots,A_{GA}$ do

If $rand() \leq gbest$ then

Step 5: Perform Crossover operation

Step 6: Conduct Mutation process

End If

End For

End For

Step 7: Performing allocation

$$A_{GA} \leftarrow A_{GA} + 1$$

End

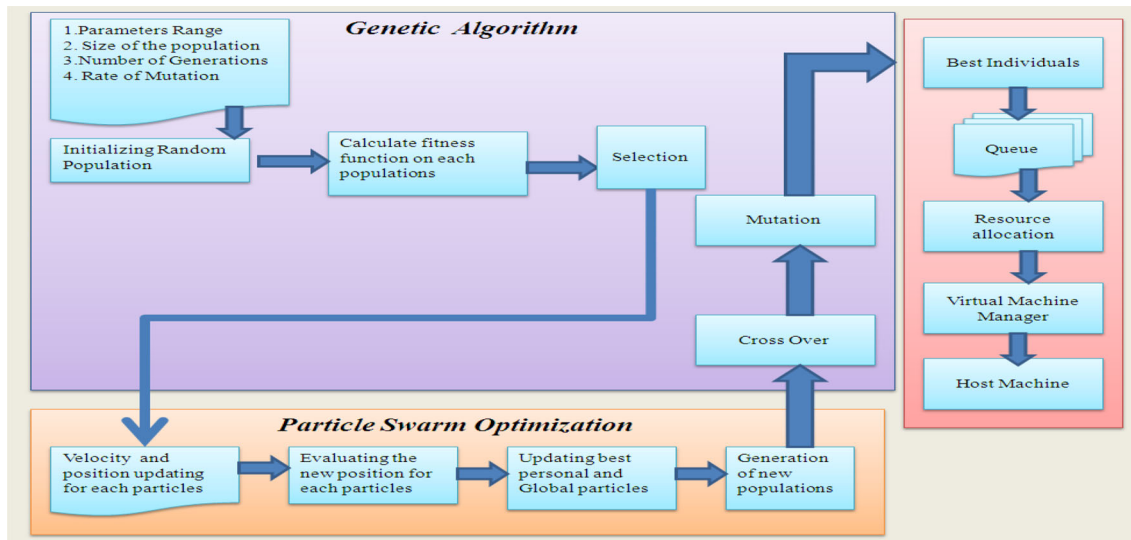


Fig. 2 HGPSO-based task scheduling

Table 1 Simulation parameters

Category	Factors	Values
Resource	Range	0-25
	MIPS	1000
	Image size	10,000 MB
	Ram	1024 MB
	Bandwidth	2000 B/S
	Pes number	25
Task	Range	50
	Input	600
	output	900

5 Experimental results and analysis

In this section, experimental setup and result analysis are presented. The proposed Hybrid Genetic-Particle swarm optimization algorithm is experimented based on the

execution time, availability and throughput. The result of the proposed technique is compared with the genetic algorithm and Particle swarm optimization algorithm.

Table 1 details the cloud simulation environment parameters. The maximum number of tasks taken is 50. 25 resources are considered for the simulation.

Figure 3 shows the completion time for the proposed GA-PSO and it compares with the GA, PSO and GA_PSO_HEFT algorithms when the tasks vary from 10 to 50. It can be observed that completion time of proposed HGPSO algorithm is minimized when compared with GA and PSO algorithms.

Figure 4 presents the scalability comparison between GA-PSO and GA, PSO algorithms when the tasks vary from 10 to 50. It can be concluded that scalability of proposed HGPSO algorithm is greater than GA, PSO and GA_PSO_HEFT algorithms.

Figure 5 shows the availability of proposed GA-PSO and GA, PSO algorithms when the task values differ from

Fig. 3 Completion time

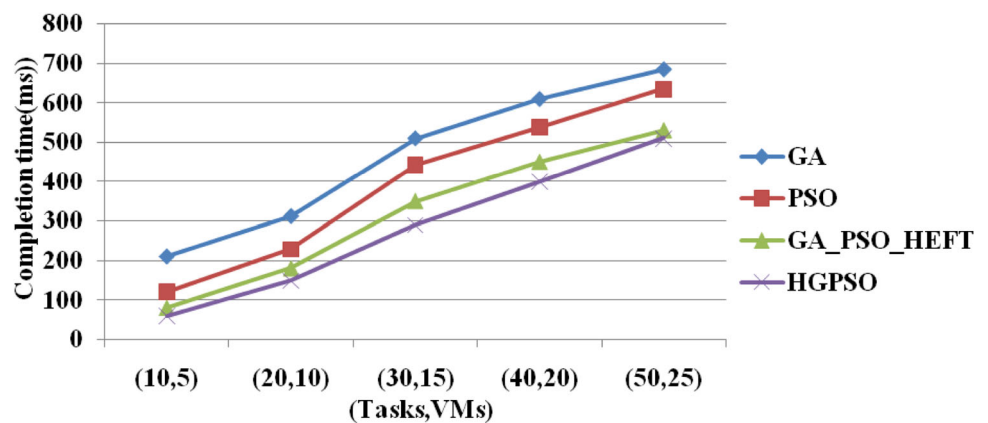


Fig. 4 Scalability

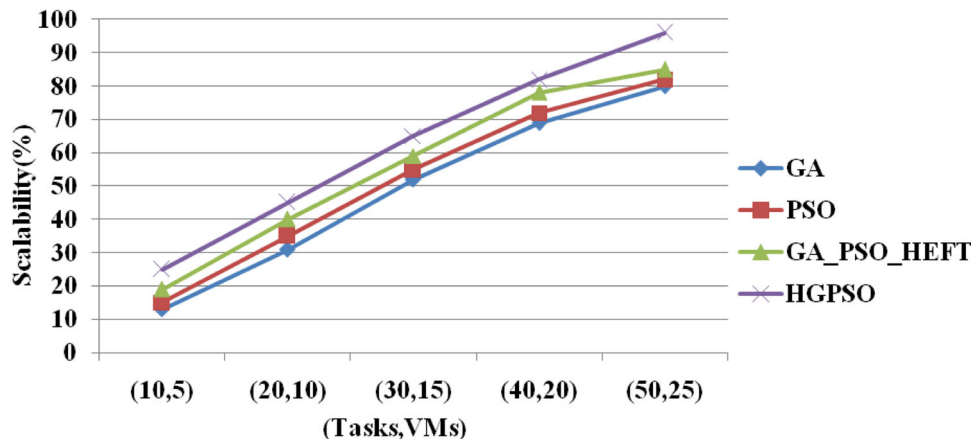
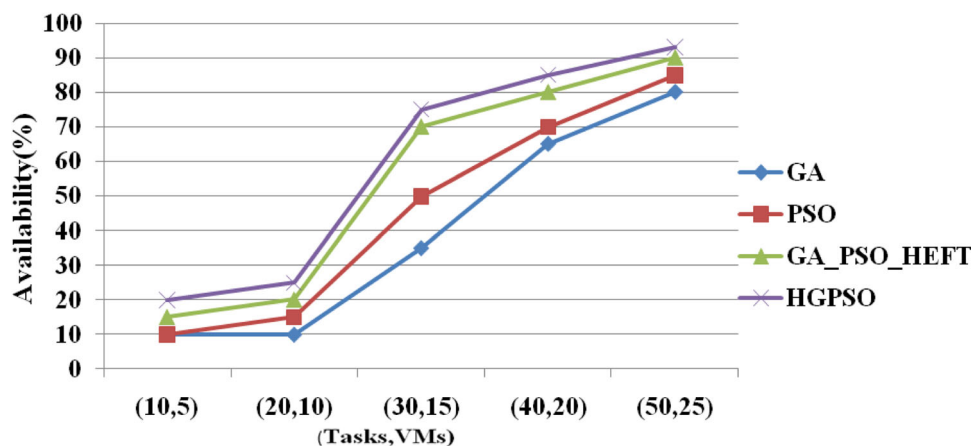


Fig. 5 Availability



10 to 50. It can be noted that the availability of proposed HGPSO algorithm is higher than GA, PSO and GA_PSO_HEFT algorithms.

6 Conclusion

Task allocation plays a major issue in cloud computing environment. A hybrid task scheduling method is proposed in this paper.

Genetic and Particle Swarm Optimization algorithms are combined in this work to allocate the suitable resources to the user tasks. The proposed algorithm decreases execution time and improves availability and scalability rate when compared to genetic and particle swarm optimization algorithms. Furthermore, more QoS parameters will be considered as a novelty in improving the task allocation.

References

- Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., Brandic, I.: Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Gener. Comput. Syst.* **25**(6), 599–616 (2009)
- Mocanu, E.M., Florea, M., Andreica, M.I., Țăpuș, N.: Cloud computing—task scheduling based on genetic algorithms. In: *IEEE System Conference (Syscon)*, pp. 1–6 (2012)
- Ghanbari, S., Othman, M.: A priority based job scheduling algorithm in cloud computing. *Procedia Eng.* **50**, 778–785 (2012)
- Premalatha, K., Natarajan, A.M.: Hybrid PSO and GA for global maximization. *Int. J. Open Probl. Comput. Sci. Math.* **2**(4), 597–608 (2009)
- Kaveh, A., Malakouti Rad, S.: Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design. *Iran. J. Sci. Technol. Trans. B.* **34**, 15–34 (2010)
- Alejandra Rodriguez, M., Buyya, R.: Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds. *IEEE Trans. Cloud Comput.* **2**(2), 222–235 (2014)
- Mudjihartono, P., Setthawong, R., Tanprasert, T.: Parallelized GA-PSO algorithm for solving Job Shop Scheduling Problem. In: *2nd International Conference on Science in Information Technology (ICSITech)*, pp. 103–108 (2016)
- Meng, Q., Zhang, L., Fan, Y.: A hybrid particle swarm optimization algorithm for solving job shop scheduling problems. In:

- Zhang, L., Song, X., Wu, Y. (eds.) Theory, Methodology, Tools and Applications for Modeling and Simulation of Complex Systems, AsiaSim 2016, SCS AutumnSim 2016, vol. 644, pp. 71–78. Communications in Computer and Information Science. Springer, Singapore (2016)
9. Manasrah, A.M., Ali, H.B.: Workflow scheduling using hybrid GA-PSO algorithm in cloud computing. *Wirel. Commun. Mob. Comput.* **2018**, 1–16 (2018)
 10. Kamaliniya, A., Ghaffari, A.: Hybrid task scheduling method for cloud computing by genetic and PSO algorithms. *J. Inf. Syst. Telecommun.* **4**, 271–281 (2016)
 11. Shyamala, K., Sunitha Rani, T.: An analysis on efficient resource allocation mechanisms in cloud computing. *Indian J. Sci. Technol.* **8**(9), 814–821 (2015)
 12. Chalack, V.A., Razavi, S.N., Gudakahriz, S.J.: Resource allocation in cloud environment using approaches based particle swarm optimization. *Int. J. Comput. Appl. Technol. Res.* **6**(2), 87–90 (2017)
 13. Zeng, Z., Truong-Huu, T., Veeravalli, B., Tham, C.-K.: Operational cost-aware resource provisioning for continuous write applications in cloud-of-clouds. *Clust. Comput.* **19**, 1–14 (2016)
 14. Sontakke, V., Patil, P., Waghmare, S., Kulkarni, R., Patil, N.S., Saravanapriya, M.: Dynamic resource allocation strategy for cloud computing using virtual machine environment. *Int. J. Eng. Sci. Comput.* **6**(5), 4804–4806 (2016)
 15. Singh, S., Chana, I.: Q-aware: quality of service based cloud resource provisioning. *Comput. Electr. Eng.* **47**, 138–160 (2015)



A. M. Senthil Kumar received his B.Tech. degree in Information Technology from Anna University, Chennai in 2005 and M.Tech. Degree in Anna University Coimbatore in 2010 and currently he is pursuing Ph.D. in Anna University, Chennai. He is working as Assistant Professor & Head in Computer science and Engineering department of Tejaa Shakthi Institute of Technology for women.



M. Venkatesan completed his Ph.D. in Anna University Coimbatore in 2011 and he has published 20 research papers in both conference and international journal. Nearly 10 candidates are doing their research work under his supervision and he is working as Principal in K.S.R. Institute for Engineering and Technology.