IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

An Improved Method for Scheduling of Scientific Workflows

Merly Mathew[1], Jayalekshmi S[2]

PG Scholar, Department of CSE, LBSITW, Trivandrum, India [1]

Associate Professor, Department of CSE, LBSITW, Trivandrum, India [2]

Abstract: Cloud computing is the latest distributed computer paradigm where the resources are delivered over the Internet as services. It follows a pay-as-you-go model where users are charged with respect to their use. A lot of researches are being done on the scheduling in cloud computing and most of them are about workflow and job scheduling. A workflow is defined as a sequence of tasks interconnected via data or computing dependencies. Many scheduling policies have been proposed which aim to maximize the amount of work completed while meeting QoS constraints such as deadline and budget. However many of them are not optimal because they do not incorporate some basic principles of cloud computing such as the elasticity, heterogeneity of the resources. In this paper a meta-heuristic optimization technique, HDPSO is used to minimize the execution cost. Other parameters like makespan, execution time are considered within heterogeneous and homogeneous environment of virtual machines.

Keywords: Cloud Computing, HDPSO, QoS, Scheduling, SLA, Workflow.

I. INTRODUCTION

The progress in technology, processing and storage and also the success of the Internet, make computing resources cheaper, more powerful and more available than ever before. As a result, a new computing model called cloud computing has arised. In this cloud computing, resources (e.g. CPU and storage) are provided as utilities that are leased and released by users in an on-demand fashion through internet.

The cloud providers usually host their services on internet and make it available for the customers who like to purchase them. The customers and cloud providers enter into an agreement through negotiation called Service Level Agreement (SLA). SLA clarifies the roles, set charges and expectations and also provides mechanisms for resolving service named problems within a specified time period. SLA covers performance, reliability conditions with respect to Quality of Service (QoS).

There are two entities in cloud computing: the infrastructure providers and service providers. The infrastructure providers manage cloud platforms and lease resources based on their usage. On the other hand, service providers rent resources from infrastructure providers to serve the end users.

Mainly, there are three service models in cloud computing: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). In SaaS, services are provided using a cloud infrastructure delivering applications to the user's browser. It also helps the organizations with limited resources to deploy and maintain the software, at the same time reducing the energy consumption and expenses. PaaS provides computational resources where applications and services are developed and hosted via a platform.

In cloud computing, scheduling is the process of mapping tasks onto resources and the systems (e.g. CPU time, bandwidth and memory) efficiently. Scheduling plays an important role in cloud computing to allocate the resources to corresponding job effectively. Scheduling can be of two types based on the dependency of job: Independent Scheduling (Task Scheduling) and Dependent Scheduling (Workflow Scheduling).

In cloud, the virtualization concept is used for providing the tasks a minimum completion time, better performance, resource utilization and quick response time. For task allocation, cloud uses virtual machine which is scalable but scheduling them is a major problem. Task scheduling is an important issue which greatly influences the performance of cloud computing environment.

The workflows are commonly modelled as a set of tasks interconnected via data or computing dependencies. Workflow scheduling mainly automate the procedures involved in the process of passing the data and files between the participants of the cloud, while maintaining the constraints. It is not possible to generate an optimal solution within a polynomial time, considering the large solution space for workflow scheduling. Hence, workflow scheduling is a NP hard problem and most of the algorithms focus on generating approximate or near optimal solutions.

Workflow Scheduling generally has two steps Resource provisioning and Scheduling. In resource provisioning, selection of resources to run tasks is done and in

Scheduling, map the tasks onto best suited resource. As a result, a schedule is generated meeting the different user defined Quality of Service (QoS). Since in grids and clusters, there is a static pool of resources and predefined configuration, they focus mainly on the scheduling phase. Also cost is least considered in these grids and clusters, whereas in cloud which is based on pay per use, cost is an important factor.

This paper is based on a variant of meta-heuristic optimization technique, Particle Swarm Optimization (PSO) named as Hybrid Discrete Particle Swarm Optimization (HDPSO). Inspired on the social behavior of bird flocks, Kennedy and Ebehart introduced the technique PSO. It is based on a swarm of particles moving through space and communicating to find the optimal search direction. PSO has better performance than other algorithms and is easier to implement since it has few parameters to tune with. In this paper, a cost-minimized scheduling technique is used in cloud environment considering the heterogeneity of virtual machines.

II. RELATED WORK

Generally there are two categories of the scheduling algorithm: Static Scheduling and Dynamic Scheduling. In static scheduling, tasks arrive simultaneously and the available resource schedule is updated after each task is scheduled. However in dynamic scheduling, there is no fixed task and machine set allocation. Dynamic scheduling is of two types: On-line mode heuristic scheduling and Batch mode heuristic scheduling. In online mode heuristic, tasks are scheduled when they arrive in the system. In Batch mode, when tasks arrive in the system, they are queued and collected into a set. The scheduling will start after a definite period of time.

The workflow scheduling algorithms are of two types:
a) Heuristic: Heuristic algorithms are based on priority where the user can use his knowledge to allocate priority for cloud resources and workflow applications.
b) Meta Heuristic: These algorithms do not need human interface. They provide a solution to workflow applications which are near optimal. The examples of meta heuristic algorithms are Ant Colony Optimization (ACO), Genetic Algorithms (GA) and Particle Swarm Optimization (PSO).

The various task scheduling techniques are:
A. QoS Guided Min-Min Heuristic
H E. Xiaoshan et al. [1] in 2003 suggested the QoS Guided Min-Min heuristic [Batch mode heuristic algorithm]. Here some tasks require higher network bandwidth, whereas some others need only a lower network bandwidth. In this algorithm, conventional Min-Min is used for matching the QoS request and services between the tasks and hosts.

In Min-Min algorithm, the earliest completion time and the host that obtains it is found. Then assign the task with minimum completion time to the host that gives the earliest completion time. Only one QoS issue has been addressed which is the bandwidth constraint. The QoS guided Min-Min algorithm executes tasks with higher bandwidth before those with lower bandwidth. If two tasks are having the same bandwidth, it acts similar to Min-Min.

Consider that there are four tasks T1, T2, T3 and T4 and two resources, R1 and R2. Let R1 and R2 have bandwidth 200 MBBS and 100 MBBS respectively. Let the completion time of each task on resource is as in the Table I. The Min-Min algorithm executes the smaller tasks first. Hence the order of execution will be T2, T1, T3 and T4. In QoS guided Min-Min higher bandwidth will be taken, which is given by Resource R1. So the task will be completed by R1.

TABLE I: COMPLETION TIME OF TASK ON RESOURCES

Task/Resource	R1	R2
T1	3	6
T2	2	4
T3	4	8
T4	5	9

B. QoS based predictive Max-Min, Min- Min Switcher
M. Singh et al [2] proposed the QoS based predictive Max-Min, Min-Min switcher algorithm in 2008. In this algorithm, an appropriate selection among QoS based Min-Min or QoS based Max-Min algorithm is used for scheduling the next job. The performance is predicted using the history information about the execution of jobs. This algorithm combines the efficiency of Max-Min along with Min-Min especially in makespan, the length of time interval between the initial processing of first task to the entire completion of the last task. The performance of the algorithm is better when it has the lowest makespan. The makespan is a measure of throughput. Here the non-dedicated property of resources is also considered.

Both Max-Min and Min-Min have two phases of execution. The first phase is same for both the algorithms where, the minimum expected completion time for each task is calculated. In Min-Min, the task with minimum expected completion time is chosen where for Max-Min, the task with maximum completion time is selected and assigned to the machine.

In this algorithm, the Min-Min will execute small tasks and Max-Min will execute the large tasks in parallel. The length of tasks is used to make a scheduling decision among QoS based Max-Min and QoS based Min-Min algorithms. Tasks with high QoS requests are mapped before mapping tasks with lower QoS requests.

C. RASA
In 2009, S. Parsa et al [3] introduced a task scheduling algorithm called RASA (Resource Aware Scheduling Algorithm) which combines the advantage of both Min-

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

Min and Max-Min algorithm. First, it estimates the completion time of the task on each resource and then applies both the algorithms alternatively. RASA uses the Min-Min algorithm to execute the small tasks first and then apply Max-Min to avoid the delay in the execution of large tasks. It also supports concurrency in the execution of the large and small tasks. It achieves the lower makespan with good QoS, considering the scalability and distribution of resources.

D. Improved Cost based Scheduling Algorithm

In 2010, Mrs. S. Selvarani et al [4] introduced an improved cost- based scheduling algorithm for mapping the tasks to available resources in cloud efficiently. This scheduling algorithm divides all user tasks depending on priority of each task. This scheduling algorithm measures both resource cost and computation performance. It also improves the computation or communication ratio by grouping the user tasks according to a cloud's processing capability. It then sends the grouped jobs to a resource.

E. TS-QoS

In 2013, X. Wu et al [5] introduced a task scheduling algorithm based on QoS driven in cloud computing (TS QoS) [Dynamic batching]. In this TS-QoS algorithm, the priority of tasks is computed according to the special attributes of the tasks such as user privilege, task length and pending time of task in queue. Then the tasks are sorted based on priority. The algorithm then calculate the completion time of each task on different services, and schedules each task onto a service which can complete the task as soon as possible according to the sorted task queue. But in this process, priority can change dynamically and increase continuously which help to solve the starvation problem following the First Come First Serve (FCFS) principle. It achieves good performance and load balancing by using both priority and completion time.

The various workflow scheduling techniques are

F. Budget Constrained Genetic Algorithm

In 2006, J. Yu et al. [6] suggested Genetic Algorithms, (GAs) applying the principle of evolution. It generates a high quality solution which is derived from a large search space in polynomial time. Any solution in the search space of the problem is represented by an individual (chromosomes). It maintains a population of individuals that evolves over generations. The quality of an individual in the population is determined by a fitness function. The fitness value indicates how good the individual is compared to others in the population. In GA, an initial population is created consisting of random solutions. New offsprings are then generated by applying genetic operators like selection, crossover and mutation. Fitness of each individual in the population is evaluated and repeated.

G. Ant Colony Optimization

In 2009, W. N. Chen et al. proposed the Ant Colony Optimization (ACO) [7], based on how ants find a path between their colony and the source of food. The ants are generated and mapping is done with the path and the objective function to be evaluated. The user can specify the QoS parameters while submitting the workflow application, preferring and optimizing them. ACO finds a schedule that meets all user imposed QoS constraints like deadline, budget and reliability. In ACO, the ants keep record of each and every node that they visit and record that data for future decision making. As a result they deposit pheromones during their movement for other ants to select the next nodes. Each ant works independently and represents a virtual machine looking for a host to get allocated.

H. PSO-Based Heuristic

In 2010, S. Pandey et al. proposed a Particle Swarm Optimization (PSO) heuristic for scheduling workflow applications in cloud [8]. This algorithm was developed by Dr. Eberhart and Dr. Kennedy in 1995. It considers both computation cost and data transmission cost and workloads are distributed with minimal cost. It mainly considers resource utilization and time as the main parameters.

The particle in PSO is generally the workflow and its tasks. The dimension is the number of tasks in the workflow and the moving range of particle is the number of resources in the resource pool. The fitness function will be the total execution cost of the schedule.

In this PSO based algorithm, a particle is represented by its position and velocity. Each particle has a best position, pbest and a global best solution, gbest. The particles fitness value will be compared with pbest. If the current value is better than pbest, update pbest to that current value and location. Similarly compare the particles fitness value with gbest and if current value is better, then update gbest to that current value and location.

This PSO algorithm is simple and effective for applications with low computational cost. It is being widely used in data mining, pattern recognition, environmental engineering etc.

I. Revised Discrete Particle Swarm Optimization

In 2010, Z. Wu et al. proposed the Revised Discrete Particle Swarm Optimization (RDPSO) [9], which schedule applications among cloud services considering both computation cost and data transmission cost. It achieves better performance on makespan and cost optimization.

The PSO algorithms usually give a better performance as it considers the dependencies between cost and tasks. In RDPSO, a set based concept is introduced into PSO, where each task is mapped onto a set of services. Also due to the discrete property of scheduling, the gbest will only have a few values to select from.

J. Deadline based Resource Provisioning

In 2014, M.A. Rodriguez et al developed a scheduling algorithm based on the meta-heuristic optimization

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016

technique, Particle Swarm Optimization (PSO) [10]. It aims to minimize the overall workflow execution cost while meeting deadline constraints on an IaaS cloud environment. Here the IaaS cloud features like pay-as-you-go model, heterogeneity, elasticity and dynamicity of resources are considered. Usually it performs better than the current algorithms considering cost and deadline as the main parameters.

III.PROBLEM FORMULATION

A workflow is represented by a DAG (Directed Acyclic Graph), G = (T, E) where T represents the tasks T= {t_1, t_2, . . . , t_n } and E represents the data dependencies among the tasks. In workflow scheduling, a large task is divided into different subtasks, where each are allocated to resources to achieve a predefined objective.

A sample workflow is as in Fig.1. Each node represents the tasks and the directed edges represent the data transfer time between the tasks.

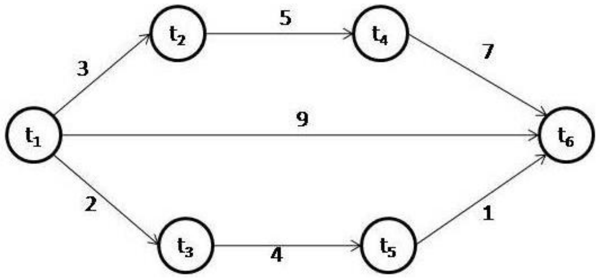


Fig. 1. A Sample Workflow

A. Problem Definition

The total execution cost (TEC) may include both execution cost (EC) and transfer cost (TC).

EC = costpersec * actual CPU time

TC = costperBW * filesize

TEC = EC + TC

Here, as the main aim is to minimize the execution cost, this is represented as Equation

Minimize TEC (Total Execution Cost)

B. Proposed Approach

Particle Swarm Optimization is an evolutionary technique based on behavior of animal flocks (e.g. fish or bird). A particle represents an individual moving through search space and is represented by velocity at a particular point. The velocity is determined by the best position the particle is in so far (pbest) and the best position in which any particle is in (gbest). The fitness function describes the quality of particle's position.

Each particle is represented by its position and velocity. Particles keep track of its best position (pbest) and global best solution (gbest) and change values towards the pbest and gbest values. The algorithm iterate until the stopping criterion, which is commonly either maximum number of iterations or predefined fitness value. The pseudo code for the PSO algorithm is shown as in Algorithm 1. In each iteration the particle update its position and velocity according to the equations respectively.

$$\vec{x}_i(t+1) = \vec{x}_i(t) + \vec{v}_i(t)$$

$$\vec{v}_i(t+1) = \omega.\vec{v}_i(t) + c_1 r_1 (\vec{x}_i^{*}(t) - \vec{x}_i(t)) + c_2 r_2 (\vec{x}^{*}(t) - \vec{x}_i(t))$$

where:
ω = inertia,
$c_1 = c_2$ = acceleration coefficients
$r_1 = r_2$ = random number ϵ [0,1]
$\vec{x}_i$ = current position of particle i
$\vec{x}_i^{*}$ = best position of particle i
$\vec{x}^{*}$ = position of best particle in population

Parameter c_1 is called cognitive parameter as it defines the previous best position and c_2 is called social parameter as it is relative to other neighbors.

ALGORITHM 1:- PARTICLE SWARM OPTIMIZATION
1. Set the dimension of particle as d
2: Initialize the particle's population with random position and velocities.
3: for each particle, calculate its fitness value do
4: Compare the particle's fitness value with the particle's pbest. If the current value is better than pbest, then set pbest to the current value and location.
5: Compare the particle's fitness value with the global best gbest. If the particle's current value is better than gbest, then set gbest to the current value and location.
6: Update the position and velocity according to the equations.
7: end for
8: Repeat from step 3 until the stopping criterion is met.

On defining the meaning and dimension of particle, the particle represents workflow and its tasks and hence the dimension of particle defines the number of tasks in the workflow.

In the proposed approach, HDPSO [11] is used instead of PSO for resource provisioning. HDPSO is the hybrid combination of Min-Min and DPSO. The pseudo code for HDPSO is as in Algorithm 2.

ALGORITHM 2:- HDPSO
1: Generate initial population using Min-Min.
2: Apply fitness function and evaluate each particle in initial population.
3: Find out best position of each particle and global best position of particles, pbest and gbest respectively for the initial population.
4: Update the position and velocity according to the equations.
5: Repeat until the stopping criterion is met i.e, maximum number of iterations.

Both PSO and HDPSO are then evaluated under homogeneous and heterogeneous environments of virtual machines. In homogeneous, there is only one virtual machine and in heterogeneous, there are more than one virtual machine.

IV. EXPERIMENTAL EVALUATION

A. Experimental Environment and Parameter Settings

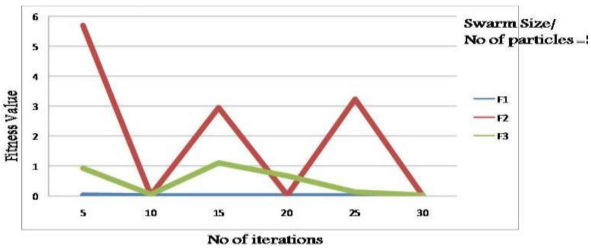
The evaluation is done in a simulated environment using Workflow Simulator [12] with Java NetBeans IDE 8.1. The different parameters set for PSO and HDPSO are as in Table II.

Table II: PARAMETER SETTINGS FOR PSO AND HDPSO

Parameters	Values
Population Size	10
Maximum Iterations	10
Inertia Weight (ω)	0.5
Acceleration Coefficient (C_1)	1.0
Acceleration Coefficient (C_2)	1.0
Maximum Velocity	4.0

B. Result Analysis

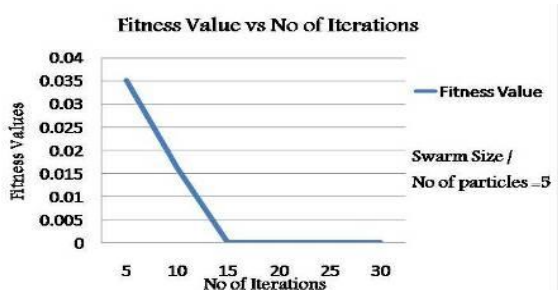
For performance analysis, the workflows mainly Montage, CyberShake and Sipht are considered. The Graph1 shows the effect of different fitness function on PSO. Out of the three fitness functions, F1 gives the minimum fitness value and hence it is taken as fitness function for PSO.



F1 = $(2.8125\text{-}x+xy^4)^2 + (2.25\text{-}x+xy^2)^2 + (1.5 - x+ xy)^2$
F2 = $(x\text{-}y)^2 + ((x+y\text{-}10)/3)^2$
F3 = $100(y\text{-}x^2)^2 + (1\text{-}x)^2$

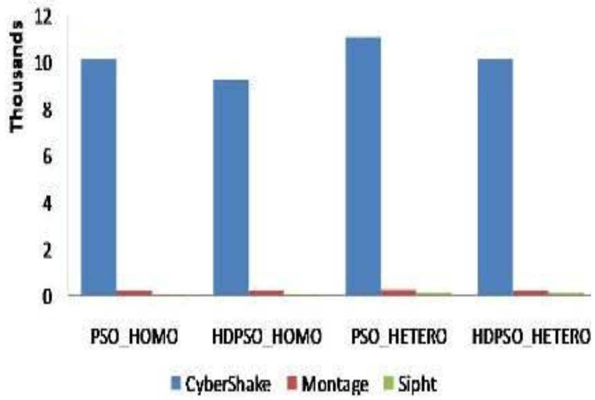
Graph 1: Effect of different fitness functions

Using the fitness function F1, the effect of fitness value with increase in iterations is as in Graph 2.

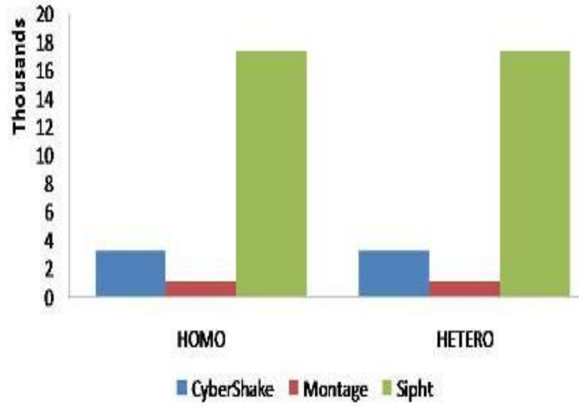


Graph 2: Effect of fitness Values with increase in no of iterations

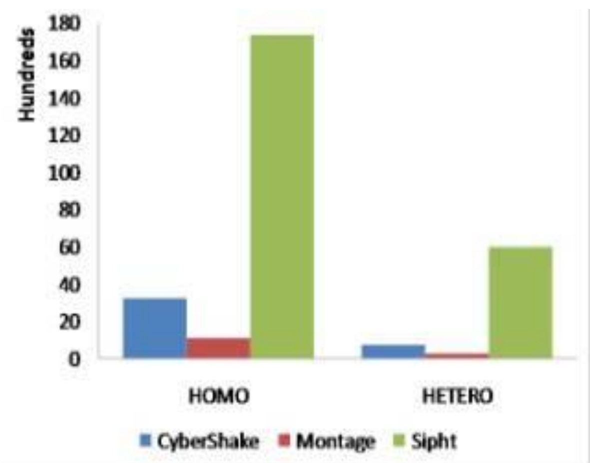
The algorithms PSO and HDPSO are considered in both homogeneous and heterogeneous environment of virtual machines. The effect of execution cost, time and makespan on the workflows namely CyberShake, Montage and Sipht is as in Graph3, Graph4 and Graph5 respectively.



Graph3: Effect of Execution Cost on workflows



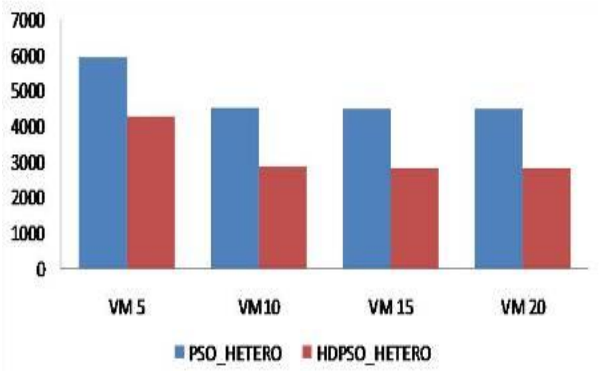
Graph4: Effect of Execution Time on workflows



Graph5: Effect of Makespan on Workflows

On comparing the value of makespan, by varying the number of virtual machines it can be seen that the value decreases. The makespan is compared on Sipht workflow by increasing the number of virtual machines from 5 to 20 and is plotted as in Graph6.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 7, July 2016



Graph 6 : Comparing makespan by varying VM on Sipht

V. CONCLUSION

Cloud computing which is based on pay per use has execution cost as an important factor. In this paper, it considers various principles like pay-as you go model, elasticity and heterogeneity of the resources. Out of all workflow scheduling algorithms, PSO is used because it has faster convergence, fewer parameters to tune and easier to implement. HDPSO is a hybrid of DPSO and Min-Min to overcome the local search capability of PSO. The experimental results show that the use of HDPSO instead of PSO in resource provisioning helps to minimize the execution cost. Also HDPSO has lesser makespan than PSO in heterogeneous environment of virtual machines which makes HDPSO to have better performance than PSO. In future, any other method for further reducing the execution cost can be done.

REFERENCES

[1] Xiaoshan HE, Xian-He Sun, Gergor von Laszewski," QoS guided Min- Min heuristic for grid task scheduling, " Journal of Computer Science and Technology , pp.442-451, May 2003.

[2] M.Singh and P.K.Suri," Max-Min, Min-Min : A QoS Based Predictive Max- Min, Min-Min Switcher Algorithm for Job Scheduling in a Grid ," Inform. Technology Journal 7(8) , pp.1176-1181, 2008.

[3] Saeed Parsa and Reza Entezari-Maleki," RASA: A New Task Scheduling Algorithm in Grid Environment , " World Applied Sciences Journal 7 (Special Issue of Computer & IT) , pp. 152-160, 2009.

[4] Mrs. S.Selvarani, Dr.G.Sudha Sadhasivam," Improved cost-based algorithm for task scheduling in Cloud computing ," IEEE, pp.1 -5, 2010.

[5] Xiaonian Wu, Mengqing Deng, Runlian Zhang, Bing Zeng, Shengyuan Zhou," A task scheduling algorithm based on QoS-driven in Cloud Computing," Information Technology and Quantitative Management (ITQM2013), pp.1162-1169,2013.

[6] J. Yu and R Buyya," A budget constrained scheduling of workflow applications on utility grids using genetic algorithms," Proc. 1st Workshop Workflows Support Large-Scale Sci., pp. 01-10 , 2006.

[7] Wei Neng Chena and Jun Zhang," An Ant Colony Optimization Approach to a Grid Workflow Scheduling Problem with Various QoS Requirements," IEEE Transactions on System, Man and Cybernetics, Applications and Reviews, Vol. 39, No.1, pp. 29-43,2009.

[8] S. Pandey, L. Wu, S.M. 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.

[9] Z. Wu, Z. Ni, L. Gu," A Revised Discrete Particle Swarm Optimization for Cloud Workflow Scheduling," Computational Intelligence and Security (CIS), pp. 184-188, 2010.

[10] Maria Alejandra Rodriguez and Rajkumar Buyya," Deadline based Resource Provisioning and Scheduling Algorithm for Scientific Workflows on Clouds," IEEE Transactions on Cloud Computing, Vol. 2, No. 2, pp.222-235, 2014.

[11] Purnima Devi, Mala kalra, "Workflow Scheduling Using Hybrid Discrete Particle Swarm Optimization (HDPSO) In Cloud Computing Environment" , International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE), Vol. 3, Issue 12, pp.12301-12307, December 2015.

[12] W.Chen, E.Deelman, "WorkflowSim: A Toolkit for Simulating Scientific Workflows in Distributed Environments", IEEE 8th International Conference on E-Science (e-Science), pp. 1 – 8, Oct. 2012.