ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 8, August 2016

Particle Swarm Optimization Based Task Scheduling for Cloud Computing

Gifty Garg, Seema Sabharwal, Anurag Jain

Abstract**—**Task scheduling is the important requirement in cloud computing, whole cloud computing facilities efficiency is dependent on task scheduling. In cloud computing, task scheduling by considering different parameters like make span, cost, scalability, time, reliability, availability, resource utilization, etc. it does the allocation of best suitable resources to the task that is to be executed. The main motive of research is to propose a new algorithm using particle swarm optimization technique which has better and less response time and do better task scheduling on VM's. Using GA the problem of complexity and convergence increases the response time and does not balances the load among the VM's. So, we introduce a new algorithm using particle swarm optimization that resolves these problems. The objective is to test the proposed approach through simulator and do the comparative analysis with genetic algorithm based approach on the following parameter (a) Makespan (b) Efficient utilization of VM and Do the comparison on above mentioned parameters in various scenarios.

Index Terms— Cloud computing, Particle swarm optimization, Makespan, Response Time, Task Scheduling, Vm Utilization.

I. INTRODUCTION

A cloud is defined as a type of parallel and distributed system which is a collection of virtualized and interconnected computers that are provisioned dynamically. These computers are presented as single computer resource which is based on the service level agreement established or signed between the consumer and service provider.

National Institute of Standards and Technology (NIST) defined cloud computing as "a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minima management effort or service provider interaction." Cloud computing is a shared infrastructure as it can attach huge system pools and provides storage and complexity resources via internet to the users. It has the ability to provide anything as a service like platform as a service, software as a service and infrastructure as a service.

Iaas make available consumer's specified hardware immediately when requested. It provides consumer various services like storsge, processing, computing resources where consumer can deploy and run other software.

Paas It provides specific programming language and tools to developer to develop software.

Saas It provides access to the applications to providers that are running oooon cloud infrastructure. Providers can access these applications from various client devices with the use of a thin client interface.

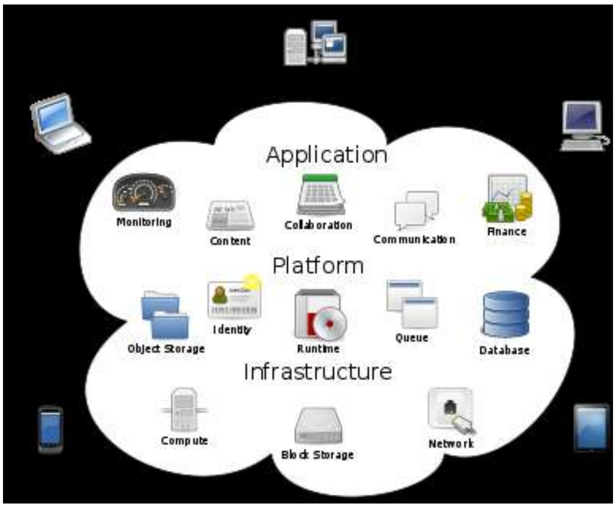

Fig.-1 cloud computing service models

As cloud computing has many advantages and services to provide but two of the major problem of cloud computing is task scheduling and resource management.

In cloud computing and grid computing two main problems areas of problems are task scheduling and provisions of resources. Scheduling of tasks must be done efficiently so that execution time and cost can be reduced. Various researchers give many algorithms for task scheduling which are discussed in related work.

The remaining sections of this paper are organized as follows: section II gives the overview of genetic algorithm. Section III describes the basics of particle swarm optimization. Section IV gives related work regarding task scheduling problems in cloud computing. Section V presents the proposed work. Section VI presents the experiment results and section VII presents the conclusions of implemented work.

Manuscript received August, 2016.

Gifty Garg*, computer science department, Geeta institute of management and technology, Kanipla, Kurukshetra, Haryana, India, +91-9996303910.*

Seema Sabharwal, *computer science department, Geeta institute of management and technology, Kanipla, Kurukshetra, Haryana, India +91-9996740777.*

Anurag Jain*, computer science department, Chandigarh group of colleges, Landran, Mohali, Punjab, India, +91-9466096567.*

II. GENETIC ALGORITHM

In the evolutionary computing genetic algorithm is a rapidly growing area of artificial intelligence. Genetic algorithms are based on the Darwin's theory of evolution [8]. For evolution simulation of natural system is done which follow the principle of survival of the fittest. When the search space is very large and can't be traversed by classical search techniques then genetic algorithms are used for e.g. when a problem solution requires equilibrium and evaluation of many different unrelated variables then a random search space is represented with in a defined search space [8].

In genetic algorithm a set of solutions is called a population. One single solution of the population is called the chromosome. In a number of generations fittest solutions are selected for each generation to form a new population. Mutation, reproduction and crossover are three main operators that are used during the cycle. The same cycle is repeated for a number of generations until a termination criterion is met. When a chromosome with a highest fitness value is evaluated the cycle is terminated [8].

Characteristics of genetic algorithm differentiated from other conventional techniques

1. Sample population representation must be derived.
2. Encoded versions of variables are directly manipulated by GA's rather than the variables themselves.
3. Stochastic operators are used rather than the deterministic.
4. Sampling is used by GA'S and they ignore all information except the outcome of the solution.
5. GA starts searching by considering points from population rather than from a single point.[8]

Basic Steps of GA:

Population Generation: Population is like the database. Population contains the information about the individuals. For a healthy population it is necessary to have several members. Unhealthy members are generated by small population. From one generation to other, population changes its individuals. Having a large population provides a large search space. If the generation structure is uniform at all gene positions without reaching an optimal structure this is called premature population [7].

Seeding: For optimization firstly population is generated which is then seeded with set of parameter values that can influence search through space. Previous experiments values can also be used to provide initial population. Alternate values can also be used. Seeding generates random data it is performed by random selection [7].

Chromosome Representation: in genetic algorithm different methods are used to represent chromosome e.g. binary, gray, integer or floating data types. Some types of encoding is discussed below:

Binary Encoding: A binary string is encoded by each chromosome. Each bit in the string represents some characteristic of solution. [7].

Octal Encoding: Octal encoding uses octal numbers to make the string [7]

Permutation Encoding: In permutation encoding a string of integers/real number is used to represent chromosome [7].

Value Encoding: In value encoding a sequence of some values represents a chromosome. Values can be anything selected to problem e.g. real numbers, chars, and an object [7].

Tree Encoding: In tree encoding a tree of some objects such as functions and commands in a program represents a chromosome. It is mainly used for solving program or expressions i.e. for genetic programming [7].

Selection Techniques: According to Darwin's theory of evolution the best ones survive to create new offspring. As chromosomes selection is done from the population for crossover. The main problem here is how to select these chromosomes. Several methods for selection of chromosomes are discussed in this section:

Roulette Wheel Selection: Fitness value is the key according to which parents are selected. The better chromosomes have more chances to be selected. Consider a roulette wheel where all the population chromosomes are placed on it. According to the fitness function value of every chromosome, the size of the section in the roulette wheel is decided i.e. bigger the value larger the section. For e.g.

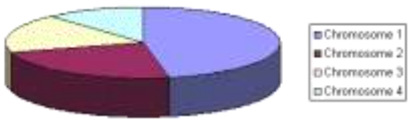

Fig.-2Roulette Wheel Selection [7]

Rank Selection: Roulette wheel selection method will have problems when there is big difference between the fitness values. For example, a best chromosome will have least chances to be selected if its fitness is 90% of the sum of all fitness values. In rank selection firstly the ranking of population is done then the fitness value of every chromosome is determined. The worst chromosome will have fitness value 1 and the best one will have N. Here fig1.3 shows situation before ranking and fig 1.4 shows situation after ranking.

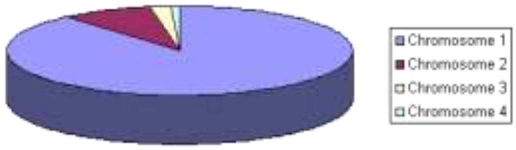

Fig.-3Situation before ranking (graph of fitnesses) [7]

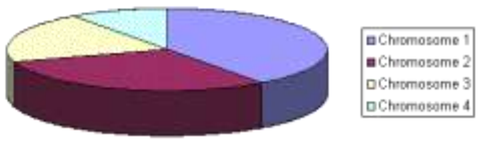

Fig.-4 Situation after ranking (graph of order numbers) [7]

Now all the chromosomes have equal chances to be selected.

Elitism: A new population is created where first best chromosome or few best chromosomes are copied. The rest is done in classical way. Individuals that are not selected are lost or get destroyed by mutation or crossover. This improves the GA's performance [8].

Crossover: Crossover is the process of producing child by taking two parent solutions. Population contains the enriched better individuals after the process of selection. Reproduction does not create new ones but makes the clones of good strings. Crossover operator is applied to the mating pool with the hope that it creates a better offspring. [8]

Single Point Crossover: Single point crossover is used in the traditional genetic algorithm, where one crossover point is selected. First chromosome's binary string is selected till the point of crossover and rest is the other parent string.

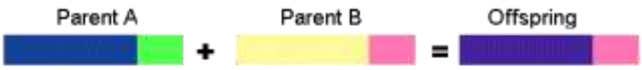

Fig.-5 Single Point Crossover

11001011+11011111 = 11001111

Two Point Crossover: Here two crossover points are selected, from the first parent the string from the beginning to the first crossover point is copied, the part from the second crossover point is copied from the other parent and rest is copied from the first parent again.

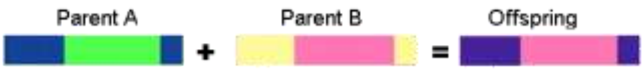

Fig.-6 Two Point Crossover

11001011 + 11011111 = 11011111

Uniform Crossover: Randomly bits are copied from the first or from the second parent.

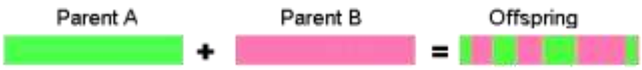

Fig.-7 Uniform Crossover

11001011 + 11011101 = 11011111

Arithmetic Crossover: To make a new offspring some arithmetic operations are performed.

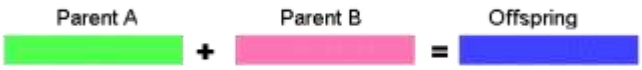

Fig.-8 Arithmetic Crossover [7]

11001011 + 11011111 = 11001001 (AND)

Mutation: Mutation takes place after crossover. To maintain the genetic diversity from one generation of a population of chromosomes to the next mutation operator is used. During evolution mutation occurs and usually sets 0.01 as a fairly low first choice value according to a user definable mutation probability. One or more gene values are alter by mutation from its initial state. This resulted new gene is added to the gene pool. Genetic algorithm may be able to result in better solutions with these new gene values. Mutation prevent the population from stagnating at any local optima.

III. PARTICLE SWARM OPTIMIZATION

Kennedy and Eberhart proposed a algorithm on the Particle swarm optimization which is based on the simulation of social behavior of birds within the flock. Particle swarm optimization is a population base searched algorithm.

Let us take a D-dimensional search space, where each particle in the search space is defined as a solution to a problem. The ith particle of the swarm is represented as: Xi = (xi1, xi2, xi3……xiD) and the velocity of particle is defined as Vi = (vi1, vi2, vi3………viD). Equations (1) & (2) update the particles at each generation. In the iteration t, best position for all the particles xg is accomplished when velocity vi(t) is being updated to pull the ith particle towards its best position xpi. Best fitness value from the preceding generation is the best position for the particles. The current velocity of each iteration t is based the vi(t-1) i.e. velocity of previous iteration.

Here r1, r2 are uniform random variables generated independently, c1, c2 are two positive constants and w is the inertia weight.

Position of each particle in the hyperspace is updated by equation (2) using the computed vi(t) and without the loss of generality. Allocation of Vms to tasks is done by PSO and guarantees appropriate execution of tasks with their load of Vm. In PSO we calculate the load of Vms.

Load of Vmi= (resource of Vmi / total resource)*N.

Equations (1) & (2) are as follows:

$$V_i^{K+1} = WV_i^K + c_1r_1*(pbest^i - X_i^k) + c_2r_2*(gbest - X_i^k) \quad (1)$$

$$X_i^{k+}1 = x_i^k + V_i^{K+1} \quad (2)$$

Nomenclature

V_i^K = velocity of particle i at k iteration
$V_i^{K+}1$= velocity of particle i at k+1iteration
W = weight of inertia
ci = coefficients of accerlation; j = 1, 2
ri = random number between 0 and 1; i = 1, 2
x_i^k = current position of particle i at iteration k
X_i^k+1= current position of particle i at k+1iteration
pbesti = particle I best position
gbest = best particle position in a population

IV. RELATED WORK

(Mc Evoy. G and Schulze.B, 2011) explored that during the scientific execution of applications on the cloud computing, it has some effects on schedulers. Author present premises of aspects of cloud infrastructure regarding the provisioning and architectural, that are not same from other cloud environments, and which implicates that they may take scheduling decisions by considering the relevant policies like improving performance. As authors research towards scientific application deployment, they test and proposed a preliminary workload classification, which is based on mode of usage that may improve early scheduling decisions.

(Mahalle, H. S et.al, 2013) during his research the author analyzed and discussed the performance of three algorithms. For the all three applied policies the request time (Round Robin, Equally spread current execution load, Throttled Load balancing) is kept same so that after the change of algorithm there should be no effect on data centers request time. In the experimental work an illustration for the analysis of cost is

ISSN: 2278 – 1323
International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)
Volume 5, Issue 8, August 2016

calculated. For two algorithms Round Robin, Equally spread current execution load the calculate cost for virtual machine usage per hour is same but is less for the Throttled Load balancing algorithm, so, on cloud datacenters more efficient results are given by Throttled Load balancing algorithm in terms of cost .

(Zhong.H et al, 2010) analyzed the cloud computing problems of optimization or sub-optimization and proposed an optimized scheduling algorithm for it. Authors investigated that to permit the maximum usage of physical resources it might be possible to assign VMs in a flexible way. Authors suggested for the selection of optimal or sub-optimal allocation for the VMs requests IGA can be used for the automated scheduling policy. IGA introduces the idea of dividend policy and make use of shortest genes. As according to the requests of text exemplified IGA method is twice faster than that of traditional GA scheduling method in grid environment and resource utilization rate is higher than the open source IAAS cloud system.

(Dutta.D and Joshi.R C, 2011) proposed a genetic algorithm approach to cost base multi QOS job scheduling. Authors proposed a model for cloud computing environment and popular genetic operators such as cross operators like PMX, OX,CX & mutation operators like swap & insertion mutation are used to produce a better schedule. In finite time a best solution is assumed by this algorithm.

Guo L,Zhao S,Shen S, 2012 Jiang C analyzed a task scheduling model and proposed a particle swarm optimization algorithm for minimizing the processing cost. This algorithm is based on the small position value rule. On comparing embedded PSO algorithm in crossover and mutation and in the local research with the PSO algorithm, new proposed PSO show faster convergence results. It is prove from the experiment result that for cloud computing PSO algorithm is more suitable.

Pandey S, Wu L, Guru S, Buyya R,2010 analyzed both the computation cost and data transmission cost and presented a heuristic based on the particle swarm optimization (PSO) to schedule applications to cloud resources. By varying its computation and communication costs it is used for workflow application. The results of experiments show that cost savings and better distribution of workload between resources can be achieved by PSO.

V. PROPOSED WORK

A. Model structure:

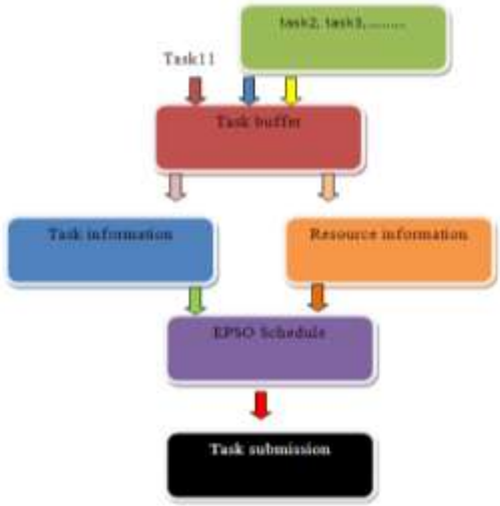

Fig.-9 Proposed model structure

The main motive of our model is the allocation of tasks to virtual machines with considering reliability. Our proposed model structure is shown in figure 1. There are five phases in our proposed model. Model phase's concepts are:

1. Task Buffer: In the cloud computing there are millions of users require executing tasks. Collection of tasks from the users is the responsibility of task buffer.
2. Task Information: necessary information of the tasks arrived into cloud computing environment for execution is provided by this phase. This information is about the Expected Execution Time (EET), Expected Transmission Time (ETT), Resources-Required (RR) and Round Trip Time (RTT).
3. Resource Information: This phase is responsible for the collection of information about resources in cloud computing environment. Resources of cloud computing are Hosts, Datacenter, and virtual machines (VMs). Datacenter contains information of the storage list, VMs list, host list and cost of memory, cost of BW and others. Each host can contain more than one VM. The hosts and VMs contain information such as ram, mips, bandwidth and other information. To the next phase these information machines are passed.
4. EPSO: Efficient PSO is used to reschedule the tasks that are failure to schedule. There are two problems of PSO. First problem, allocation of tasks to virtual machine may fail. Second problem, allocation of task to more than one VM. This phase solve the problems by reschedule wrong tasks and take in account the load balancing of virtual machine. Reliability, users assert task executed without failure, minimize average response time, minimize makespan time, effective utilization of all Vms and improvement in other parameters is achieved by this phase.
5. Task Submission: Allocation plan from previous phase is received by this phase and after that it allocates each task to virtual machines based on plan.

B. Proposed Algorithm an Flow chart of Efficient particle swarm optimization (EPSO):

Input: No. of tasks K
Output: Average response time and vm utilization
Steps:

For each particle
Initialize particle
END
Do
For each particle, calculate fitness function value
If the fitness value is better than the best fitness value (pBest) in history
Set current value as the pBest
End
Choose the particle with the best fitness value of all the particles as the gBest

For each particle
Calculate particle velocity according to equation (a)
Update particle position according to equation (b)
End
While maximum iterations or minimum error criteria is not attained.

Equation (a) & (b)

V[]=v[]+c1*rand()*(pBest[]-present[])+c2*rand()*(gBest[]-present[])……(a)

present[]=present[]+v[]…….(b)

Fitness function used:

```
double avg = prob.avgET;
double cpuLB=(makespan/avg);
fitness=(1.0/cpuLB);
```

C. Proposed algorithm flow chart:

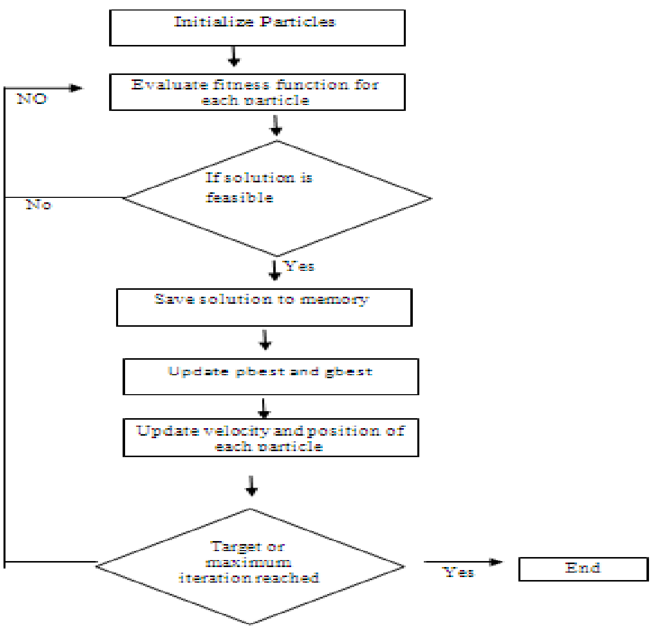

Fig.-10 flow chart of proposed algorithm

VI. EXPERIMENT RESULTS

A simulation program is implemented in JAVA language with the help of CloudSim tool kit to optimize the minimization of average response time and efficient VM utilization for efficient task scheduling in cloud environment.

Results

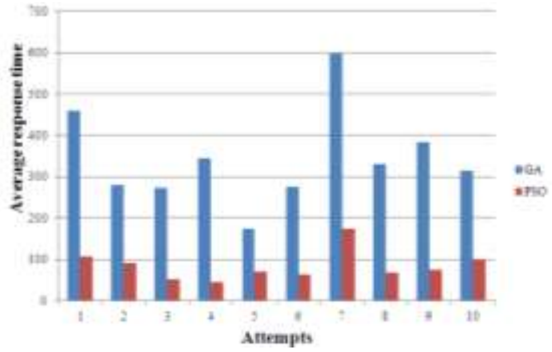

Fig.-11 Graph showing average response time of GA and PSO

This graph shows the average response time of GA and PSO for ten attempts. It is very much clear from the graph then PSO has minimum average response time with respect to the GA. This graph is built for the 20 no. of tasks and 5 VMs which are static. If we take the first entry average response time of GA for 20 tasks is 460.6 and on the other hand average response time of PSO for the same 20 tasks is 107.6. As for efficient task scheduling average response time should be less which is of PSO.

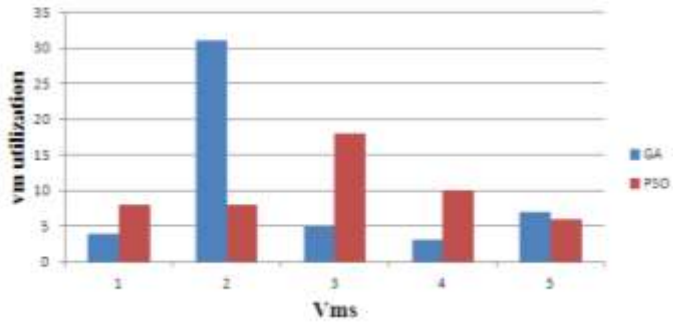

Fig.-12 Graph showing VM utilization by GA and PSO when total no. of tasks is 50

This figure shows the graph between VMs and their algorithm for GA and PSO. There are total 50 tasks and 5 VMs. VMs are static and this graph shows how many times each VM is utilized. The graph clearly shows that PSO gives better result as there is a balance between the utilization of every VMs but in the case of GA 2nd VM is used for about 32 times on the other hand 4th VM is used for 3 times, 1st VM is used for 4 times only.

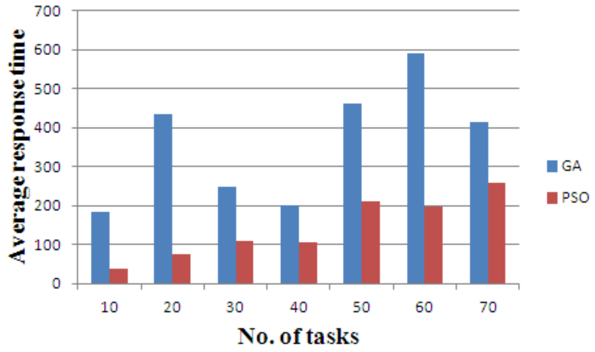

Fig.-13 Graph showing average response time of GA and PSO when no. of tasks is dynamic

This figure shows the graph of average response time of GA and PSO when the no. of tasks is dynamic. This figure shows the same result that PSO has minimum average response time than GA. This figure shows the feasibility of the proposed algorithm that it is feasible for any no. of tasks and gives the same results

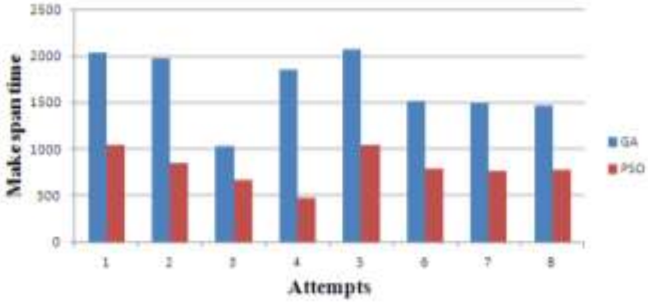

Fig.-14 Graph showing make span of GA and PSO

This figure shows the graph of make span time of GA and PSO for ten attempts. It is very much clear from the graph that PSO has minimum make span time with respect to the GA. This graph is built for the 20 no. of tasks and 5 VMs which are static. If we take the first entry make span time of GA for 20 tasks is 2040.085 and on the other hand makes pan time of PSO for the same 20 tasks is 1040.065. As for efficient task scheduling make span time should be minimum which is of PSO.

VII. CONCLUSION

During the research work we observe many researches done on the optimization of task scheduling using various techniques like Genetic Algorithm and Particle Swarm Optimization.
In this dissertation a novel technique for efficient task scheduling in terms of minimum average response time and effective utilization of VM's in Cloud environment has been proposed.
We have compared the results of GA technique and proposed algorithm under PSO technique for the average response time and VMs utilization. It has been found that PSO give better results in context of response time and utilization of VM's.

ACKNOWLEDGMENT

The author would like to express a deep sense of gratitude and thanks profusely to Ms. Seema Sabharwal Assistant Professor, Department of Computer Science & Engineering, GIMT, Kanipla and Mr. Anurag Jain Associate Professor Department of Computer Science & Engineering, Chandigarh Group of Colleges, Landran, Mohali for his supervision. Without their wise counseling and able guidance, it would have been impossible to complete the research in this manner.

REFERENCES

[1] Mc Evoy, G., & Schulze, B. (2011, December). Understanding scheduling implications for scientific applications in clouds. In Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science (p. 3). ACM.

[2] Mahalle, H. S., Kaveri, P. R., Chavan, V., Goel, I., Garg, V., Garg, S., & Raghuvanshi, D. (2013). Load Balancing On Cloud Data Centres. International Journal, 3(1).

[3] Zhong, H., Tao, K., & Zhang, X. (2010, July). An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems. In ChinaGrid Conference (ChinaGrid), 2010 Fifth Annual (pp. 124-129). IEEE.

[4] Dutta, D., & Joshi, R. C. (2011, February). "A genetic: algorithm approach to cost-based multi-QoS job scheduling in cloud computing environment". In Proceedings of the International Conference & Workshop on Emerging Trends in Technology (pp. 422-427). ACM.

[5] Guo L,Zhao S,Shen S, Jiang C, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm," Journal of Networks(7);2012, pp. 547-553.

[6] Pandey S, Wu L, Guru S, Buyya R,"A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments",24th IEEE International Conference on Advanced Information Networking and Applications (AINA);2010.

[7] Kumar, Manoj, Husian, Mohammad, Upreti, Naveen and Gupta, Deepti (2010) "Genetic Algorithm: Reviewand Application" International Journal of Information Technology and Knowledge Management" vol. 2, 2, pp. 451-454.

[8] Sivanandan S.N. & Deepa S.N. (2008) "Introduction to Genetic Algorithm" Springer Berlin Heidelberg New York.

[9] Guo L,Zhao S,Shen S, Jiang C, "Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm," Journal of Networks(7);2012, pp. 547-553.

[10] http://lancet.mit.edu/mbwall/presentations/IntroToGAs.

[11] http://www.cs.qub.ac.uk/M.Sullivan/ga/ga_index.html.

[12]http://www.obitko.com/tutorials/genetic-algorithms/encoding.php.

[13] Amazon Elastic Computing Cloud, aws.amazon.com/ec2

[14] Amazon Web Services, aws.amazon.com

[15] Xing, Y., & Zhan, Y. (2012). Virtualization and Cloud Computing. In Future Wireless Networks and Information Systems (pp. 305-312). Springer Berlin Heidelberg.

Gifty Garg is a student of M.Tech. computer science engineering in Geeta instituite of management and technology, Kanipla, Kurukshetra. She received her B.Tech. degree from Kurukshetra University, India in 2014. She has published a paper on "basics of cloud computing" in IRJET journal and done research work on "Particle swarm optimization based task scheduling for cloud environment.

Seema Sabharwal is working as Assistant Professor, computer science department in Geeta institute of management and technology Kanipla, Kurukshetra. She is UGC Net qualified conducted by CBSE in Dec, 2015. She has 11 years of teaching experience. She received her B.Tech & M.T.ech degree in computer science from Kurukshetra University, Kurukshetra, India, in 2003 and 2007 respectively. She has guided 3 M.T.ech students. She has about 7 publications in international journal & conferences. Her research interest in the areas of database and cloud computing. Her email id is sabharwalseema@gmail.com.

Anurag Jain is working as Associate Professor, computer science department in Chandigarh engineering college Landran, Mohali. He is in teaching field since 2003. He received his B.Tech. & M.Tech. degree from Kurukshetra University, India, in 2003 and 2009 respectively. He has guided 7 M.Tech. students. He has about 15 publications in international journals and conferences. Currently he is pursuing Ph.D. from M.M. University Mullana Ambala in the area of cloud computing. His research interests are in the areas of load balancing & security in Cloud Computing. His email id is er.anuragjain14@gmail.com.