

# Cluster based Hybrid Approach to Task Scheduling in Cloud Environment

Y. Home Prasanna Raju<sup>1</sup>  
Department of CSE  
Acharya Nagarjuna University  
Guntur, A.P., India

Nagaraju Devarakonda<sup>2</sup>  
Department of IT  
Lakireddy Bali Reddy College of Engineering  
Vijayawada, A.P., India

**Abstract**—Cloud computing technology enables sharing of computer system resources among users through internet. Many numbers of users may request for sharable resources from a cloud. The sharable resources must be effectively distributed among requested users with in a less amount of time. Task scheduling is one of the ways of handling the user requests effectively in a cloud environment. There were many existing biologically inspired optimization techniques worked with task scheduling problems. The proposed paper is aimed at clubbing clustering techniques with biologically inspired optimization algorithms for deriving better results. A new hybrid methodology KPSOW (K-means with PSO using weights) has been proposed in the paper, which makes use of the strengths of both the K-means and PSO algorithms with the inclusion of weights concept. The results have shown that KPSOW has made considerable changes in reducing the makespan and improves the utilization of computing resources in the cloud.

**Keywords**—Task scheduling; cloud computing; clustering; k-means; particle swarm optimization; makespan

## I. INTRODUCTION

In the Computer science and information technology field, usage of internet plays an important role for sharing of resources among many people. Many technologies came for supporting the distribution of resources through a network. Distributed computing is one of the technologies which support the distribution of resources in a network. Task scheduling is the mostly used key factors in distributed system. Simulated annealing techniques [1] can be applied for scheduling tasks in a distributed environment for better results. Cloud computing is one of the distributed technologies which provides a platform for sharing of resources via pay per use model through internet. Cloud provides services [2] to users in three categories. The categories are Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS) and Infrastructure-as-a-Service (IaaS). Cloud can be viewed as a Network-as-a-Service (NaaS) [3] using virtualization process. There are many challenges/issues [4,5] to be faced for a reliable cloud computing environment. Clouds can be of different types like private cloud, public cloud and hybrid cloud which is the combination of both the private and public clouds. To utilize the cloud services effectively, task scheduling can be used in the cloud environment. For effective utilization, the parameters that can be considered are makespan, energy consumption, resource utilization etc. There must be minimum energy consumption while handling the cloud tasks. How energy can be minimized using virtual

machine scheduling [6] in cloud centers was shown by Chaima Ghribi, Makhlof Hadji and Djamel Zeghlache. Balancing of load in cloud environment [7, 8, 9] is another important aspect to be considered for speedy response from the cloud. If the load is properly balanced, the computing resources can evenly get the cloud tasks from the scheduler which creates a balancing environment, even when high complexity tasks or more tasks enter the cloud. Priyansh Srivastava, Bhavesh Gohil, and Dhiren Patel [10] showed the load balancing model for a cloud using Cloudsim tool. Genetic algorithms [11, 12] can also be useful for task scheduling. Genetic algorithms belong to a class of evolutionary algorithms which are used for generating high quality optimization solutions. They rely on bio-inspired operations like selection, crossover, mutation etc. Another way of giving optimizing solutions to task scheduling is through bio-inspired algorithms like Particle Swarm Optimization (PSO) [13, 14, 15, 16] and Ant Colony Optimization algorithms [17, 18] etc. PSO imitates the behavior of birds searching for food. Birds move to next location where more food is available. The bird's movement is based on its local search criteria. Every time bird's best position and its velocity is considered to meet the global search criteria which is final optimized solution whereas ACO imitates the behavior of ants searching for food. When an ant finds food, it moves to that location by releasing a pheromone on its travelled path. Now the other ants follow the path by smelling that pheromone. The pheromone may evaporate as the time goes on. So the ant's movement during the search path is based on the concentration of pheromone laid on that path. Optimized paths are found using the above ant's behavior.

The tasks to be scheduled are of different types like having different complexity levels. If the similar tasks are taken into groups before allocating them to computational resources, then there is a chance of generating optimizing solutions for a cloud environment. Hence to obtain better makespan results in cloud environment, the proposed paper is made making use of clustering techniques with the help of bio-inspired algorithms for cloud task scheduling problem. A new hybrid algorithm KPSOW has been proposed in the paper. KPSOW combines the strengths of both K-means [19, 20, 21, 22] and PSO algorithms using weights concept.

The rest of the paper is organized as follows. Sections 2, 3 and 4 explain the working nature of K-Means algorithm, FCFS scheduling algorithm and Particle Swarm Optimization algorithm, respectively. The three sections also explain how

these algorithms can be mapped to task scheduling problem in cloud environment. Section 5 explains methodology which gives a complete idea of proposed work. Section 6 describes and depicts the results of proposed work. Finally, conclusions are given in Section 7.

## II. K-MEANS CLUSTERING

K-means clustering algorithm is one of the popular used algorithms for clustering. K defines number of clusters to be generated from the process. It collects a set of tasks as input and separates them into clusters by finding distances between mean values of the clusters. Euclidean distance measure is used for finding mean distances. Number of clusters to be generated is to be given as domain knowledge to the algorithm. The algorithm proceeds as follows. Let us assume K value is 2 and there are n tasks. Initially, each cluster is allocated a single task randomly. Now the clusters task length is considered as mean value in the first iteration. Euclidean distances are applied to all the remaining task lengths in the set from the mean values of cluster1 and cluster2. The task lengths which are having minimum mean distance is allocated to those corresponding clusters. Now, new mean values are calculated for the newly generated clusters. Mean squared error value is calculated at each iteration step to find error while forming clusters. Calculation of mean squared error value is shown in the equation (1).

$$e = \sum_{p=1}^K \sum_{t \in C_q} |t - md_q| \quad (1)$$

Where  $e \rightarrow$  Mean squared error

$K \rightarrow$  number of clusters to be generated

$C \rightarrow$  represents a cluster

$t \rightarrow$  is a task in the cluster  $C_q$

$md_q \rightarrow$  a mean value in the cluster  $C_q$

Current error value is compared with the previous iteration error value. If the error is converged or there are no more changes in the cluster objects then the algorithm is stopped. The K-means clustering algorithm can be used in cloud environment for grouping similar type of complex tasks.

## III. FCFS SCHEDULING ALGORITHM

FCFS is one of the simplest scheduling algorithms used for scheduling tasks from the task ready queue. It is commonly referred to as First-Come-First-Serve scheduling algorithm. FCFS is used when all tasks are given similar priority. The working nature of FCFS algorithm is: it executes the tasks as their arrival order in the ready queue: i.e. as the FCFS name suggests, the task which comes first will get executed. FCFS has the property called FIFO (First-In-First-Out). The same original FCFS algorithm can be used to process the tasks by computational resources in the order of their presence in the cloud environment. Let us assume there are 15 tasks in the ready queue and 3 computational resources (virtual machines) in a cloud environment. By using FCFS algorithm, task1 is assigned to virtual machine1 and task2 is assigned to virtual machine2 and task3 is assigned to virtual machine3. When task1 is executed successfully then task4 is assigned to virtual machine1 and when task2 is executed successfully then task5 is assigned to virtual machine2. The same process is repeated until all tasks get executed. For this scenario, a total of 5 tasks

are assigned to each virtual machine on an average. Non-preemption and not having resources utilization in parallel are the common problems of FCFS. But FCFS is the simplest task scheduling algorithm for optimizing resources. There were papers [23] giving an analysis of scheduling algorithm with priority was done based on FCFS.

## IV. PARTICLE SWARM OPTIMIZATION

One of the Bio-Inspired Optimization algorithms is Particle Swarm Optimization (PSO) algorithm. As the name suggests, PSO simulates the behavior of birds in the process of searching for food. The birds follow a certain strategy to search for food. The strategy which takes less time to search for food is called a best strategy. In PSO algorithm, bird strategies are called as particles. Each strategy is assigned a fitness value. Depending upon the application, the particle which has an optimum fitness value is treated as an optimum solution to the given problem. The algorithm starts with a set of particles. Fitness values are calculated for all particles in each iteration. For each iteration, two values are updated. First one is 'pbest' and the second one is 'gbest'. 'pbest' is personal best position of each particle and 'gbest' is global best position of particles so far in all iterations. From the next iteration, particle positions and velocities are updated with the help of previously calculated best values as shown in the equations (2) and (3).

$$UV[] = UV[] + l_1 * rand() * (pbest[] - current[]) + l_2 * rand() * (gbest[] - current[]) \quad (2)$$

$$current[] = current[] + UV[] \quad (3)$$

where

$UV[] \rightarrow$  array of updated velocities of particles

$current[] \rightarrow$  array of current positions of particles

$l_1$  and  $l_2 \rightarrow$  learning factors, usually value 2 is taken to both the factor variables

$rand() \rightarrow$  random function which takes values between 0 and 1

The above process is repeated up to a maximum number of iterations or up to the optimal solution is converged. The same PSO approach can be mapped to cloud task scheduling problem for obtaining an optimal solution. The assignment of cloud tasks to virtual machines is considered as a particle. The time it takes to execute tasks by respective virtual machines is considered as a fitness function. The position of particles is the placement of tasks to the virtual machines. Better optimal solution can be obtained using PSO approach compared to FCFS algorithm.

## V. METHODOLOGY

One of the main problems in cloud computing environment is the task-scheduling problem. The task-scheduling problem is mainly concerned about the mapping of application tasks and computing resources in order to achieve the balanced work load and efficient execution of application tasks using the limited resources. There are different task-scheduling algorithms that can be adopted, but suitable to the situation. There were many surveys done on task scheduling

[24, 25, 26]. Selecting the best scheduling policy is the prime concern. Based on this scheduling policy, the application tasks can be mapped to the computing resources and then executed. The scheduling goal assumed in the proposed algorithm KPSOW is the minimization of task completion time. KPSOW makes use of the strengths of both the K-means and PSO algorithms with the inclusion of weights concept. The working behavior of KPSOW is shown in Fig. 1.

The basic idea of proposed work is separating the cloud tasks into low complexity tasks and high complexity tasks and assigning low complexity tasks to low performance computing resources and high complexity tasks to high performance computing resources, thereby makespan of the scheduling tasks can be reduced. Let us consider there are N number of tasks  $T_1, T_2, T_3, \dots, T_N$  and M number of computing resources  $VM_1, VM_2, VM_3, \dots, VM_M$  in a cloud. Here the tasks can be considered as cloud tasks and computing resources can be considered as virtual machines (VMs). Let the lengths of tasks be  $TL_1, TL_2, TL_3, \dots, TL_N$ , and performances of virtual machines be  $VMP_1, VMP_2, VMP_3, \dots, VMP_M$ . By using the length of each task, total cloud tasks are separated into two separate groups by calculating the Euclidian distance between them. K-means algorithm is used for separating the cloud tasks into groups. Let the generated groups be C1 & C2 and number of tasks in each group be nc1 and nc2. Later find out the weights for each cluster using the equation (4) equation (5).

$$WC1 = \sum_{i=1}^{nc1} TL_i \quad (4)$$

$$WC2 = \sum_{i=1}^{nc2} TL_i \quad (5)$$

where WC1 → weight of cluster1  
WC2 → weight of cluster2

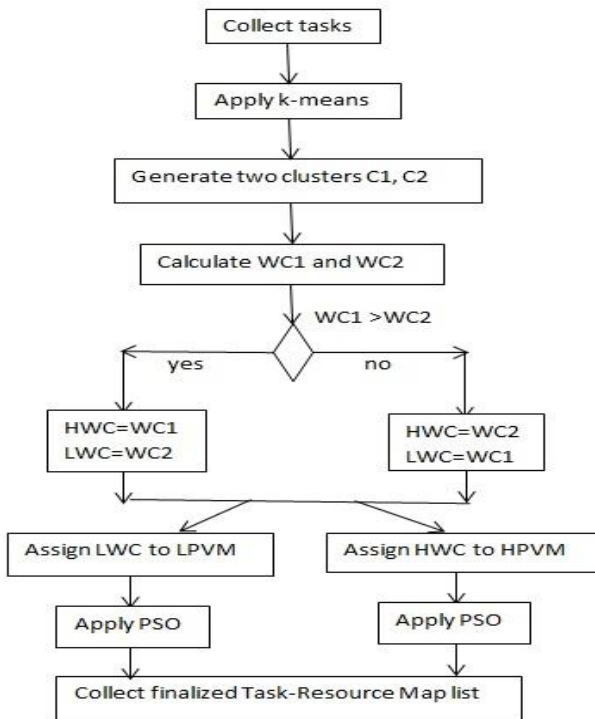


Fig. 1. Working Nature of Proposed KPSOW.

Now compare the weights generated from equation (4) & (5) and assign low weight value to Light Weight Cluster (LWC) variable and high weight value to Heavy Weight Cluster (LWC) variable. LWC represents the low complexity tasks group and HWC represents the high complexity tasks group. Next step is assigning low complexity tasks group to low performance VM (LPVM which are low VMP machines) and high complexity tasks group to high performance VM (HPVM which are high VMP machines). At last, do schedule tasks from LWC group to LPVM and HWC group to HPVM by minimizing the makespan using the algorithm PSO. PSO has been implemented as explained in the section4. After running the PSO algorithm, the final Task-Resource Map list is collected. The final Task-Resource Map list is considered as the best scheduling solution for minimizing the makespan. Makespan is calculated by using the equation (6).

$$makespan = \sum_{TL \in VM_1} \frac{TL}{VMP_1} + \sum_{TL \in VM_2} \frac{TL}{VMP_2} + \dots + \sum_{TL \in VM_M} \frac{TL}{VMP_M} \quad (6)$$

where  $VM_1 \rightarrow 1^{st}$  Virtual machine  
 $VM_M \rightarrow M^{th}$  Virtual machine  
 $VMP_1 \rightarrow 1^{st}$  Virtual machine performance  
 $VMP_M \rightarrow M^{th}$  Virtual machine performance

The objective of proposed method is to find out the minimum makespan when cloud tasks are executed by the virtual machines in a cloud.

## VI. EXPERIMENTAL RESULTS

Cloudsim simulation tool has been used for evaluating the performance of proposed KPSOW method. Simulation has been performed with a total of 5 virtual machines. These 5 virtual machines are grouped into two categories based on their performances. Assume that first three virtual machines ( $VM_1, VM_2, VM_3$ ) are considered as low performance VMs and the last two virtual machines ( $VM_4, VM_5$ ) are considered as high performance VMs. The constraint that is considered for low performance VMs is  $VMP_1 < VMP_2 < VMP_3$  and for high performance VMs is  $VMP_4 < VMP_5$ . The constraint makes sure that less number of tasks is allocated to low performance virtual machines and more number of tasks is allocated to high performance virtual machines. Virtual machine performances have been taken in the range 500 to 600 MIPS and 1100 to 1300 MIPS for low performance VMs and high performance VMs respectively. Cloud task lengths are taken randomly in between 500 to 1000 MIPS. The proposed KPSOW method has been run for 50, 100, 150, 200 cloud tasks separately with all five virtual machines and makespan is compared with the existing methodologies FCFS (First come First Serve) and PSO (Particle Swarm Optimization). Comparison of makespan is shown in Table I and Fig. 2. Results show that KPSOW just took 12.64 sec to schedule 50 cloud tasks where as FCFS and PSO took 15.23 and 13.35 sec respectively. Similarly KPSOW took 23.45 sec to schedule 100 cloud tasks where as FCFS and PSO took 32.07 and 24.33 sec respectively. To schedule 150 cloud tasks, KPSOW took 33.23 sec where as FCFS and PSO took 45.25 and 40.59 sec respectively. At last KPSOW took 38.6 sec to schedule 200 cloud tasks where as FCFS and PSO took 57.47

and 47.81 sec respectively. The results show that KPSOW has done well in reducing the makespan.

The proposed paper has also tested the VM utilization percentage against the above said methodologies for all 50, 100, 150, 200 tasks separately. VM utilization percentage has been calculated using the equation (7).

$$P_{Vi} = \frac{t_{vi}}{T} * 100 \tag{7}$$

Where

$P_{Vi}$  → represents the utilization percentage of  $i^{th}$  virtual machine

$t_{vi}$  → represents the total tasks distributed to  $i^{th}$  virtual machine.

$T$  → total tasks considered

Comparison of VM utilization in percentages is shown in Tables II to V and Fig. 3 to 6.

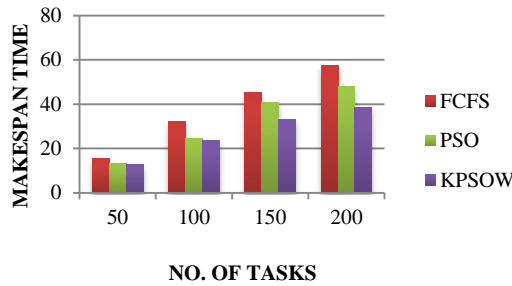


Fig. 2. Comparison of Makespan.

TABLE I. MAKESPAN COMPARISON VALUES

Techniques	No. of Tasks			
	50	100	150	200
FCFS	15.23	32.07	45.25	57.47
PSO	13.35	24.33	40.59	47.81
Proposed KPSOW	12.64	23.45	33.23	38.6

TABLE II. VM UTILIZATION IN % FOR 50 TASKS

Techniques	VIRTUAL MACHINE NUMBER				
	VM1	VM2	VM3	VM4	VM5
FCFS	20.0	20.0	20.0	20.0	20.0
PSO	26.0	16.0	10.0	4.0	44.0
Proposed KPSOW	16.0	28.0	56.0	24.0	76.0

TABLE III. VM UTILIZATION IN % FOR 100 TASKS

Techniques	VIRTUAL MACHINE NUMBER				
	VM1	VM2	VM3	VM4	VM5
FCFS	20.0	20.0	20.0	20.0	20.0
PSO	24.0	8.0	17.0	23.0	28.0
Proposed KPSOW	32.07	30.18	37.77	36.17	63.82

TABLE IV. VM UTILIZATION IN % FOR 150 TASKS

Techniques	VIRTUAL MACHINE NUMBER				
	VM1	VM2	VM3	VM4	VM5
FCFS	20.0	20.0	20.0	20.0	20.0
PSO	26.66	11.33	16	14.66	31.33
Proposed KPSOW	26.25	32.5	41.25	27.14	72.85

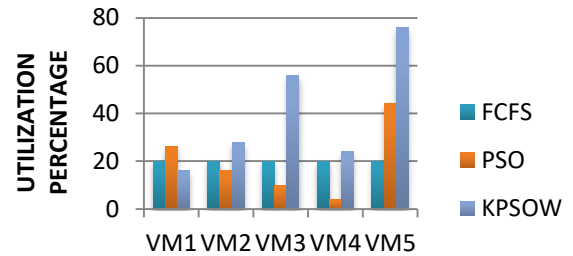


Fig. 3. Comparison of VM Utilization % for 50 Tasks.

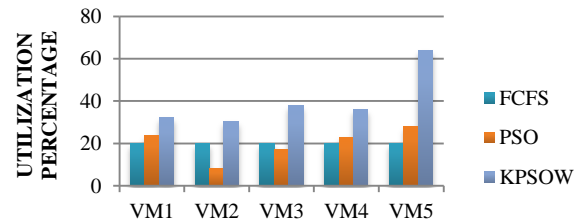


Fig. 4. Comparison of VM Utilization for 100 Tasks.

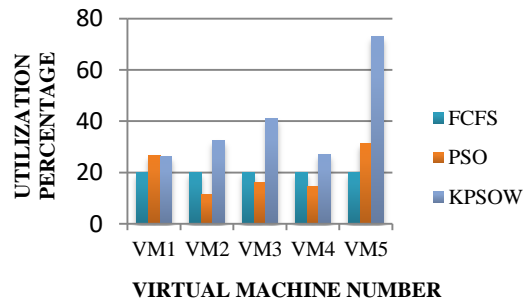


Fig. 5. Comparison of VM Utilization for 150 Tasks.

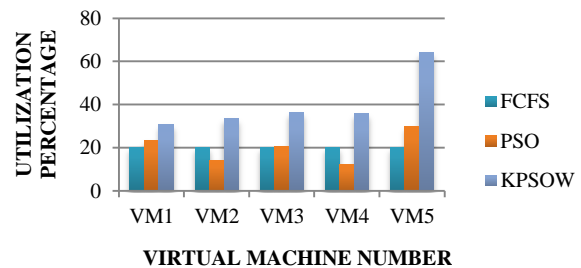


Fig. 6. Comparison of VM Utilization for 200 Tasks.

TABLE V. VM UTILIZATION IN % FOR 200 TASKS

Techniques	VIRTUAL MACHINE NUMBER				
	VM1	VM2	VM3	VM4	VM5
FCFS	20.0	20.0	20.0	20.0	20.0
PSO	23.5	14.0	20.5	12.0	30.0
Proposed KPSOW	30.55	33.33	36.11	35.86	64.13

Utilization figures state that KPSOW algorithm has utilized virtual machines with proper load balancing. The proposed algorithm has followed the initial constraint assumption. Results show that the number of tasks allocated to VM<sub>3</sub> is greater than VM<sub>2</sub> and VM<sub>2</sub> is greater than VM<sub>1</sub> in LPVMs & number of tasks allocated to VM<sub>5</sub> is greater than VM<sub>4</sub> in HPVMs for proposed KPSOW algorithm. PSO has better utilization ratio compared to K-means. So the proposed methodology effectively allocates less number of tasks to low performance virtual machines and more number of tasks to high performance virtual machines. Hence, KPSOW has given better utilization ratio for all virtual machines compared to FCFS and PSO methodologies.

### VII. CONCLUSION

Task scheduling is an effective way of scheduling for getting tasks executed faster in cloud environment. Scheduling tasks with a clustering approach is one of the ways of ordering jobs in cloud environment for effective execution. The proposed KPSOW algorithm is the one such method. KPSOW has effectively used cluster weights concept for forming low complexity and high complexity task groups. KPSOW can be used in a cloud when there is more number of cloud tasks with varied complexities. Results are stating that KPSOW has made remarkable changes in reducing the makespan and also improves the utilization of computational resources when compared to the earlier methodologies. KPSOW shows simplicity in executing complex tasks in a speedy way and will be helpful to the society where the reliable cloud environment is required.

### REFERENCES

[1] M. Kashani and M. Jahanshahi, "Using simulated annealing for task scheduling in distributed systems," in Computational Intelligence, Modelling and Simulation, 2009. CSSim '09. International Conference on, sept. 2009, pp. 265–269.

[2] C.N. Hoefler and G. Karagianni, "Taxonomy of Cloud Computing Services", IEEE 2010.

[3] F. Baroncelli, B. Martini and P. Castoldi, "Network virtualization for cloud computing", annals of telecommunications-Annales des télécommunications, vol. 65, (2010), pp. 713-721.

[4] T. Dillon, C. Wu, and E. Chang, "Cloud computing: issues and challenges", in Advanced Information Networking and Applications (AINA), 2010 24th IEEE International Conference on, (2010), pp. 27-33.

[5] Maricela-Georgiana Avram (Olaru), "Advantages and Challenges of Adopting Cloud Computing from an Enterprise Perspective," The 7<sup>th</sup> International Conference Interdisciplinary in Engineering (INTERENG 2013).

[6] Chaima Ghribi, Makhlof Hadji and Djamal Zeghlache, "Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms", In 13th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid), IEEE, pp. 671–678, (2013).

[7] R. Lee and B. Jeng, "Load-balancing tactics in cloud", in Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2011 International Conference on, (2011), pp. 447-454.

[8] N. J. Kansal and I. Chana, "Cloud load balancing techniques: A step towards green computing", International Journal of Computer Science Issues, vol. 9, no. 1, pp. 238–246, 2012.

[9] Ramezani, F., Lu, J., Hussain, K., "Task-Based System Load Balancing in Cloud Computing Using Particle Swarm Optimization", Journal of Parallel Programming 42, 5, 739-754, 2014.

[10] Priyansh Srivastava, Bhavesh Gohil, and Dhiren Patel, "Load Management Model for Cloud Computing Using Cloudsim", International Journal of Computer Theory and Engineering, Vol. 9, No. 5, October 2017.

[11] Chenhong Zhao, Shanshan Zhang, Qingfeng Liu, Jian Xie, Jicheng Hu, "Independent tasks scheduling based on Genetic Algorithm in Cloud Computing", 978-1-4244-3693-4/09/\$25.00 2009 IEEE.

[12] GAN Guo-ning, HUANG Ting-lei, GAO Shuai, "Genetic Simulated Annealing Algorithm for task scheduling based on Cloud Computing Environment", 978-1-4244-6837-9/10/\$26.00 2010 IEEE.

[13] S. Pandey, L. Wu, S. Mayura Guru, R. Buyya, "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments," 24th IEEE international conference on advanced information networking and applications, PP 400-407, 2010.

[14] Lizheng Guo, Guojin Shao, Shuguang Zhao, "Multi-Objective Task Assignment in Cloud Computing by Particle Swarm Optimization", 978-1-61284-683-5/12/\$31.00 2012 IEEE.

[15] Ali Al-maamari and Fatma A. Omara, "Task Scheduling Using PSO Algorithm in Cloud Computing Environments", International Journal of Grid Distribution Computing Vol. 8, No.5, (2015), pp.245-256.

[16] Masdari, M., Salehi, F., Jalali, M., Bidaki, M., 2016. "A Survey of PSO-Based Scheduling Algorithms in Cloud Computing", Journal of Network and Systems Management 25, 1, 1-37.

[17] K. Li, G. Xu, G. Zhao, Y. Dong, and D. Wang, "Cloud task scheduling based on load balancing ant colony optimization," in Chinagrid Conference (ChinaGrid), 2011 Sixth Annual, aug. 2011, pp. 3–9.

[18] X. Lu and Z. Gu, "A load-adaptive cloud resource scheduling model based on ant colony algorithm", in Cloud Computing and Intelligence Systems (CCIS), 2011 IEEE International Conference on, (2011), pp. 296-300.

[19] Khan, S., Ahmad, A., "Cluster centre initialization algorithm for kmeans clustering", Pattern Recognition Lett. 25, 1293–1302, 2004.

[20] A M Fahim, A M Salem, F A Torkey, "An efficient enhanced k-means clustering algorithm", Journal of Zhejiang University Science A, vol. 10, pp. 1626-1633, July 2006.

[21] Lloyd, S. P. (1957). "Least square quantization in PCM". Bell Telephone Laboratories Paper. Published in journal much later: Lloyd, S. P. (1982). "Least squares quantization in PCM" (PDF). IEEE Transactions on Information Theory. 28 (2): 129–137.

[22] E.W. Forgy (1965). "Cluster analysis of multivariate data: efficiency versus interpretability of classifications". Biometrics. 21 (3): 768–769. JSTOR 2528559.

[23] Ms. Rukhsar Khan, Mr. Gaurav Kakhani, "Analysis of Priority Scheduling Algorithm on the basis of FCFS & SJF for Similar Priority Jobs", International Journal of Computer Science and Mobile Computing, Vol.4 Issue.9, September- 2015, pg. 324-331.

[24] Raja Manish Singh, Sanchita Paul, Abhishek Kumar, "Task Scheduling in Cloud Computing: Review", (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7940-7944.

[25] P.Nagendra Babu1, M.Chaitanya Kumari, S.Venkat Mohan, "A Literature Survey on Cloud Computing", International Journal of Engineering Trends and Technology (IJETT) – Volume 21 Number 6 – March 2015.

[26] P. Akilandeswari and H. Srimathi, "Survey and Analysis on Task Scheduling in Cloud Environment", Indian Journal of Science and Technology, Vol 9(37), DOI: 10.17485/ijst/2016/v9i37/102058, October 2016.