Volume 6, Issue 1, January 2019 ISSN: 2348-8565 (Online)

IJMERT

International Journal of Modern Engineering and Research Technology

Website: http://www.ijmert.org Email: editor.ijmert@gmail.com

Cloud Resource Cost Minimization using PSO Algorithm

Garima Thakur
M. Tech Research Scholar
Department of Mathematics,
Shri Ram Group of Institutions
Jabalpur, (M.P.) [INDIA]
Email: singhgarima6389@gmail.com

Jayesh Jain
Assistant Professor,
Head of the Department,
Department of Computer Science & Engineering
Shri Ram Group of Institution Jabalpur
Email: Jayeshjain.it@gmail.com

Sapna Choudhary
Assistant Professor,
Head of the Department,
Department of Computer Science & Engineering
Shri Ram Group of Institution Jabalpur
Email: choudharysapnajain@gmail.com

ABSTRACT

In the current scenario, Cloud computing carved itself as an emerging technology which enables the organization to utilize hardware, software and applications without any upfront cost over the internet. A very efficient computing environment is provided by cloud computing where the customers or several tenants are in need of multiple resources to be provided as a service over the internet. The challenge before the cloud service provider is, how efficiently and effectively the underlying computing resources like virtual machines, network, storage units, and bandwidth etc. should be managed so that no computing device is in under-utilization or over-utilization state in a dynamic environment. A good task scheduling technique is always required for the dynamic allocation of the task to avoid such a situation. A Particle Swarm Optimization (PSO) package is integrated in our simulator so as to achieve an effective result where PSO will randomly find the suitable Physical host in heterogeneous environment so as to transfer the load.

Through this paper we are going to present the new Algorithm based on task scheduling technique,. A comparison of this proposed Algorithm is performed on our simulator which shows that, this will outperform the existing techniques like EFT.

Keywords:— *Cloud Computing, Task Scheduling, Resource Optimization, EFT, PSO, QoS.*

I. INTRODUCTION

Cloud computing has become an influential architecture to perform large-scale and complex computing. It facilitates convenient on-demand network access to a shared pool of configurable computing resources. These resources are rapidly provisioned and released with minimal management effort or service provider interaction. Flexibility and scalability are the most critical properties of cloud environments, which make it a highly complex and large distributed environment. Moreover, cloud computing provides facilities for processing massive amounts of data. As millions of users submit their computing tasks to the cloud system, task scheduling mechanism forms a vital role in cloud computing environments.

Cloud computing at the present is an open research field which has not been explored much. Cloud computing is a comprehensive solution that delivers IT as a service. It is an internet based computing solution where resources are shared. Cloud computing provides computation, software applications, data access, data management and storage resources without requiring cloud users to know the location and other details of the computing infrastructure. End users access cloud based applications through a web browser or a light weight desktop or mobile app while the business software and data are stored on servers at a remote location. Cloud application providers strive to give the same or better service and performance than if the software programs were.

Task scheduling refers to a set of techniques for assigning a group of tasks to the available virtual machines (VMs). The main challenge of the task scheduling mechanism is to increase resource utilization without affecting the quality of services. Task scheduling algorithms are categorized into deterministic and metaheuristics algorithms. Deterministic scheduling algorithms include traditional algorithms such as First Come First Served (FCFS) [1], Round Robin (RR) [2], Shortest Job First (SJF) [3] and Load Balance over Slow Resources (LBSR) [4]. These algorithms are the basic algorithms for scheduling various tasks. However, they usually cannot find the optimal solution in a reasonable time, especially when the problem complexity increases. Metaheuristics algorithms include Hill Climbing (HC) [5], Simulated Annealing (SA) [6], Tabu Search (TS) [7], Ant Colony Optimization (ACO), Cuckoo Search (CS) and Particle Swarm Optimization (PSO). These algorithms are based on swarm intelligence, and they are used to find an approximated solution. This could be useful when the exact solution is too costly to obtain using deterministic algorithms. PSO algorithm is one of the most popular algorithms that are used to solve optimization problems in task scheduling processes. However, scheduling a large number of tasks using PSO algorithm can reduce the system performance. Moreover, most of the algorithms that are based on PSO either is based on modifying the PSO algorithm or based on working with the PSO (hybrid) produced results to schedule a limited number of tasks.

In this paper, an enhanced scheduling algorithm is proposed. The algorithm is based on the PSO algorithm and aims at scheduling a large number of tasks without affecting the system performance. The proposed algorithm divides the submitted tasks into a number of batches in a balanced and dynamic way. It considers two parameters which are the number of tasks and the total length of tasks per each batch. Then, it allocates a sub-optimal solution for each batch. Finally, sub-optimal solutions are appended to find the final allocation plan.

1.1 Scheduling in Cloud Computing

Task scheduling is a procedure used to allocate incoming tasks to the available resources. The main goal of tasks scheduling algorithms is to maximize the resources utilization without affecting the service parameters of the cloud. The basic scheduling process consists of three processes which are done in the cloud environment. The first process is the information providing process, in which the task scheduler collects task information and resources information from the task manager and the resource manager. The second process is a selection process, in which the target resource is selected based on specific parameters of the resource and the task. These parameters include task size, task priority, reliability factor, activity-based cost, and dynamic slotted length of the tasks. Then, the task scheduler sends the task allocation plan to the resource manager. The task distribution is the final process. In this process, the task manager allocates each task to the appropriate resources.

As expected, scheduling a large number of tasks causes computing efficiency degradation. That leads to long makespan, long wait times, and increased costs. Therefore, the problem of optimal allocation of a large number of tasks to the available VMs is a great challenge. Scheduling in cloud computing is considered as a problem with a large solution space. Thus, it takes a long time to find an optimal solution. Deterministic scheduling algorithms are much easier and faster to implement because they are all designed based on one or a few rules for managing and arranging the tasks. However, these algorithms cannot find the optimal solution in a reasonable time, especially when the problem becomes complex or the number of tasks is too large. In contrast, metaheuristics-based techniques have been proved to achieve near-optimal solutions within a reasonable time for such problems.

Metaheuristics-based algorithms generally provide better results than deterministic algorithms in terms of quality [16]. Therefore, metaheuristics have great popularity due to its efficiency and effectiveness. Metaheuristics task scheduling algorithm has two types namely; single solution-based metaheuristics, and population-based metaheuristics. Single solution algorithms focus on modifying and improving a single candidate solution such as HC, SA, and TS. On the other hand, population-based algorithms maintain multiple candidate solutions. Therefore, these algorithms use population characteristics to guide the search. Metaheuristics generally are slower than deterministic algorithms. That is why the recent trend in some research areas tries to pre-process the input data to accelerate the execution time of metaheuristics algorithms. Recent research that works on metaheuristics algorithms usually focuses on three main goals: modifying the fitness function, modifying the operators, and hybrid metaheuristics. Research aiming at adjusting the fitness function [8] is based on redesigning or adding the fitness function to fit better with the cloud environment. Whereas, modifying the operators usually involves redesigning or adding the transition operators inspired by other scheduling algorithms. Finally, research that considers hybrid metaheuristics aims at using different scheduling algorithms to enhance the performance of the original algorithm [9].

Giving the fact that PSO is an evolutionary computational technique motivated by the social behavior of the particles; the PSO algorithm does not require any gradient information of the optimization function. Moreover, it only utilizes primitive mathematical operators and few parameters. In PSO, each member of the population is called a particle, and the population is called a swarm. Starting with a randomly initialized population and moving in randomly chosen directions, each particle goes through the searching space and remembers the best previous positions of itself and its neighbors. Finally, all particles tend to fly towards better positions over the searching process until the swarm moves close to an optimum position. PSO algorithm suffers from partial optimism, which degrades the regulation of its speed and direction. When the PSO search space expands, the chance of finding an improved optimal solution becomes harder [10].

A Cloud provider first constructs a computing system called cloud. In this we have several virtual machines interconnected and the provider processes the task of the users. "Cloud computing is not a well behaved model for providing wanted, user- required, flexible access to a shared pool of configurable computing resources that can be quickly provided and released with low care effort or service are going to study the divisible task scheduling of high performance computing algorithms" [18]. Cloud computing environment where multiple virtual machines (VMs) can share physical resources (CPU, memory, and bandwidth) on a single physical host and multiple VMs can share the bandwidth of a data center by using network virtualization. Because many users and applications essentially share system resources, a proper task-scheduling scheme is difficult to resource utilization and system performance.

II. Review of Literature

2.1 Task Scheduling Approaches

There are different scheduling algorithms available for cloud computing. In this paper we will discuss about five scheduling algorithms they are First Come First Serve (FCFS), Round Robin (RR), Genetic algorithm, Match-making algorithm and generalized priority algorithm.

A. First come first serve (FCFS) Algorithm.

Given n processes with their burst times, the task is to find average waiting time and average turn around time using FCFS scheduling algorithm. First in, first out (FIFO), also known as first come, first served (FCFS), is the simplest scheduling algorithm. FIFO simply queues processes in the order that they arrive in the ready queue.

In this, the process that comes first will be executed first and next process starts only after the previous gets fully executed. Here we are considering that arrival time for all processes is 0.

How to compute below times in Round Robin using a program?

- ***Completion Time:*** Time at which process completes its execution.
- ***Turn Around Time:*** Time Difference between completion time and arrival time. Turn Around Time = Completion Time – Arrival Time
- ***Waiting Time(W.T):*** Time Difference between turn around time and burst time. Waiting Time = Turn Around Time – Burst Time

B. Round Robin (RR) Algorithm.

Round Robin is a CPU scheduling algorithm where each process is assigned a fixed time slot in a cyclic way.

- It is simple, easy to implement, and starvation-free as all processes get fair share of CPU.
- One of the most commonly used technique in CPU scheduling as a core.
- It is preemptive as processes are assigned CPU only for a fixed slice of time at most.

The disadvantage of it is more overhead of context switching.

C. Genetic Algorithm.

Genetic Algorithms(GAs) are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. Genetic algorithms are based on the ideas of natural selection and genetics. These are intelligent exploitation of random search provided with historical data to direct the search into the region of better performance in solution space. They are commonly used to generate high-quality solutions for optimization problems and search problems.

Genetic algorithms simulate the process of natural selection which means those species who can adapt to changes in their environment are able to survive and reproduce and go to next generation. In simple words, they simulate "survival of the fittest" among individual of consecutive generation for solving a problem. Each generation consist of a population of individuals and each individual represents a point in search space and possible solution. Each individual is represented as a string of character/integer/float/bits. This string is analogous to the Chromosome.

D. Particle Swarm Optimization:

PSO consists of a swarm of particles; each of which represents a solution of the problem. In this work, the particles are represented as a set of VMs that are allocated for the tasks. Every particle in the swarm behavior has two main characters: a position x that notates the suggested location and a velocity v that means the speed of moving. The best solution for one p article is called the best personal experience (pbest), while the best solution among all particles in the population is called the best group experience (gbest) solution. The position of a particle at any instant in time is influenced by its personal best position (pbest) and the position of the other best particles in the global problem space (gbest). In more detail, PSO uses adaptive movement, which shifts a particle's position in each iteration. The way in which PSO updates particle xi at generation t is shown in equation 1.

(Kennedy and Eberhart [25]):
$xi\ (t) = xi\ (t - 1) + vi\ (t)$ .... (1)

where: xi (t) is the current position of particle i at iteration t.

xi (t-1) is the position of particle i at iteration t-1.

vi (t) is the velocity of particle i at iteration t .

The velocity of particle i at time t is given in equation 2 (Kennedy and Eberhart [12]):

$vi\ (t) = w \times vi\ (t - 1) + r1 \times C1 \times (pbesti - xi) + r2 \times C2 \times (gbesti - xi\ (t))$... ....(2)

In equation 2, w is used to control the movement of the particle and improve the convergence of PSO. The numbers r1 and r2 are used to provide randomness in the movement of particles inside the swarm in the range of [0-1] to control the movement of the particles in the search space. Both C1 and C2 are positive constants, which are specific parameters to control the effect of the personal and global best particles. Their values are set to fixed equal values because the particle is influenced by its own best position (pbest) and the best position of its neighbors (gbest) [11].

Particle swarm optimization (PSO) is a unique technique of an artificial intelligence in order to solve optimization problems which are complex in nature. This technique is basically inspired by the social animal's collective behavior such as bird flocks. Therefore, way to gather the information of load of every node and way to use measurement of PSO and local agent's evaluation to schedule resource is the main aim in clouding system. In order to tackle above problems we have integrated Particle Swarm Optimization (PSO) package in our environment where PSO will find the VM according to suitability on the random basis so as to transfer the load. It is a basic method that gives to solve an optimized problem by repeating trying to enhance a solution. PSO solves an issue by having different number of solutions, which are also called as particles and movement these particles around in the search-space according to simple mathematical formulae over the particle's position as well as velocity. The movement of each particle is known through its local best known position and is also known as to find the best known positions in the search- space, these are updated on basis of better positions and then they are located by other particles. It is expected to give the best particle known as swarm toward the best solutions. The figure 1 below shows the concept of pbest and gbest in PSO.

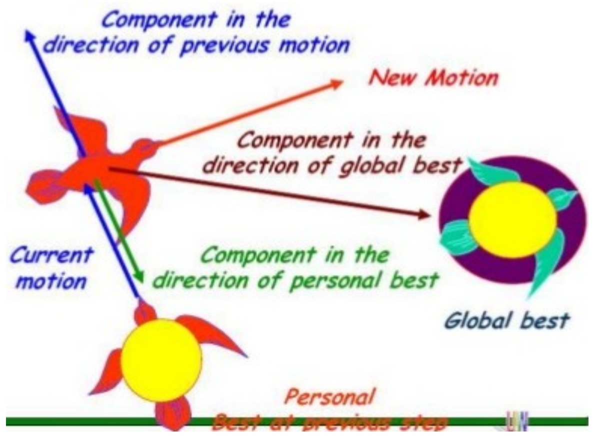

Figure 1: Pbest and Gbest in PSO.

2.2 PSO Variants in Task Scheduling

Zhan et al [12] has proposed an algorithm, hybridization of Particle Swarm Optimizations and Simulated Annealing. In this algorithm SA is used to optimize the particle by the jumping ability of SA. The better swarm is obtained by the fast searching ability of PSO.Outcomes of this algorithm shows that the combined effect of SA and PSO reduces the average operation time of the tasks and increases the utilization ratio of resources. Ali et al. [13] has proposed an algorithm Dynamic adaptive Particle swarm optimization (DAPSO) to solve the PSO affinity problem that helps in fast convergence. The drawback of DAPSO, the inertia weight once decreased cannot be increased to search new regions and not able to recover its exploration mode. To defeat this he additionally proposed MDAPSO a combination of DAPSO and Cuckoo search (CS) algorithms that performs better than the original PSO and provides better execution regard to makespan and resource usage. MDAPSO obtains a balance between local and global exploration and reaches an optimum solution. Rana et al. [14] stated that PSO provides optimised results in terms of execution, accuracy and efficiency when compared to other evolutionary algorithms like GA, ACO and SA. The review of this paper provides better clustering results that helps in better predictions and analysis of data. Uma et al. [15] has proposed a hybrid approach of GA and PSO algorithm .In this approach they introduced a inertia weight that was dynamically changed in PSO that balance the global and local exploration and provides fast convergence Awad et al[16] has proposed a Load Balancing Mutation (balancing) Particle Swarm Optimisation. They presented an algorithm to improve the reliability and the task that was failed to allocate was rescheduled to the available resources. Bansal et al. [17] analysed the inertia weight based on three criteria and stated that chaotic inertia weight is the best to obtain better precise in PSO.

Zhang et al. [18] stated that a heuristics approach based on particle swarm optimisation can give better schedule for large scale optimisation problems than genetic schedule Priyadharshini et al. [19] proposed an algorithm based on Parallel Bee Colony Optimisation Partcle Swarm Optimisation (PBCOPSO). In hybrid Parallel approach, parallel execution of BCO and PSO provides optimised makespan and better utilisation of resources. PBCOPSO showed an improved resource utilisation when compared to Min

–Min algorithm by 5.0383% and Improved BCO by 3.7243%. Abdi et al. [20] stated that uncertainty reduces the possibility of algorithm to convert to best solution and they have generated the initial population randomly by considering the shortest job to fastest processor algorithms (SJFP). By merging SJFP in standard PSO, the results of this algorithm reduces the makespan compared to PSO and GA. Wang et al. [21] has proposed a self adaptive learning based PSO (SLPSO). SLPSO uses four searching strategies that improves the universality of PSO Variants.

Pooranian et al [22] has proposed an algorithm for grid environment. They combined PSO and gravitational emulation search (PSO- GELS) to solve task scheduling problem. The position vector values are changed from continuous to discrete by rounding the real values. Chaotic activity is used to schedule the tasks. Since PSO cannot search locally, by Combining GELS algorithm the hybrid algorithm avoids local optima. SO-GELS algorithm reduces the makespan by 29.2% compared with Simulated Annealing algorithm.

Sivanandam et al [23] stated that a hybrid heuristic mode combining Particle Swarm Optimisation with Simulated Annealing and dynamically reducing inertia value that gives near optimal solutions compared to fixed inertia. Izhakian et al [24] proposed a Discrete Particle Swarm Optimisation (DPSO) algorithm that provides feasible solutions to reduce makespan and flow time simultaneously. Job allocation is represented by a 2D matrix. Assumptions made are jobs are independent and premption not allowed. Results obtained show that performance of DPSO is better than fuzzy PSO.

Zuo et al [25] introduced four modified velocity updating strategies to improve the global search ability and robustness. The first strategy updates the velocity based on different particle’s information .The second strategy helps in quick convergence by learning the global information. The third strategy is for all dimensions they select the particle's pbest. The fourth strategy randomly selects the particle's pbest so that the algorithm searches a smaller region to reduce the complexity. They proposed a Self -adaptive learning of PSO (SLPSO), a task represented by particle's dimension (position). The task that has higher priority (position with larger value) is assigned in the scheduling process first. Compared with standard PSO experimental results show that SLPSO provides a better schedule in reduced time for large problem size.

III. PROPOSED WORK

Users access cloud computing using networked client devices, such as desktop computers, laptops, tablets and Smartphone. Some of these devices - cloud clients - rely on cloud computing for all or a majority of their applications so as to be essentially useless without it. Examples are thin clients and the browser based Chrome book. Many cloud applications do not require specific software on the client and instead use a web browser to interact with the cloud application. A client can send a request to the server asking for any resource at any point of time. Allocation and scheduling of the resources in an efficient manner is one of the challenges in cloud computing. The scheduling algorithm chosen for the project is Best Resource Selection (BRS) algorithm. In BRS, work schedule tasks on resources based on the earliest finish time. The resources are selected based on the performance. In general cases mostly the resource scheduling is done but not the optimization of those resources. By applying the optimization of scheduled resources it is found that the cost can be reduced significantly. In the proposed method the resources are scheduled are optimized by using the PSO technique. The task scheduling is done by the server using the Best Resource Selection Algorithm or Earliest Finish Time. After the completion of this scheduling particle Swarm Optimization technique is applied on resources of scheduled tasks. The price calculation system depends on the environmental utilization and the time required by the task for its execution in the server. The price calculation system uses the defined amount of cost for each VM types defined by the server. Different types of defined VM types have different costs these costs are defined by the size of memory allocated to the defined VM types. Once the task completes its execution the cost of the task is calculated depending upon the VM type requested by the task and the amount of time it spent in the server for execution.

3.1 Proposed Algorithm

The scheduling algorithm is as follows:

Proposed Scheduling PSO (TaskList, VMs)

Initialize the RAM size, processing elements and disk space as defined.

repeat

Set the node T as the request from the ready queue

allocate the resources of T by deducting from the maximum limit

for all requests in the ready queue do

Compare the resource utilization time of the node T and other requests

Set T as the request which has shorter time

Set the other requests in the waiting queue

end for

Once the time is over for the request T, de allocates the resources.

Change the next request from the waiting queue to the ready queue

Update the ready queue list

until there are unscheduled requests.

3.2 Proposed Algorithm

In proposed system input is number of tasks and system will allocate virtual machines to these tasks. If numbers of tasks is less than or equals to numbers of virtual machines than simply assign task to available virtual machines otherwise perform task scheduling using PSO.

Algorithm Proposed Main (task, VMs)

{

TaskList=Empty

Assign incoming task to taskList as per their arrival time

TaskList={x1, x2…xn}

m=number of VMs

n=number of tasks

Do

If n>m than

Call ProposedSchedulingPSO (TaskList, VMs)

Else

Assign all tasks from TaskList to VMs.

End If

While (TaskList is not empty)

}

IV. Results and Discussion

In general cases mostly the resource scheduling is done but not the optimization of those resources. In the proposed method the resources are scheduled optimally by using the PSO technique. The user and provider are benefited mutually as the resources are not being wasted and the cost for the user is reduced significantly.

Table 1: Comparison of Costs in Existing and Proposed System will be Shown for Separate Requests:

Request	Client ID	Type of resource	Cost using BRS(in RS)	Cost using PSO on BRS(in RS)
1	1	m1.small	260	200
2	1	m1.large	990	660
3	1	m1.large	900	940
4	1	m1.large	1980	1060
5	1	m1.xlarge	1700	1220
6	1	c1.medium	2300	2300
Total			**8130**	**6420**

Evaluation between existing and proposed method will be shown graphically in chart below:

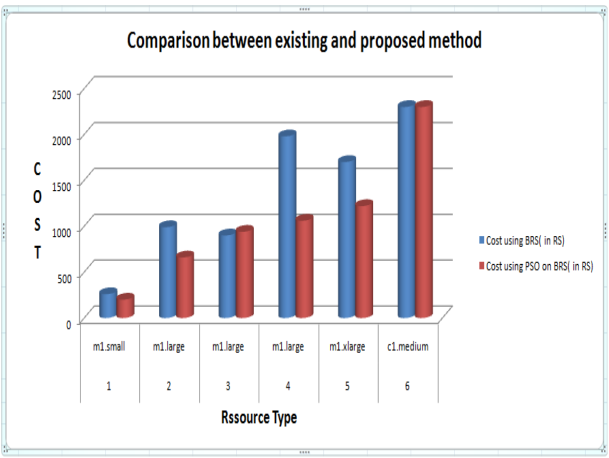

Figure 2: Comparison Between Existing and Proposed Method chart.

V. Conclusion

Issue of task scheduling is one of the main issues for cloud computing. Optimal task scheduling is essential for allocating resources, reducing the execution time, increasing throughput and decreasing the cost. Proposed work is an enhanced task scheduling algorithm for the cloud computing environment. Resource allocation problem has been an important topic in the fields of networking, parallel and distributed computing. Cloud computing enables enterprises to reserve group of resources, use them to establish common platform and run computation tasks. In this thesis we have worked on the issue of optimizing the scheduling of resources deciding how many and what kind of resources need to be reserved for these computations after the computations have been linearized into the form of a sequence of tasks. This algorithm is implemented on server (cloud) system to reduce time as well as cost of the processes and to allow a higher degree of resource sharing. We compare the cost savings when using PSO and existing 'Best Resource Selection' (EFT) algorithm. Results show that PSO can achieve very good cost savings as compared to BRS, and a good distribution of workload onto resources. The experimental results prove the efficiency of the proposed algorithm by minimizing cost. So, the fairness has been satisfied at all levels. The proposed task scheduling algorithm could be further extended by considering dependent tasks, and also considering dynamic task scheduling.

REFERENCES:

[1] W A. Karthick, E. Ramaraj, and R. Subramanian, "An efficient multi queue job scheduling for cloud computing," in proc. IEEE world congress on computing and

communication technologies (wccct), Trichirappalli, India, pp. 164-166, Mar. 2014.

[2] P. Samal, and P. Mishra, “Analysis of variants in round robin algorithms for load balancing in cloud computing,“ International Journal of Computer Science and Information Technologies (IJCSIT), vol. 4 (3) , pp 416-419, 2013

[3] P. Gupta, and N. Rakesh, “Different job scheduling methodologies for web application and web server in a cloud computing environment,” in Proc. IEEE Third International Conference on Emerging Trends in Engineering and Technology, Goa, India, pp. 569-572, Nov. 2010.

[4] W. Saber, R. Rizk, W. Moussa, and A. Ghonem, "LBSR: Load balance over slow resources," in Proc. International Conference on Computer Applications & Technology (ICCAT), Cairo, Egypt, Jan. 28-29, 2017.

[5] M. Brototi, K. Dasgupta, P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," in Proc. Procedia 2nd International Conference on Computer, Communication, Control and Information Technology (C3IT), vol. 4, pp. 783–789,Feb. 2012.

[6] G. Gan, T. Huang, S. Gao, “Genetic simulated annealing algorithm for task scheduling based on cloud computing environment,” in Proc. IEEE International Conference on Intelligent Computing and Integrated Systems (ICISS), pp. 60–63 ,2010.

[7] P. Yi, H. Ding, and B. Ramamurthy, “A tabu search based heuristic for optimized joint resource allocation and task scheduling in grid/clouds,” in Proc. IEEE International Conference on Advanced Networks and Telecommunications Systems, Kattan kulathur, India, pp. 15–18, Dec. 2013.

[8] Singh, S., Chana, I.: Q-aware: quality of service based cloud resource provisioning. Comput. Electr. Eng. Elsevier. doi:10.1016/j.compeleceng.2015.02.003

[9] Chana, I., Singh, S.: Quality of service and service level agreements for cloud environments: issues and challenges. In: Cloud Computing-Challenges, Limitations and R&D Solutions, pp. 51–72. Springer International Publishing (2014).

[10] Neeta Patil, Deepak Aeloor, “A Review – Different Scheduling Algorithms In Cloud Computing Environment”, IEEE 2017.

[11] Tripathi, P. K., Bandyopadhyay, S., & Pal, S. K. (2007). Multiobjective particle swarm optimization with time variant inertia and acceleration coefficients. Information Sciences, 177(22), 5033-5049.

[12] Shaobin Zhan, Hongying Huo Shenzhen “Improved PSO based Task Scheduling Algorithm in Cloud Computing”, Institute of Information Technology, Shenzhen, China Journal of Information and Computational Science 9:13 (2012) 3821–3829.

[13] Ali Almaamari 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.

[14] Sandeep Rana, Sanjay Jasola, Rajesh Kumar, "A review on Particle Swarm Optimisation algorithms and their applications to data Clustering", Artif Itell Rev(2011) Springer(2010) 35:211 -222.

[15] S. Uma, K. R. Ganhi, E. Kirubakaran, "A hybrid PSO with dynamic inertia weigh and GA approach for discovering classification rule in data mining", International Journal of Computer Applications, Vol.40. (2012).

[16] A.I. Awada, A.El-Hefnawy, H.M. Abdel Kader, "Enhanced Particle Swarm Optimisation Task Scheduling in Cloud Computing Envionments", International Conference on communication, management and Information Tchnology(ICCMIT 2015). Procedia Computer Science.

[17] J.C. Bansal, P.K. Singh, Mukesh Saraswat, Abhishek Verma, Shimpi Singh Jadon, Ajith Abraham," Inertia Weight Strategies in Particle Swarm Optimisation", 978-1-4577-1123-7 IEEE 2011.

[18] Lei Zhang, Yuehui Chen, Runyuan Sun, Shan Jing and Bo Yang, "A Task Scheduling Algorithm Based on PSO for Grid Computing", International Journal of Computational Intelligence Research. ISSN 0973-1873 Vol.4, No.1(2008), pp.37-43.

[19] Jemini Priyadharshini, L. Arockiam, "PBCOPSO: A Parallel Optimisation Algorithm for Task Scheduling in Cloud Environment", Indian Journal of Science and Technology, Vol 8 (16), July 2015.

[20] Solmaz Abdi, Seyyed Ahmad Motamedi, Saeed Sharifian."Task Scheduling using Modified PSO Algorithm in Cloud Computing Environment". International Conference on Machine Learning, Electrical and Mechanical Engineering(ICMLEME '2014) Jan.8-9, 2014. Dubai(UAE).

[21] Wang. Yu., Li Bin., Thomas., Wang Jian yu., Yuan Bo., Tian Qiongjie," Self adaptive learning based particle Swarm Optimisation", Information Science 181,4515-4538.2011.

[22] Pooranian Z, Shojafar M, Abawajy JH, Abraham A."An efficient metatheuristic algorithm for grid Computing", J Comb Optim 2013:1-22.

[23] Sivanandam SN, Visalakshi P, "Multiprocessor Scheduling Using Hybrid Particle Swarm Optimisation wit dynamically- Varying Inertia", Int J Computer Sci Appl 4(3):95-106.

[24] Izakian H. Ladani BT, Zamanifar K, Abraham A."A novel particle swarm optimisation approach for grid job Scheduling", Inf Syst Techno manage, Vol.31.Springer: 2009. p.100-9.

[25] Xingquan Zo, Guoxiang Zhang,Wei Tan, "Self Adaptive Learning PSO-Based Deadline Constrained Task Scheduling for Hybrid Iaas Cloud", IEEE Transactions on Automation Science and Engineering, Vol.11, No.2. April 2014.

* * * * *