Dynamic PSO for Task Scheduling Optimization in Cloud Computing

M.S. Sudheer, Dr.M. Vamsi Krishna

Abstract--- Task scheduling is still a challenge in cloud computing as no existing scheduling algorithms are not effectively provisioning and scheduling the resources in the cloud. Existing authors considered only metrics like makespan, execution time and turnaround time etc. and the previous authors concentrated only to optimize the above mentioned metrics. But no existing authors were considered about the effective provisioning of the resources in the cloud i.e, compute, storage and network capacities and still many resources in the cloud were underutilized. In this paper, we want to propose an algorithm which can effectively utilize the resources in the cloud by extending Particle Swarm Optimization by addressing the metrics Bandwidth utilization and Memory utilization particularly. We have simulated this algorithm by using cloudsim and compared the modified Dynamic PSO with the PSO algorithm and it outperforms in terms of Bandwidth and Memory utilization and the makespan is also optimized.

Keywords--- Task Scheduling, Meta heuristic, Cloud Computing, Makespan, Resource Utilization, Bandwidth Utilization, Memory Utilization.

I. INTRODUCTION

According to NIST [1], Cloud Computing can be defined as a paradigm for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources. In Cloud Computing, users can get the resources on demand as per the SLA (Service Level Agreement) made with the cloud provider. The resources in the cloud can be of different categories as storage, memory, compute and network which can be given to the user as-a-service. The architecture of the cloud can be of two types.

- Generic architecture.
- Market oriented architecture.

Generic architecture is a three layered architecture front end facing with an application which can be run from a web browser or mobile device or any type of PDA which can be connected to the internet. The middle tier is the network through which the applications are connected and accessed at any time. The last tier is of the storage devices which are may be virtual or physical devices. Market oriented architecture is a customized version of generic architecture in which the cloud user submit the requests to the broker in turn the broker submits these requests to the service request examiner and admission control which can check whether the user has the ability to get the resources according to the SLA made with the cloud provider. This component can take care about examine the request, pricing, allocation of the resources to the user, Monitoring of VMs. The below layers consists of virtual machines which can be given as a resource to the user on top of physical storage devices.

Scheduling is a huge challenge in cloud computing and it is difficult to provision the resources effectively in the cloud as the workload in the cloud may vary from time to time. Scheduling in cloud computing can be of two types. 1) Task Scheduling and 2) Workflow Scheduling. Task Scheduling can be defined as the assigning the virtualized resources to the particular job for specified amount of time [2]. Workflow scheduling can be defined as assigning the resources to the interdependent tasks in the distributed resources [3]. Scheduling in Cloud can be in terms of computations, Storage, Memory and Network resources and these were effected by different parameters like makespan, throughput, execution time and response time. We cannot say that an algorithm is a best scheduler and resources were effectively provisioned in the cloud based on the above metrics. Most of the authors in the earlier papers considered only the above metrics and said that the scheduler is effective but we want to focus on the effective provisioning in terms of compute, storage and memory not only to optimize the above mentioned metrics. In the existing algorithms makespan is reduced when compared to conventional schedulers but the effective provisioning of the resources in the cloud is questionable. This reason motivated us to propose a new scheduling algorithm which is used to provision the resources in the cloud effectively by using PSO (Particle Swarm optimization) as it is having the faster convergence towards the solution.

The rest of the paper is organized as follows. Section II describes about the Literature Survey, Section III describes about the PSO algorithm, Section IV describes about the Implementation and Results, Section V describes about the Conclusions and future work.

II. LITERATURE SURVEY

This section describes about the existing scheduling algorithms in the cloud computing which uses PSO algorithm to implement their Scheduling strategy. Pandey, Suraj, et.al [4] proposed a scheduling algorithm to schedule the workflow applications which considers the total execution cost and as well as the data transfer cost and communication cost by using PSO and this algorithm is compared with the Best resource selection algorithm and the proposed algorithm is outperformed for the above mentioned metrics in terms of the saving of the cost for 3 times and the distribution of the workload onto the resources. Zhan, Shaobin et.al, [5] proposed a new scheduling algorithm which can be used to reduce the task average running time and increases the rate availability of resources. For this they have used simulated annealing

Manuscript received September 16, 2019.

M.S. Sudheer, Research Scholar, Department of CSE, Centurion University of Technology and Management, Parlakhemudi, Odisha, India. (email: mssudheer2015@gmail.com)

Dr.M. Vamsi Krishna, Professor, Department of CSE, Chaitanya Engineering College, Kakinada. (email: vkmangalampalli@gmail.com)

algorithm to be added to the Particle Swarm Optimization algorithm which increases the convergence speed of the algorithm. Guo, Lizheng, et al. [6] proposed an algorithm which can be used to reduce the processing cost. The authors were compared this proposed algorithm with the PSO embed in SPV and PSO algorithm embed in crossover and mutation (CM-PSO) and PSO embed in local search (L-PSO) and they got the processing cost to be reduced and have shorter running time for PSO rather than the above mentioned algorithms.

Zuo, Xingquan, Guoxiang Zhang, and Wei Tan et al [7], proposed a framework for resource allocation for IaaS hybrid clouds when IaaS provider cannot have the resources and need to meet the demands. This can be done by outsourcing the tasks to the external clouds. This has to be done automatically by self-learning approach which was proposed using PSO with the IP (integer programming) model i.e, SLPSO by considering the each task as a particle and as a whole represents all the tasks priorities. With this approach authors have achieved a good resource allocation in IaaS clouds by meeting the QoS requirements.

Jena, R. K. et.al [8], proposed a multiobjective scheduling framework to optimize the energy and processing time. This model was simulated on cloudsim simulator and it was compared with best resource selection algorithm and random scheduling algorithm. The results showed that the proposed model reduced 30 percent of energy consumption and 25 percent of the makespan are reduced and this algorithm also reduces the number of failed tasks.Abdi, Solmaz, Seyyed Ahmad Motamedi, and Saeed Sharifian.et.al,[9] proposed a scheduling algorithm i.e, modified PSO in which SJFP(shortest Job to faster processor) included to PSO which can be used to minimize the makespan. It was compared with the genetic algorithm and standard PSO but the proposed algorithm reduced the makespan compared with the other two algorithms.

Al-Maamari, Ali, and Fatma A. Omara et.al,[10] proposed a task scheduling algorithm called as MDAPSO which combines both dynamic PSO and cuckoo search algorithm which can be used to minimize the makespan and increase the utilization ratio of workflow in cloud computing. This was implemented in cloudsim and evaluated the performance of MDAPSO and compared with the PSO and DAPSO and the proposed algorithm is outperformed with respect to the metrics makespan and resource utilization.

Awad, A. I., N. A. El-Hefnawy, and H. M. Abdel_kader et.al, [11] proposed a task scheduling algorithm which uses Load Balancing Mutation PSO based allocation model based on the parameters reliability and availability. This LBMPSO is implemented in cloudsim with 6 data centers with 50 VMs and 1000 tasks. This algorithm achieves the reliability by rescheduling the failure task by considering the resource availability. This LBMPSO was compared with the standard PSO, LCFP algorithms but it outperforms in terms of makepan, roundtrip time and transmission cost. Kumar, Dinesh et.al, [12] proposed a scheduling model using PSO which focuses on minimizing the Resource wastage and the usage of servers. It is used to allocate the virtual machines to the corresponding physical machines based on the resource wastage and the number of servers used. It is evaluated with respect to the performance and scalability and the results were compared with the different server placement models i.e., Best-Fit, First-fit and Worst-fit. Chitra, S., et al. [13] proposed a new enhanced PSO to speed up the convergence of the search criteria. This algorithm mainly focuses on to speedup ratio, makespan and load balancing of the system. The proposed algorithm was evaluated and compared with GA and standard PSO it outperforms GA and PSO in the aspects of speedup ratio, makespan and load balancing.

Verma et al [14] proposed a new scheduling framework for mapping of workflow applications to the cloud resources. These authors proposed an algorithm previously i.e, BPSO (Biobjective Priority PSO) to schedule the workflow applications by minimizing the execution cost and meeting the deadline constraints. Now these authors have extended the BPSO by adding cost time balance factor in the fitness function. The proposed algorithm is simulated by using synthetic workflows and it is compared against BHEFT and BTGA algorithms and the extended BPSO outperforms over the existing algorithms. Manasrah et.al [15], developed a new workflow scheduling algorithm i.e, GA-PSO to efficiently allocate the tasks to the resources. The main aim of this algorithm is to minimize the makespan and the execution cost and to fairly distribute the load among the resources. This simulation was done in workflowsim simulator and the results shown that the total execution time for the tasks was greatly reduced over GA, PSO, HSGA and WSGA algorithms.

Cho, Keng-Mao, et al., [16] proposed a new scheduling algorithm which combines both ACO and PSO to schedule the resources in the cloud. This algorithm will predict the new requests based on the history of the workloads and make a decision whether this workflow is suited to the resource. If the workflow is not suitable to the cloud resource it will pre-reject the workflow requests. With this use of algorithm, computing time of the scheduling can be reduced. Liu, Li, et al. [17] proposes a scheduling algorithm which describes about the penalty function for the constraints and coevolution methodology is used to adjust the crossover and mutation probabilities to get the faster convergence. This algorithm was simulated in work flowsim simulator and compared with other genetic algorithms HEFT and PSO and it competes with the other algorithms by minimizing the total execution cost and meeting the deadline constraints.

Khalili, Azade et al. [18] proposed an algorithm which uses LDIW (Linear Descending Inertia Weight) model to schedule the tasks to the resources which can minimizes the makespan. This algorithm was simulated in cloudsim and each particle is iterated for 1000 times and the number of particles is 100. This algorithm is simulated with different inertia weights and out of those LDIW has outperformed in terms of makespan. Kumari et.al,[19] proposed a scheduling algorithm which uses the combination of PSO and GA by switching the tasks based on the priority to foreground and background. The experiments are simulated in the cloudsim and compared with the BATS and CMCBF algorithms and the proposed algorithm is outperformed in terms of resource utilization. Abdullahi et.al,[20] proposed a scheduling algorithm which can be used to schedule the tasks in the cloud in an optimized way by combining simulated annealing and symbiotic organism search algorithms i.e, SASOS approach. The proposed approach is simulated in Cloudsim. The proposed algorithm outperforms in terms of convergence speed, makespan and response time than SOS. He, Hua, et al, [21] proposed a scheduling algorithm which is used to achieve multiple objectives by using Adaptive acceleration by extending PSO algorithm to maintain the particle diversity and to get the resource utilization and to reduce the average task running time. The algorithm was simulated on cloudsim. It gives the optimal solutions in task running time, cost and running time. Zhou, et al.,[22] proposed a task scheduling algorithm which can be used to address the energy consumption problem in cloud computing. The proposed algorithm was simulated on the cloudsim. It was compared with the other algorithms and the proposed algorithm reduces the total processing cost.

Kumar, et.al,[23] proposed a new strategy for PSO algorithm by changing the inertia weights using the modified simple random inertia weight and modified oscillating inertia weight techniques. This strategy was implemented by using the cloudsim simulator. The proposed strategy proved to be have better convergence than the normal PSO algorithm. Ibrahim, et.al,[24] proposes a new algorithm to reduce the makespan and to decrease the price of executing the independent tasks. The proposed algorithm was implemented by using the amazon and google cloud pricing models and this approach reduces the makespan by 20 % and pricing by 67% approximately. Kumaret.al, [25] proposed a new scheduling algorithm by combining the GA and PSO by to allocate the tasks to the resources in an optimized way. The proposed algorithm maintains a queue manager in which the tasks are stored. Tasks are allocated to the resources based on the demand in the queue manager. With this proposed approach scalability and availability were improved when compared to GA and PSO algorithms.

Dordaie.et.al, [26] proposed a new framework to schedule the tasks to the corresponding resources by using the combination of PSO and Hill climbing algorithms to optimize the makespan by using random DAGS. Almezeini, Nora et.al, [27] devised a new scheduling model which is aimed to reduce the makespan and the degree of imbalance. The proposed algorithm was compared with the standard GA and PSO and the proposed model outperformed the both algorithms in terms of makespan and degree of imbalance. Kumar et.al,[28] proposed a scheduling algorithm not only to allocate the resources but also to reduce the energy consumption in an efficient way by using the PSO-COGENT algorithm. The proposed algorithm was simulated on cloudsim. It reduces the makespan, execution time, energy consumption and increases the throughput when compared with PSO algorithm.

Agarwal et.al,[29] proposed a scheduling algorithm which can be used to allocate the resources in the cloud to the tasks by using PSO algorithm. The main aim of this algorithm is to minimize the overall response time. This algorithm was simulated by using cloudsim and the proposed approach outperforms when compared with GA and Greedy approaches in terms of overall response time. Saleh, Heba, et al.[30] proposed a new scheduling framework which can be used to increase the resource utilization by dividing the tasks into subtasks. This was simulated on the cloudsim and the proposed algorithm achieved a good results for the makespan, standard deviation and degree of imbalance when compared with the Honey bee and ant colony algorithms.

Table 1: Summary of Cloud Scheduling Algorithms

Study	Algorithm Used	Type of Scheduling	Experiment Type	Tools Used
[4]	PSO Algorithm	Workflow Scheduling	Mathematical Model	---
[5]	SA and PSO algorithms	Task Scheduling	Simulation	Cloudsim
[6]	PSO algorithm	Task Scheduling	Simulation	Cloudsim
[7]	PSO Algorithm	Task Scheduling	Simulation	Cloudsim
[8]	PSO Algorithm	Task Scheduling	Simulation	Cloudsim
[9]	Modified PSOAlgorithm	Task Scheduling	Simulation	Cloudsim
[10]	MDAPSO algorithm	Task Scheduling	Simulation	Cloudsim
[11]	LBMPSO algorithm	Task Scheduling	Simulation	Cloudsim
[12]	PSO Algorithm	Task Scheduing	Mathematical Model	----
[13]	Enhanced PSO algorithm,	Task Scheduling	Mathematical Model	---
[14]	BPSO Algorithm	Workflow Scheduling	Mathematical Model	---
[15]	GA-PSO Algorithm	Workflow Scheduling	Simulation	Workflowsim
[16]	ACO and PSO Algorithms	Workflow Scheduling	Simulation	Workflowsim
[17]	Modified PSO algorithm	Workflow Scheduling	Simulation	Workflowsim
[18]	LDIW model	Task Scheduling	Simulation	Cloudsim
[19]	PSO and GA Algorithms	Task Scheduling	Simulation	Cloudsim
[20]	Symbiotic Organism Search	Task Scheduling	Simulation	Cloudsim
[21]	Extended PSO	Task Scheduling	Simulation	Cloudsim
[22]	modified PSO	Task Scheduling	Simulation	Cloudsim
[23]	PSO Algorithm	Task Scheduling	Simulation	Cloudsim
[24]	modified PSO	Task Scheduling	Real time implementation	Amazon EC2 and Google cloud
[25]	GA and PSO	Task Scheduling	Mathematical Model	---
[26]	PSO and Hill Climbing Algorithms	Task Scheduling	Mathematical Model	---
[27]	Modified PSO	Task Scheduling	Mathematical Model	---
[28]	Modified PSO	Task Scheduling	Simulation	Cloudsim
[29]	PSO Algorithm	Task Scheduling	Simulation	Cloudsim
[30]	IPSO Algorithm	Task Scheduling	Simulation	Cloudsim

In the table 1, we have mentioned the different scheduling algorithms in cloud computing, the kind of scheduling whether it is task flow or workflow scheduling and through which tool the corresponding algorithm is implemented as mentioned in the above table. From the table 1 we can also identify that most of the authors were using PSO model to schedule the resources in the cloud computing as it have faster convergence but it is also have a disadvantage i.e, local optimum problem so that the rate of convergence is slow when the workload is dynamic and heterogeneous[31]. Due to this reason we have chosen another heuristic approach that dynamically behaves according to the workload and the environment in the cloud.

Table 2: Comparison of Scheduling Algorithms

	Metrics								
Algorithms	**Reduces Makespan**	**Reduces Communication cost**	**ReducesAvg. Waiting Time**	**Reduces Computational cost**	**Improves Memory Utilization**	**Improves QOS**	**Reduces execution Cost**	**Avg Migration**	**Electricity price per unit**
[4]	-	-	-	√	-	-	√	-	-
[5]	-	-	-	-	-	-	√	-	-
[6]	-	-	-	-	-	-	-	-	-
[7]	-	-	-	-	-	√	-	-	-
[8]	√	-	-	-	-	-	-	-	-
[9]	√	-	-	-	-	-	-	-	-
[10]	√	-	-	-	-	-	-	-	-
[11]	√	-	-	-	-	-	-	-	-
[12]	√	-	-	-	-	-	-	-	-
[13]	-	-	-	-	-	-	-	-	-
[14]	-	-	-	-	-	-	√	-	-
[15]	√	-	-	-	-	-	√	-	-
[16]	-	-	-	-	-	-	-	-	-
[17]	-	-	-	-	-	-	√	-	-
[18]	√	-	-	-	-	-	-	-	-
[19]	-	-	-	-	-	-	-	-	-
[20]	√	-	-	-	-	-	-	-	-
[21]	-	-	-	√	-	-	√	-	-
[22]	-	-	-	-	-	-	-	-	-
[23]	-	-	-	-	-	-	-	-	-
[24]	√	-	-	-	-	-	-	-	-
[25]	-	-	-	-	-	-	-	-	-
[26]	√	-	-	-	-	-	-	-	-
[27]	√	-	-	-	-	-	-	-	-
[28]	√	-	-	-	-	-	-	-	-
[29]	-	-	-	-	-	-	-	-	-
[30]	√	-	-	-	-	-	-	-	-

In table 2, most of the scheduling algorithms addressed the metrics like makespan, execution cost and total processing cost. The earlier authors were not addressed the metrics like Bandwidth utilization, Memory utilization.

Initially we have chosen PSO algorithm to model the scheduling approach to address the new metrics like Bandwidth Utilization and Memory utilization but when the workload in the cloud is high and dynamic because of this reason PSO may not be a proper choice to model the scheduling algorithm.

Due to this reason we have chosen a new heuristic approach which is called as Modified Dynamic Adaptive PSO which is a combination of DAPSO and Cuckoo search algorithms [10].

III. PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

PSO is a nature inspired bird flocking algorithm [19] which can be used to find the optimal solution in search space. In this algorithm, each solution is to be assumed as a particle and these particles are searching for the optimized solution and it is assumed as food in this scenario. Each particle is having their own fitness values which are to be calculated to get the optimal solution. It can be calculated by updating the position and velocity in each iteration of the algorithm. It is initialized with random particles and then searches for the optimal solution by updating the two best values i.e, local best and global best. The particle swarm optimizer checks for the so far best solution achieved by any particle are to be represented as global best i.e., gbest. Particles which achieve a best solution achieved so far is to be represented as pbest. In every iteration [32] after finding the values particles updates its velocity and positions according to the following equations.

$$Velocity[\,] = velocity[\,] + K1 * random(\,) * (pbest[\,] - present[\,]) + K2 * random(\,) * (gbest[\,] - present[\,])\text{------ }1$$

$$present[\,] = present[\,] + velocity[\,]\text{ ---------}2$$

In the equations 1 and 2 K1 and K2 are learning factors and random () is a random number [32] between (0, 1)

Dynamic Adaptive PSO

The disadvantage of PSO is that it can't explore in the local search spaces but it can globally search the problem space by evolving the swarm. Dynamic Adaptive PSO updates the position and velocity of the particles by adding the inertia weight to the PSO to explore the search space in a optimal way. The inertia weight can be added in the PSO and can be represented with the equation as in [33].

Cuckoo Search Algorithm

Cuckoo Search is an optimization algorithm which can combine the exploration for local search and Global search which was given in [35]. The Cuckoo species birds will lay the eggs in the nests randomly.

These nests are to be considered as the solutions in the algorithm. The solutions can be updated by random walk to the nests and if any better solution is available from the previous solution then it will be replaced by the new solution.

After this replacement process all the nests are to be given the ranks according to the fitness and the nests with the worst fraction are replaced with the new solutions [36]. The random walk through the nests can be done by Levy flights.

Through this Levy flights cuckoo search have the advantageous over GA and PSO because cuckoo search have heavy tailed mechanism and it can step towards the solutions with the random jump and it is a population-based algorithm and the number of parameters tuned compared to the other algorithms were less so that it can be easily combine the local and global search exploration in the search space [36].

Modified Dynamic Adaptive PSO

With the disadvantage of local optimum problem in the basic PSO algorithm a new modified dynamic adaptive PSO approach was used [10].

Basically PSO algorithm is a global search algorithm and it searches for the solution globally. It may get stuck into the problem at the last iteration of the PSO so that it may not recognize the local subspaces in the search space.

Here we are combining Dynamic Adaptive PSO and cuckoo search algorithm for the efficient local search spaces. Initially the algorithm runs with PSO as the main algorithm and for the last iteration to avoid the convergence problem Dynamic Adaptive PSO will be used because it have the fine tuning of inertia [33].

Dynamic Adaptive PSO has also a disadvantage that whenever the inertia weight is decreased in the algorithm, it can't explore to search the new areas. To avoid this problem we need to combine Dynamic Adaptive PSO with Cuckoo search algorithm to find the local optimum and global optimum solutions easily even the workload is heterogeneous and dynamic.

Algorithm

1. Initialize the population randomly.
2. Calculate the fitness function for the particles.
3. Compare the fitness values of each particle then identify the local best of the particle in the population if local best of any particle is better than the current local best update the local best and set the current best of the particle as local best.
4. Compare the global best of the particles and if any particle is having the optimized value rather than the current global best update the global best.
5. Update the particles using the position and velocity equations in the PSO algorithm and also update the inertia weight using Dynamic Particle Swarm optimization and run the last iteration by combining cuckoo search algorithm to get the particles convergence towards the solutions globally and locally.

Metrics Addressed

In this paper we want to address the metrics like Bandwidth utilization and Memory utilization in particular because we can provision the resources in an effective way in the cloud if the bandwidth of the network is utilized properly and utilization of the memory is provisioned efficiently because if a process has to be allocated with a memory the corresponding operating system needs to scan the free pages.

If there are no free pages then it have to choose the pages which were already used by any process.

Bandwidth utilization can be calculated as

$$Bandwidth\ Utilization\,(Bw) = \frac{Required\ bandwidth\ for\ all\ tasks}{Total\ Available\ Bandwidth}$$

Memory utilization can be calculated as

Memory Utilization= Total Memory - (Free Memory + Buffer + Cached Memory)

IV. EXPERIMENTS AND RESULTS

Cloudsim 3.0.3[34] is used as a simulator to analyze the algorithm with our metrics i.e, Bandwidth Utilization and Memory utilization. It is an open source simulator which can be used to simulate the cloud in both windows and Linux environments. It was developed at the University of Melbourne. We have used the windows environment to run this simulation with Intel i3 processor, 500 GB hard disk.

To evaluate this algorithm we have chosen 10 Data centers and the number of cloudlets we have chosen as 100, 200, 300 and 400.

The Bandwidth utilization of PSO and modified Dynamic PSO were compared in the below table.

Table 3: Comparison of PSO and Modified Dynamic PSO for Bandwidth Utilization (percentage)

PSO	Modified Dynamic PSO	Cloudlets
0.54	0.67	100
0.65	0.77	200
0.76	0.80	300
0.82	0.89	400

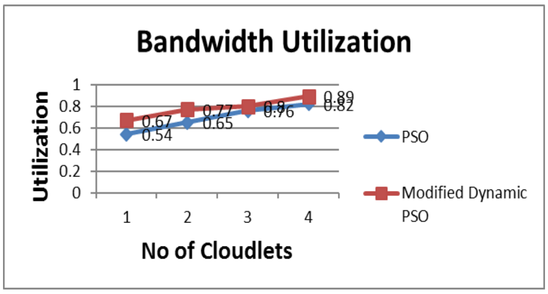

Figure 1: Bandwidth Utilization of PSO and Modified Dynamic PSO

From the above Table 3 we can observe that the bandwidth utilization of PSO and Modified Dynamic PSO for the cloudlets i.e., 100, 200, 300 and 400 respectively. From the above Figure 1 we can clearly notice that Modified Dynamic PSO is outperformed PSO in terms of Bandwidth Utilization with 10% tradeoff.

Table 4: Comparison of PSO and Modified Dynamic PSO for Memory Utilization (percentage)

PSO	Modified Dynamic PSO	Cloudlets
0.65	0.72	100
0.72	0.8	200
0.76	0.84	300
0.82	0.92	400

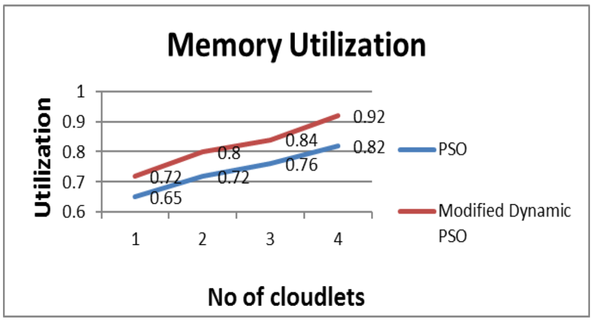

Figure 2: Memory Utilization of PSO and Modified Dynamic PSO

From the Table 4 we have identified the Memory utilization of PSO and Modified Dynamic PSO and Compared with each other. The Modified PSO is outperformed when compared with the basic PSO in terms of Memory utilization with 9% tradeoff.

Table 5: Comparison of PSO and Modified Dynamic PSO for Makespan (Seconds)

PSO	Modified Dynamic PSO	Cloudlets
2.76	1.45	100
5.24	3.27	200
7.42	5.72	300
9.62	7.25	400

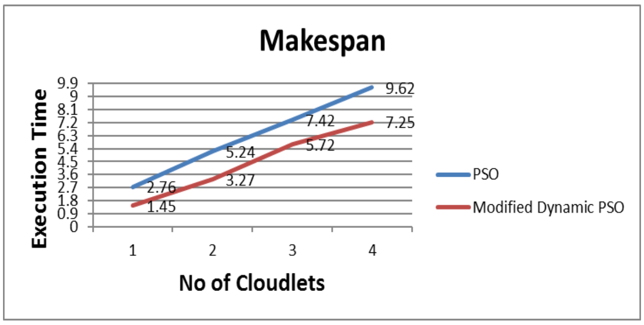

Figure 3: Makespan of PSO and Modified Dynamic PSO

Table 5 represents the Makespan of the cloudlets were scheduled through both PSO and Dynamic PSO. From the above figure we can clearly notice that Modified Dynamic PSO is outperformed PSO in terms of Makespan.

V. CONCLUSION AND FUTURE WORK

Task Scheduling is the major challenge in cloud computing as the workload in the cloud is heterogeneous and dynamic [31].To effectively provision the resources and schedule the tasks in the cloud we have used a Modified Dynamic PSO algorithm which schedules the tasks in an effective way in terms of increasing the utilization of Bandwidth and Memory and the makespan is also optimized.

We have compared Modified Dynamic PSO for the above mentioned metrics with the PSO and the Modified Dynamic PSO is outperformed in terms of Bandwidth and Memory utilization and Makespan is also optimized. In future we want to add the metrics like Average Migration time and Electricity price per unit cost in our work to check the behavior of the algorithm.

ACKNOWLEDGMENTS

I would like to thank Dr K Ganesh Reddy who helped me a lot to do the research in Cloud Computing and driven me to do the research in a systematic way. I would also like to thank Dr P Kiransree who encouraged me to do the research and Head of the department of CSE and staff of CSE SVECW to carry out my research in a smoother way.

REFERENCES

1. https://www.nist.gov/news.../final-version-nist-cloud-computing definition-published.
2. Sudheer, M. S., et al. "An effective analysis on various scheduling algorithms in cloud computing." *Inventive Computing and Informatics (ICICI), International Conference on*. IEEE, 2017.
3. Yu, Jia, Rajkumar Buyya, and Kotagiri Ramamohanarao. "Workflow scheduling algorithms for grid computing." *Metaheuristics for scheduling in distributed computing environments*. Springer, Berlin, Heidelberg, 2008. 173-214.
4. Pandey, Suraj, et al. "A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments." *2010 24th IEEE international conference on advanced information networking and applications*. IEEE, 2010.

5. Zhan, Shaobin, and Hongying Huo. "Improved PSO-based task scheduling algorithm in cloud computing." *Journal of Information & Computational Science* 9.13 (2012): 3821-3829.
6. Guo, Lizheng, et al. "Task scheduling optimization in cloud computing based on heuristic algorithm." *Journal of networks*7.3 (2012): 547.
7. Zuo, Xingquan, Guoxiang Zhang, and Wei Tan. "Self-adaptive learning PSO-based deadline constrained task scheduling for hybrid IaaS cloud." *IEEE Transactions on Automation Science and Engineering* 11.2 (2013): 564-573.
8. Jena, R. K. "Multi objective task scheduling in cloud environment using nested PSO framework." *Procedia Computer Science* 57 (2015): 1219-1227.
9. Abdi, Solmaz, Seyyed Ahmad Motamedi, and Saeed Sharifian. "Task scheduling using modified PSO algorithm in cloud computing environment." *International conference on machine learning, electrical and mechanical engineering*. 2014.
10. Al-Maamari, Ali, and Fatma A. Omara. "Task scheduling using PSO algorithm in cloud computing environments." *International Journal of Grid and Distributed Computing* 8.5 (2015): 245-256.
11. Awad, A. I., N. A. El-Hefnawy, and H. M. Abdel_kader. "Enhanced particle swarm optimization for task scheduling in cloud computing environments." *Procedia Computer Science*65 (2015): 920-929.
12. Kumar, Dinesh, and Zahid Raza. "A PSO based VM resource scheduling model for cloud computing." *2015 IEEE International Conference on Computational Intelligence & Communication Technology*. IEEE, 2015.
13. Chitra, S., et al. "Local minima jump PSO for workflow scheduling in cloud computing environments." *Advances in computer science and its applications*. *Springer,* Berlin, Heidelberg, 2014. 1225-1234.
14. Verma, Amandeep, and Sakshi Kaushal. "Cost minimized PSO based workflow scheduling plan for cloud computing." *Int J Inf Technol Comput Sci* 8 (2015): 37-43.
15. Manasrah, Ahmad M., and Hanan Ba Ali. "Workflow scheduling using hybrid GA-PSO algorithm in cloud computing." *Wireless Communications and Mobile Computing*2018 (2018).
16. Cho, Keng-Mao, et al. "A hybrid meta-heuristic algorithm for VM scheduling with load balancing in cloud computing." *Neural Computing and Applications* 26.6 (2015): 1297-1309.
17. Liu, Li, et al. "Deadline-constrained coevolutionary genetic algorithm for scientific workflow scheduling in cloud computing." *Concurrency and Computation: Practice and Experience* 29.5 (2017): e3942.
18. Khalili, Azade, and Seyed Morteza Babamir. "Makespan improvement of PSO-based dynamic scheduling in cloud environment." *2015 23rd Iranian Conference on Electrical Engineering*. IEEE, 2015.
19. Kumari, K. Raja, P. Sengottuvelan, and J. Shanthini. "A hybrid approach of genetic algorithm and multi objective PSO task scheduling in cloud computing." *Asian Journal of Research in Social Sciences and Humanities* 7.3 (2017): 1260-1271.
20. Abdullahi, Mohammed, and Md Asri Ngadi. "Hybrid symbiotic organisms search optimization algorithm for scheduling of tasks on cloud computing environment." *PloS one* 11.6 (2016): e0158229.
21. He, Hua, et al. "AMTS: Adaptive multi-objective task scheduling strategy in cloud computing." *China Communications* 13.4 (2016): 162-171.
22. Zhou, Zhou, et al. "A modified PSO algorithm for task scheduling optimization in cloud computing." *Concurrency and Computation: Practice and Experience* 30.24 (2018): e4970.
23. Kumar, Nagresh, and Sanjay Kumar Sharma. "Inertia weight controlled PSO for task scheduling in cloud computing." *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*. IEEE, 2018.
24. Ibrahim, Elhossiny, Nirmeen A. El-Bahnasawy, and Fatma A. Omara. "Task scheduling algorithm in cloud computing environment based on cloud pricing models." *2016 World Symposium on Computer Applications & Research (WSCAR)*. IEEE, 2016.
25. Kumar, AM Senthil, and M. Venkatesan. "Task scheduling in a cloud computing environment using HGPSO algorithm." *Cluster Computing* (2018): 1-7.
26. Dordaie, Negar, and Nima Jafari Navimipour. "A hybrid particle swarm optimization and hill climbing algorithm for task scheduling in the cloud environments." *ICT Express* (2017).
27. Almezeini, Nora, and Alaaeldin Hafez. "Task scheduling in cloud computing using lion optimization algorithm." *algorithms*5 (2017): 7.
28. Kumar, Mohit, and S. C. Sharma. "PSO-COGENT: Cost and Energy Efficient scheduling in Cloud environment with deadline constraint." *Sustainable Computing: Informatics and Systems* 19 (2018): 147-164.
29. Agarwal, Mohit, and Gur Mauj Saran Srivastava. "A PSO Algorithm-Based Task Scheduling in Cloud Computing." *Soft Computing: Theories and Applications*. Springer, Singapore, 2019. 295-301.
30. Saleh, Heba, et al. "IPSO task scheduling algorithm for large scale data in cloud computing environment." *IEEE Access* 7 (2018): 5412-5420.
31. Sudheer, Vamsi Krishna, et al. "A Review on Meta heuristic Scheduling algorithms in Cloud Computing". *Indian Journal of science and Technology*, Volume 12, Issue 23, 2019. DOI: 10.17485/ijst/2019/v12i23/145365.
32. http://www.swarmintelligence.org/tutorials.php
33. Shen, Xianjun, et al. "Particle swarm optimization with dynamic adaptive inertia weight." *2010 International Conference on Challenges in Environmental Science and Computer Engineering*. Vol. 1. IEEE, 2010.
34. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", Software: Practice and Experience, vol. 41, (2011), pp. 23-50.
35. X. -S. Yang and S. Deb, "Cuckoo search via Lévy flights", in Nature & Biologically Inspired Computing, 2009. NaBIC 2009. World Congress on, (2009), pp. 210-214.
36. X. -S. Yang and S. Deb, "Engineering optimization by cuckoo search", *International Journal of Mathematical Modelling and Numerical Optimization*, vol. 1, (2010), pp. 330-343.