Enhanced Scheduling Techniques to Improve the Performance of Task Assignment in Cloud Computing

W.Regis Anne, Assistant Professor, Sri Krishna College of Engineering and Technology, Coimbatore

Abstract **—Cloud computing which provides computing as a service has evolved through a number of phases which include grid and utility computing, application service provision (ASP), and Software as a Service (SaaS). Cloud computing has developed along a number of lines with web 2.0 being the recent evolution. Many milestones were achieved later like the arrival of salesforce.com, Amazon Web Services, Google Apps . Later the Killer Apps from the giant players like Google and Microsoft was a major contribution to cloud computing. Other key factors that have enabled cloud computing to evolve include the maturing of virtualization technology, the development of universal high-speed bandwidth and universal software interoperability standards In computing, virtualization is abstraction of computer resources. Virtualization offers virtual resource to users demand by optimizing resource and power consumption. The main objective of this paper is to propose a heuristic algorithm for virtual machine scheduling which is an NP hard problem.**

Keywords—Cloud;Virtualization;Scheduling; Performance

I. INTRODUCTION

Cloud computing is delivering of computing resources over the internet based on the needs of the users. Cloud computing is an extension and improvement of grid computing, distributed computing, parallel computing and Utility computing [1]. There has been a explosive increase in the use of Cloud computing and it demands better results and services. In Cloud Computing [2] scalable resources are provisioned dynamically as a service over internet in order to assure lots of monetary benefits to be scattered among its adopters.Load balancing in Cloud has become a very interesting and important research area. The load balancing concepts can be broadly classified into three sectors such as Data Centre Selection, Virtual Machine Scheduling and Task scheduling at particular data centre.

Task scheduling problem [3] is the problem of matching tasks to different sets of resources which is formally expressed as a triple (T, S, O) where 'T' is the set of tasks, each of which is an instance of problem, the set of feasible solutions is 'S' and the objective of the problem is 'O'. Task scheduling problem is NP hard that can be solved on a nondeterministic Turing machine in polynomial time.

The main objective of this paper is to propose a heuristic algorithm for task scheduling problem which generates an approximate solution in polynomial time with the objective 'O' being achieved. Then an optimization on the heuristic finds the best solution among all the feasible solution in the set S.

The proposed heuristic task scheduling algorithms utilizes the resources properly while managing the load between the resources the tasks get done in minimum execution time. The proposed scheduling algorithm defines the priority order for the different tasks and also ensures that all the tasks will meet its deadlines using this priority order.Scheduling in cloud computing can be categorized into three stages.

1) Discovering of all resources and filtering them.

2) Selecting a target resource.

3) Scheduling algorithm executes the tasks on the target resource.

The different tasks from the users are given to the task scheduler. The requirements of the tasks are mapped onto the available resources using the scheduling algorithm. The tasks are then executed on the virtual machines according to the proposed scheduling process. The flow is described in Fig .1.

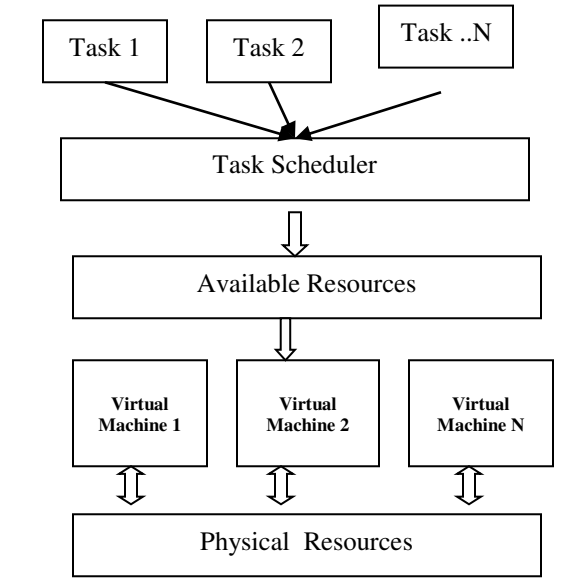

Fig 1: Task Scheduling Process

II. REVIEW OF LITERATURE

The task scheduling algorithms can be static or dynamic. In static algorithms, the basic information should be learned in advance like sequence of parallel tasks, execution time for each task and the communication cost for each task. In dynamic scheduling the allocation of the tasks to the resources and the schedule plan is done dynamically. The examples of scheduling algorithms are First Come First Served, Round-Robin, Min-Min algorithm, Max-Min algorithm and meta- heuristic algorithms like Ant Colony Optimization, Genetic Algorithm, Simulated annealing, Particle Swarm Optimization(PSO), Tabu search and many more. The Pre-emptable shortest job next scheduling algorithm [4] combine the pre-emption technique of Roud-robin algorithm with shortest process next. This algorithm gives cost benefits and improves the response time and execution time. In Shortest Job scheduling [5] includes the allocation of resources on different clouds under over-load and under-load conditions. The Optimized Activity based Costing algorithm [6] compares the optimized algorithm traditional task scheduling algorithms and has proved to be more efficient. The idea behind Ant colony Optimization [7] is to simulate the foraging behavior of ant colonies and have set up the phremone rule for scheduling. In Deadline and Budget Distribution based Cost-Time Optimization Algorithm [8] considers both deadline of executing the tasks and budget. The Improved cost-based algorithm for task scheduling in Cloud Computing Environment [9] schedules groups of task where resources are having different resource costs and different computation performance. In Genetic Simulated Annealing Algorithm for Tasks Scheduling [10] considers both Genetic Algorithm and Simulated Annealing and develops an algorithm that considers QOS requirements of different type of tasks, corresponding to the nature of tasks of the users in cloud. In this paper we first define an Modified Earliest Deadline First (MEDF) algorithm and then consider a PSO solution for an optimal schedule.

The scheduling model consists of a set of periodic tasks $T = \{t_1, t_2,$. Each is released periodically with a period and has a deadline and arrival time .The size of each task is constant. The tasks are executed on the set of viritual machines $V = \{v_1, v_2,$ where each virtual machine has a constant computation power.The time for execution of is . The finishing time of is

$$Finish(t_k) = start(t_k) + \Gamma_k$$

Objective: $Min(Finish(t_k))$

The objective is to minimize
Subjected to (i) the job must finish before its deadline and (ii) each task can be allocated to one virtual machine .

III. PROPOSED MODIFIED EARLIEST DEADLINE FIRST (MEDF) ALGORITHM

In EDF algorithm, the task which has an upcoming deadline are scheduled to the virtual machine so that these tasks will not miss their deadline. In this process, a few tasks will miss their deadlines and the utilization of resources of the virtual machines are reduced. So the Modified Earliest Deadline First algorithm the scheduler checks the constraint (Eq .(2))of the task with the available virtual machine. Among the virtual machine set that is available it chooses the virtual machine with minimum finishing time. The scheduler also checks that the resources are utilized to the maximum. If any virtual machine is not available, it waits for a time period and searches again among the virtual machines and assigns the task to a free virtual machine. The algorithm is given below,

Input: T –set of tasks, V-Set of Virtual Machine
Output: Vector of successful execution time of all the tasks T

1: $MAX = max\ (d$
2: for i = 1 to MAX do
3: RQ is empty set
4: for j=1 to No.of Tasks do
5: if ($(a_j \leq i \ \&\ t_j$ *not assig*
6:
7: end if
8: end for
9: Sort as Ascending order of Dead line of Tasks
10: for m= 1 to length(rq) do
11. FT is an empty set
12: for k = 1 to No.of Hosts do
13: if v_k *free in V &* $Finis\square(t_k$
14. FT = FT U
15: end if
16: end for
17: Assign Task
rq_m *to* $v_k \in Finish(t_k) = min$
18. end for
19.for k=1 to No.of Hosts do
20: if Task on complete execution then
21: Vector V =
22: Free
23: end if
24: end for
25: end for

The performance of MEDF performs better than the basic EDF and the resource utilization increases as the number of virtual machine or the number of task increases.

IV. OPTIMIZATION OF THE PROPOSED ALGORITHM USING MODIFIED PSO

Particle Swarm Optimization (PSO) is an optimization technique to find optimal solution based on the bird's movement and intelligence [11]. PSO has number of particles i.e.,tasks termed as fitness task that moves in the problem search space following rules impressed by the behavior of flocks of birds searching for food to find optimal solution i.e optimal schedule. The particles represent the solution to the problem. The particles position and velocity has to be updated every iteration to find the new location. PSO algorithm for task scheduling problem represents the location of particle with Minimization of execution time is $X_k = \{ x_1^k, x_i^j$ where is particle k with execution time .
The PSO algorithm for the scheduling problem has two core functions which updates the particles location and fitness function at time and is given by
Eq (3) and Eq(4).

$$V_{id}(t+1) = w\, V_{id}(t) + c_1 \times r_1 \times (P_{best}(t) - x_{id}(t)) + c_2 \times r_2 \times (P_{gbest}(t) - x_{id}(t)) \quad (3)$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t+1) \quad (4)$$

The velocity is managed by the inertia weight which influences the local and global abilities of

the algorithm. The values of c1 and c2 are acceleration control that decides the particle's next movement.The fitness function for each particle is given below,

$$f(x) = \frac{\gamma}{Min(Finish(t_k))}$$

Where is a constant. The proposed algorithm is given below,

Proposed Modified PSO Algorithm:

Input: T –set of tasks, V-Set of Virtual Machine
Output: Optimal schedule of all the tasks T

1. T is the no. of tasks.
2. V is the set of Virtual machines
3.Initialize positions or velocities of all particles
4. Set initial personal_best
5.For each particle do
6. represents the execution time of the task on virtual machine
7.Generate initial solutions for all tasks on the available resources
8.if fitness _value > personal_ bests
9. current_value= personal_best_position
10. End if
11.select the particle with best_fitness as global_best
12.Generate set of tasks and calculate the fitness for each task based on the objective function
13.For each task do
14. calculate the personal_best and global_best for each iteration
15.If maximum iteration reached or end criteria reached end for
16.Else continue
17.End for
18.Find the minimum value out of all iterations.
19.Update the task allocation to the minimum execution time.

V. SIMULATIONS AND RESULTS

In this section, the simulation setup and the results are presented.CloudSim is used to evaluate the scheduling of MEDF and Modified PSO. The experiments consist of 40 Virtual machines and 100-250 tasks under the simulation platform. The parameters setting on the proposed algorithm is shown in Table 1.

Table 1 Parameters and Value for Simulations

Parameter	Value
Number of Virtual Machines	40
Speed of PE 1000	200000MIPS
Power Consumption 0.28	3.45kW
Task Total Number of Tasks	100-250
Length of Tasks	5000-15000 MIPS
Number of Iterations	5000
External File size	600
Mutation operator	.5
Particles	30
C1,C2	1.5
W	0.5

Figure 1. represents the execution time of the basic EDF,MEDF and Proposed PSO for 40 virtual machines. The execution time of the tasks for different cloudlets.

The execution time for EDF is the highest than all the other algorithm because it does not check the constraints when assigning the jobs to virtual machine. The MEBF algorithm performs comparatively better than EDF because it checks the constraint before executing the tasks on the virtual machine. The execution time of modified PSO is the lowest because it tries to optimize the solution from different solutions that are available.

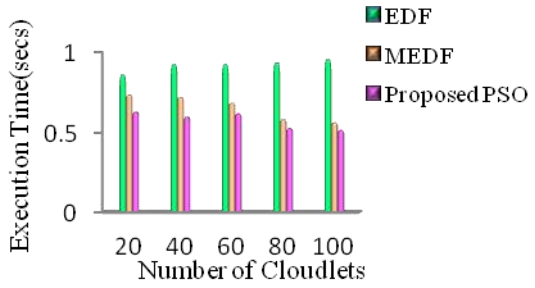

Figure 1 Execution time for different algorithms

Figure 2. represents the utilization time of the basic EDF, MEDF and Proposed PSO for 40 virtual machines. The proposed algorithms perform better and better utilizes the available virtual machines when compared to the basic EDF.

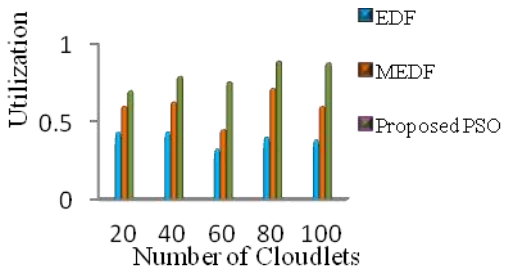

Figure 2 Utilisation vs Number of Cloudlets

VI.CONCLUSIONS

With the advancement of Cloud technologies rapidly, there is a new need for tools to study and analyze the benefits of the technology and how best to apply the technology to large-scaled applications. The proposed methods considers the scheduling problem where not only the processing time of a job is given importance but the other issues are considered such as the execution time. The proposed methods perform better than the basic algorithms.

References

[1]L.Zhu,Q.Li,L.He,"Study on Cloud Computing Resource Scheduling Strategy Based on the Ant Colony Optimization Algorithm"International Journal of Computer Science Issues,2012,pp.54-58.

[2] Bechtolsheim A, "Cloud Computing and Cloud Networking" talk at UC Berkeley,
December 2008.

[3] J. Blazewicz. "Scheduling in Computer and Manufacturing Systems" Springer-Verlag
New York, Inc., Secaucus, NJ, USA, 1996.

[4] PoonamDevi , TrilokGaba, "Implementation of Cloud Computing By Using Short Job Scheduling" International Journal of Advanced Research in Computer Science and Software Engineering, Vol. no.3, Issue 7, pp 178-183, july 2013.

[5] P. Mathiyalagan , S . Suriya , S.NSivnaam , "Modified Ant Colony Algorithm for Grid Scheduling", International Journal on computer science and Engineering Vol. 02, no 02,pp 132-139(2010).

[6] Nishant.S.Sanghani, R.J. Khimani, K.K. Sutaria, Pooja. P. Vasani, "Pre-emptable shortest job next scheduling in private cloud computing" Journal of Information ,Knowledge and research in computer engineering, vol no. 02, Issue- 02, pp 385-388 , Nov 12-Oct 13.

[7] S. Aranganatham , K.M. Mehta, "An ACO Algorithm for Scheduling data intensive application with various QOS requirements" International journal of computer Applications(0973-8887) Vol 27, no 10 , pp 1-5, August(2011)

[8] A. Verma, S. Kaushal, "Deadline constraint heuristic based genetic algorithm for workflow scheduling in cloud," Forthcoming article in international journal of grid and utility computing.

[9] S. Selvarani, G.S. Sadhasivam, "Improved cost-based algorithm for task scheduling in cloud computing," computational intelligence
and computing research, pp.1-5, 2010.

[10] Haluk, T., Salim, H., and Wu, M.Y, "Performance-effective and low complexity task scheduling for heterogeneous computing", IEEE Transaction on Parallel and Distributed Systems, vol. 13, no.3
pp: 260-274, Mar 2002.

[11] Kennedy, J.; Eberhart, R., 'Tarticle Swarm Optimization", *IEEE International*
Conference on Neural Networks, Perth, WA, Australia, Nov. 1995, pp. 1942-1948.