ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Priority Particle Swarm Optimization Based Load Balancing in Cloud Computing

Rachna Yadav [1], Mayank Namdev[2]

P.G. Student, Department of Computer Science & Engineering, SIRT, Bhopal, India[1]

Professor, Department of Computer Science & Engineering, SIRT, Bhopal, India[2]

ABSTRACT: Cloud computing is a technology that facilitates tasks by allocating virtual machine (VM) dynamically. There are so many challenges faced by cloud provider. One of the challenge is load balancing. Load balancing is a significant area of cloud computing environment which ensures that all connected devices or processors carry out same amount of work in equal time. With an aim to make cloud resources and services accessible to the cloud user easily and conveniently, different algorithms and models for load balancing in cloud computing is being developed. There are so many algorithms are available for proper load balancing but in this paper particle swarm based algorithm is focused that can balance the load in cloud computing so that resources are easily available for users. In this research work focus is on designing of task scheduling algorithm based on priority based particle swarm optimization technique that can balance load efficiently in cloud computing so that resources are easily available for users.

KEYWORDS: Cloud Computing, Load Balancing, Particle Swarm Optimization, Execution Factors, Execution Time.

I. INTRODUCTION

Cloud computing is a new era technology that provides resources and services online to the end users. We need computers in our daily life everywhere. As the uses of computing resources are increasing day by day, the computing resources that we need also go up. For establishment of small enterprises that uses computing resources, affordability is a huge factor. With establishment of such infrastructure there arise problems like machine failure, hard drive crashes, hacking, bugs, etc. These might create big issues. Cloud computing [1] offers a solution to these issues. It renders the cost of purchasing computational resources and storage as everything is available on cloud and delivered as per demand of user and we have to pay for that only for which we want to use. Only requirement of cloud computing is that we need to have internet connection available to enjoy their computational services. The term cloud is a metaphor of “Internet” that shifted the computing from individual application server to cloud of computers or resources. It is virtualized server pool which can provide different computing resources or services to end users or clients.

Basically, the cloud computing has inherited these concept from Grid, Cluster and Utility Computing. All computing resources are aggregated into a packet and delivered as a service over the internet as in the form of cloud. Numerous companies and research organizations are applying cloud computing concepts to their business including GOOGLE, AMAZON and AZURE [1,2].

Load Balancing [3-8] is method of reallocating the entire load to the distinct nodes of the collective system to make resource utilization effective and to advance the latency of the job response time, all together eliminating a circumstance during which number of the nodes are over loaded whereas some others are under loaded. Accordingly Load balancing is an especially technique that enables networks and assets by means of supplying a most throughput with minimal reaction time through dividing the traffic between servers

The assignment of a task by the scheduler is subjected to a number of constraints. Constraints are typically either time constraints or resource constraints. A task may include data entry and processing, software access, and storage functions. The datacenter classifies tasks according to the service-level agreement and requested services. Each task is then assigned to one of the available servers. In turn, the servers perform the requested task. A response or result is transmitted back to the user [9].

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Scheduling is a balancing scenario in which processes or tasks are scheduled as per the given requirements and used algorithm. The goal of scheduling algorithms in distributed systems is to spread the load on the processors and to maximize their utilization while minimizing total task execution time. Job scheduling, one of the most famous optimization problems, plays a key role to improve flexible and reliable systems [10-12]. The main purpose is to schedule jobs to the adaptable resources in accordance with adaptable time, which involves finding out a proper sequence in which jobs can be executed under transaction logic constraints.

In Cloud Computing VM scheduling algorithms are used to schedule the VM requests to the Physical Machines (PM) of the particular Data Center (DC) as per the requirement fulfilled with the requested resources (i.e. RAM, Memory, Bandwidth etc). In today's era there are so many cloud providers in market that have different capacity of Data Centers and Physical Machines available. In general scheduling algorithm works in three levels as given below [13]:

For the set of VMs find the appropriate Physical Machine.

Determine the proper provisioning scheme for the VMs.

Schedule the tasks on the VMs

The scheduling model in a cloud datacenter consists of four components, namely, computing entity, job scheduler, job waiting queue, and job arrival process [14].

Computing entity is provided through the implementation of a virtualization technique in the cloud computing system. A number of virtual machines that provide computing facilities, such as the operating system and software, are present in the cloud system to process the submitted tasks. A computing entity is characterized by its computing capacity, which indicates the number of instructions it can process in a second [15-17].

Job scheduler is an important component of the scheduling process in a cloud computing.

Job waiting queue is the line of jobs for execution waiting to get assigned to a particular machine.

Job arrival process is the procedure in which jobs arrive into the scheduling system.

II. METHODOLOGY

Though Cloud computing is dynamic however at any specific instance the aforementioned drawback of load balancing can be formulated as allocating N number of jobs submitted by cloud users to M number of processing units i.e. virtual machines within the Cloud. Load Balancer is employed for distribution load to different virtual machines in such the way that none of the nodes gets loaded heavily or lightly. The load balancing has to be done properly as a result of failure in any one of the node will cause unavailability of data. After analysing different algorithms proposed methodology is designed for load balancing of virtual machine.

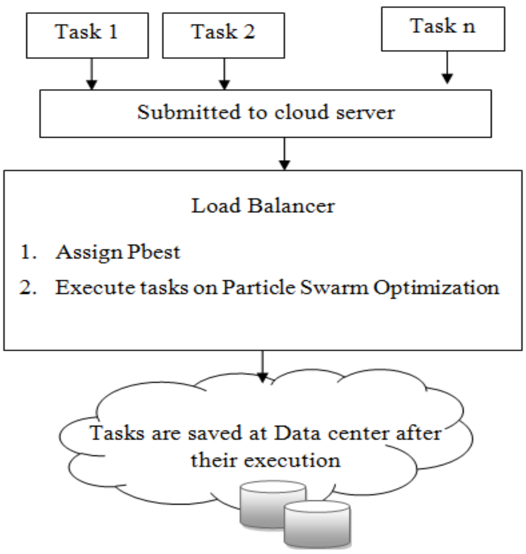

Figure 1: Proposed Architecture

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Load balancing splits the services in an equal fashion in each VM of the cloud. If new request or load is added or existing resources are removed then load balancer follows an order. This proposed method is useful in case of limited numbers of resources available in the VM cluster. To improve efficiency this research work have developed a new Load Balancing algorithm which is capable of holding users requests that doesn't get virtual machine to run on with minimum delay. Concept of priority PSO based load balancing algorithm is discussed in below sections. The proposed modified priority PSO based load balancing algorithm is based on three attributes of tasks submitted for scheduling purpose: User_type, Expected_priority, Task_length Task Length (Expected_Execution_Time. Figure 1 shows the structural diagram of proposed architecture. Load balancer maintains a ready queue and waiting queue. These queues are used to hold the requests which are switched from wait state to execution state. As the tasks arrives at the cloud end for its execution, it is first send to waiting queue and pbest is calculated for its execution.

A. Proposed Algorithm

The proposed algorithm is as given below:

Step 1: Start

Initialize the cloud simulation and assign the data center configuration, virtual machines configuration as well as host configuration.

Step 2: Submit arrived task in waiting queue.

Step 3: Send task into ready queue according to assigned fitness function according to PPSO algorithm which is discussed below section 3.3.

Step 4: Send task into ready queue according to assigned fitness function.

Step 5: Execute task

Step 6: Continue to step 2

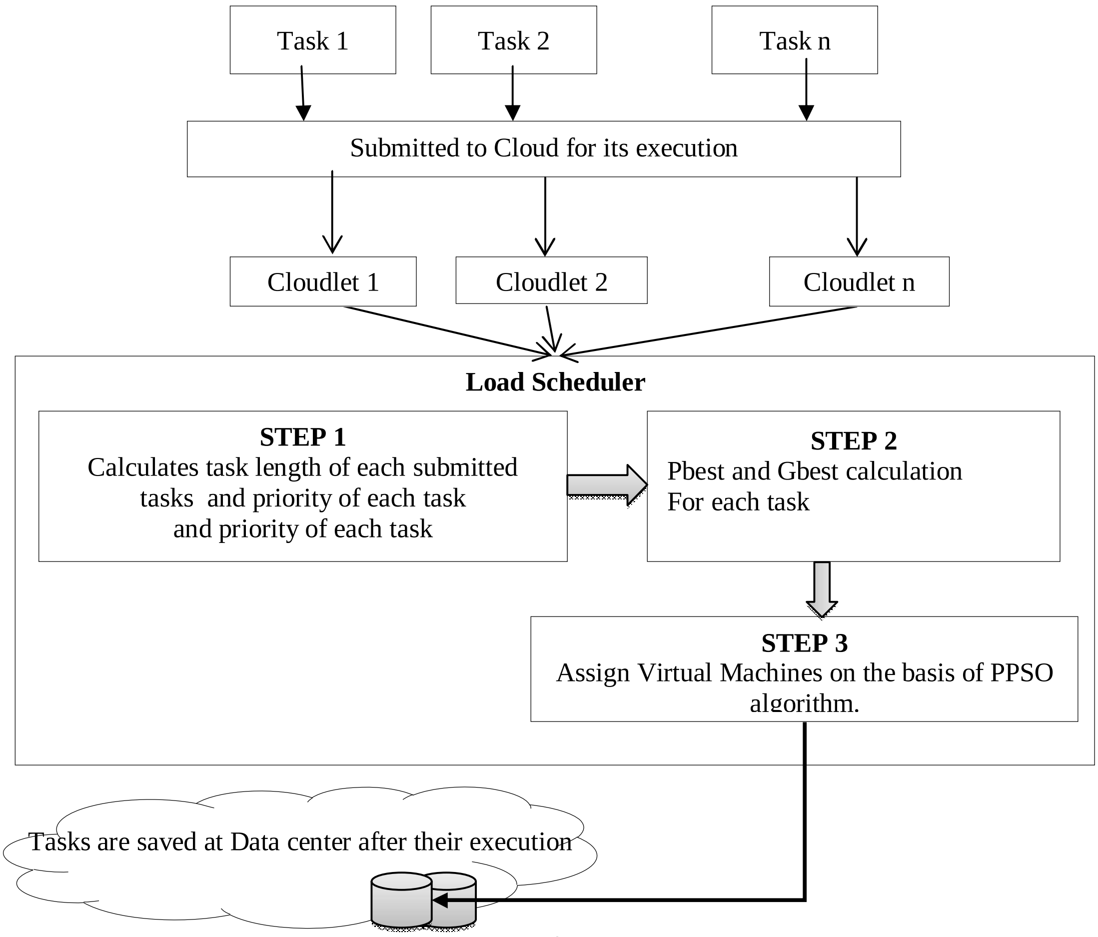

Figure 2: Flow chart of Proposed Methodology

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

The proposed algorithm is based on idea of priority PSO based assignment on which each task will be allotted a fitness function and according to assigned fitness function, tasks are executed so as to improve overall response time. After assigning fitness function to each task, each task will be scheduled on the basis of priority PSO algorithm. The proposed methodology flow chart is illustrated as below in figure 2.

Step 1: In this step length of all submitted tasks are calculated as well as priority and expected execution time is also calculated.

The priority of tasks is calculated according to normalized attributes of tasks which are as follows:

Priority (i)= a*User_type + b*Expected_priority + c*Task_length

Where a=0.4, b=0.3, c=0.3

User_type is high average low which is decided on the basis that the user is frequent or not.

Expected_priority shows the expected scheduled priority of tasks.

Task_length is the length of submitted jobs.

Step 2: Here in this step the proposed algorithm is used to find deciding factors attached with each task. These factors are used to order the submitted tasks or jobs in the ready queue (i.e. expected execution time and priority of each task).

Pbest= Priority * Priority_Weightage + Expected_Execution_Time * Expected_Execution_Time_Weightage

Step 3: In this step the submitted jobs or tasks are further scheduled by Particle Swarm Optimization Technique and is used to assign virtual machine which is discussed below:

PSO Algorithm

For each particle
Initialize particle
END
Do
For each particle
Calculate fitness value
If the fitness value is better than the best fitness value (pBest) in history
set current value as the new pBest
End
Choose the particle with the best fitness value of all the particles as the gBest
For each particle
Calculate particle velocity
Update particle position
End
While maximum iterations or minimum error criteria is not attained.

Calculation of fitness function based on task expected execution time and priority:

Here in this step the proposed algorithm extracts three major factors attached with each task. These factors are used to order the submitted tasks or jobs in the ready queue (i.e. arrival time, expected execution time and priority of each task).

pbest= Priority * Priority_Weightage + Expected_Execution_Time * Expected_Execution_Time_Weightage

Each Particle's fitness function is calculated using pbest as well as gbest which is best position among entire group of particles.

In each generation velocity and position of each particle is updated using following equation

v[] = v[] + c1 * rand() * (pbest[] - present[]) + c2 * rand() * (gbest[] - present[])

present[] = present[] + v[]

Where, v[] is the particle velocity
present[] is the current particle (solution)
pbest[] and gbest[] are defined as stated before.
rand () is a random number between (0,1).
c1, c2 are learning factors. usually c1 = c2 = 2.
then CTi =1
End For
Here CTi is the assigned credit token to the task after calculation.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

III. SIMULATION RESULTS

The proposed work is simulated using Cloudsim which is a toolkit used for modelling and simulation. The proposed work is simulated under the following conditions given in table I:

Table I: Simulation Parameters

Configuration Variable	**Value**
Data Center	1
Host	2
Virtual Machines	30, 50
Cloudlets	Simulation Files/NASAiPSC-1993-3.1-cln.swf(100-2000), CEA-Curie-2011-2.1-cln-b2.swf(100-2000)
PSO Number of Particles	10
PSO Iteration	100

A. Description of Dataset

THE NASA AMES IPSC/860 LOG

System:128-node iPSC/860 hypercube
Duration: October 1993 thru December 1993
Jobs:42050 total

This log contains three months worth of sanitized accounting records for the 128-node iPSC/860 located in the Numerical Aerodynamic Simulation (NAS) Systems Division at NASA Ames Research Center. The NAS facility supports industry, acadamia, and government labs all across the country [16]. The workload on the iPSC/860 is a mix of interactive and batch jobs (development and production) mainly consisting of computational aeroscience applications.

THE CEA CURIE LOG

System: CEA Curie
Duration: Feb 2011 to Oct 2012
Jobs:773,138

This log contains more than 20 months worth of data from the Curie supercomputer operated by CEA (a French government-funded technological research organization). The data comes from three partitions with a total of 11,808 Intel processors (93,312 cores) and an additional 288 Nvidia Graphic Processors.

B. Comparative Results

The metrics used here in for the evaluation in terms of QoS is average execution time of all submitted tasks. The simulation of the system is performed in Cloudlet with different number of Cloudlets varying from 100 to 2000 with 30 and 50 virtual machine using two different simulation files. The result of PPSO algorithm is compared with PSO algorithm under all scenarios which are discussed below. Table II and Figure 3 shows the comparative result analysis of proposed PPSO algorithm for NASAiPSC Simulation Files with 30 virtual machine.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Table II: Result Analysis of NASAiPSC Files under 30 VM

No. of VM	No. of Cloudlets	PSO Execution Time (in Sec)	PPSO Execution Time (in Sec)
30	100	144.43	122.78
	200	147.08	123.42
	300	175.69	147.7
	400	181.03	147.76
	500	201.4	156.05
	600	205.23	156.07
	700	224.93	207.58
	800	222.97	211.97
	900	226.67	212.21
	1000	234.19	212.46
	1500	328.66	271.59
	2000	386.95	314.1

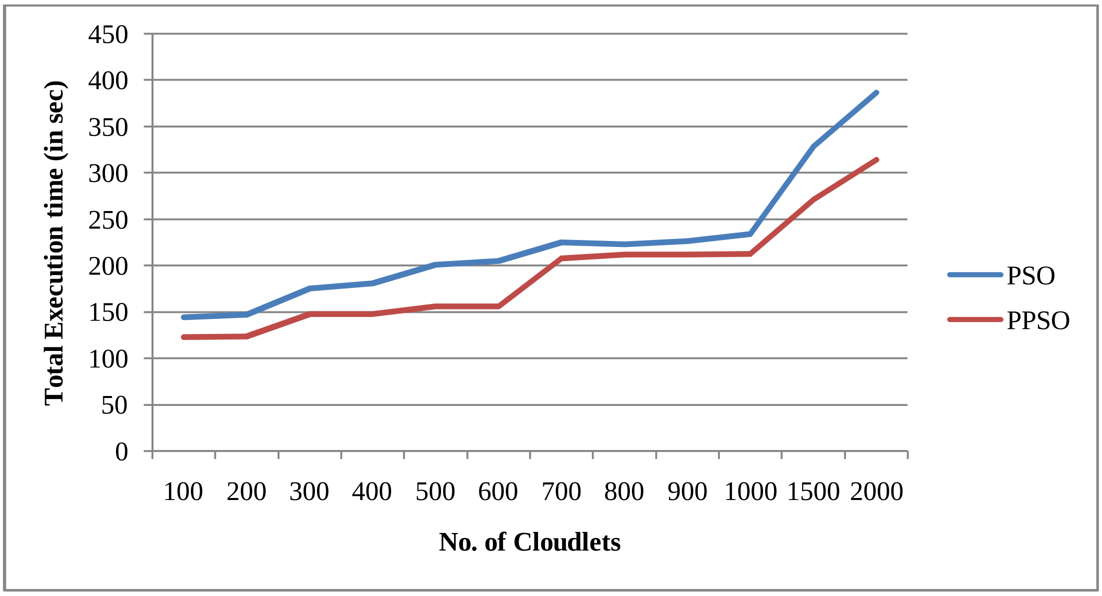

Figure 3: Result Analysis of NASAiPSC Files under 30 VM

DOI: 10.15680/IJIRCCE.2018.0603156

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Table III and Figure 4 shows the comparative result analysis of proposed PPSO algorithm for NASAiPSC Simulation Files with 50 virtual machine.

Table III: Result Analysis of NASAiPSC Files under 50 VM

No. of VM	No. of Cloudlets	PSO Execution Time (in Sec)	PPSO Execution Time (in Sec)
50	100	147	123
	200	153.31	123.63
	300	149.64	128.44
	400	189.45	128.45
	500	166.47	128.76
	600	202.41	128.82
	700	235.59	156.69
	800	193.28	156.68
	900	194.51	157.67
	1000	264.58	158.13
	1500	235.86	179.63
	2000	341.79	277.06

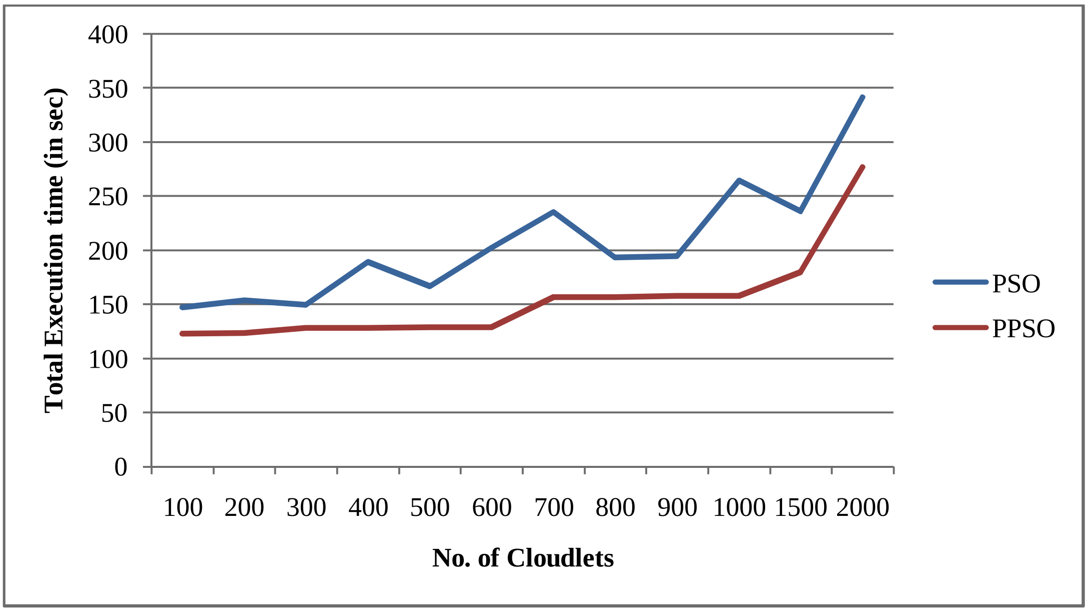

Figure 4: Result Analysis of NASAiPSC Files under 50 VM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Table IV and Figure 5 shows the comparative result analysis of proposed PPSO algorithm for CEA-Curie Simulation Files with 30 virtual machine.

Table IV: Result Analysis of CEA-Curie Files under 30 VM

No. of VM	No. of Cloudlets	PSO Execution Time (in Sec)	PPSO Execution Time (in Sec)
30	100	8909.66	5686.48
	200	7582.85	5686.63
	300	7965.03	5687.47
	400	7017.7	5687.65
	500	7221.33	5688.01
	600	35790.29	29008.37
	700	35790.82	29008.77
	800	72587.28	55923.76
	900	65769.73	55924.3
	1000	66793.39	55948.59
	1500	66027.97	55959.11
	2000	73032.03	56306.2

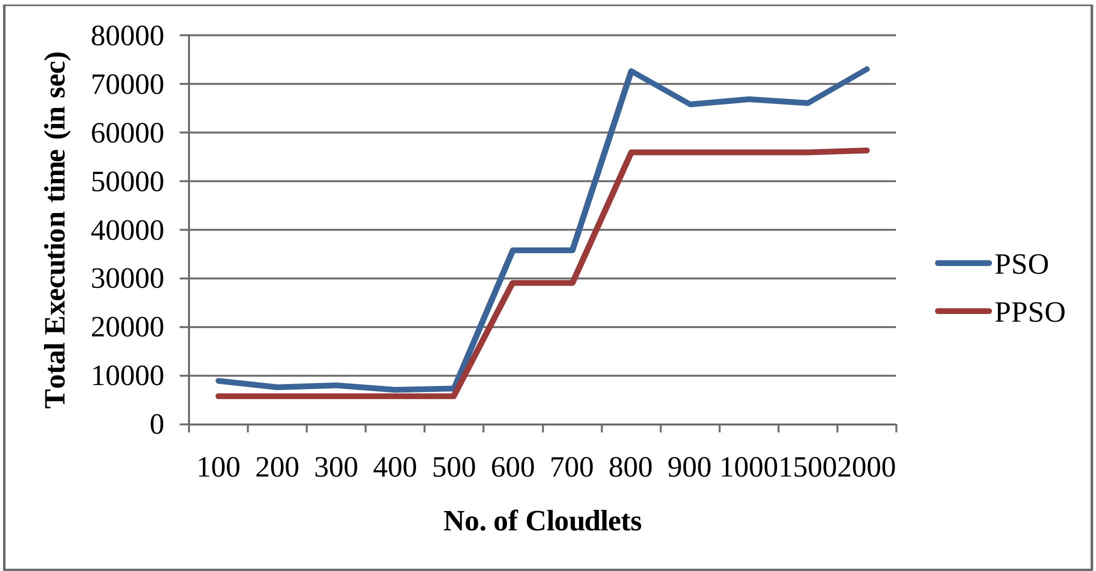

Figure 5: Result Analysis of CEA-Curie Files under 30 VM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

Table V and Figure 6 shows the comparative result analysis of proposed PPSO algorithm for CEA-Curie Simulation Files with 50 virtual machine.

Table V: Result Analysis of CEA-Curie Files under 50 VM

No. of VM	No. of Cloudlets	PSO Execution Time (in Sec)	PPSO Execution Time (in Sec)
50	100	7015.32	5686.13
	200	7959.21	5686.32
	300	7339.24	5686.71
	400	7195.09	5686.9
	500	10634.87	5687.08
	600	39236.87	29002.57
	700	35784.11	29002.84
	800	66785.55	54223.79
	900	66785.81	54224.01
	1000	66786.3	54224.4
	1500	66793.03	54229.85
	2000	341.79	277.06

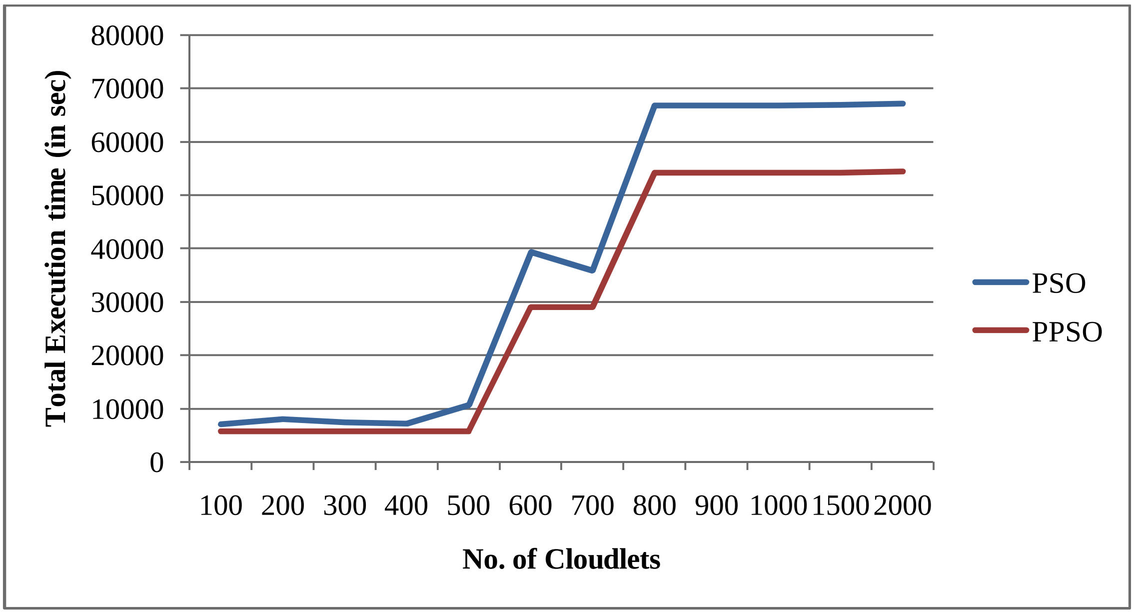

Figure 6: Result Analysis of CEA-Curie Files under 50 VM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 3, March 2018

IV. CONCLUSION

It is possible for IT service providers to provide computing resources in an pay-per-use way in Cloud Computing environments. At the same time, terminal users can also get satisfying services conveniently. But if we take only execution time into consideration when scheduling the cloud resources, it may occur serious load imbalance problem between Virtual Machines (VMs) in Cloud Computing environments. In addition to solve this problem, a new task scheduling model is proposed in this paper. In the model, we optimize the task execution time in view of both the task running time and the system resource utilization. Based on the model, a Particle Swarm Optimization (PSO) – based algorithm is proposed. In our algorithm, we improved the standard PSO, and introduce a simple mutation mechanism and a self-adapting inertia weight method by classifying the fitness values. The previous result analysis shows that swarm based load balancing algorithm will reduce the total execution time. So, proposed algorithm is enhancement of PSO based load balancing algorithm which is designed so as to minimize the make span time i.e. total execution time.

REFERENCES

1. M. Katyal and A. Mishra, "A comparative study of load balancing algorithms in cloud computing environment." in International Journal of Distributed and Cloud Computing, Volume 1 Issue 2 December 2013.
2. Teena Mathew, K. Chandra Sekaran, John Jose, "Study and Analysis of Various Task Scheduling Algorithms in the Cloud Computing Environment", IEEE, 2014.
3. Gaurang Patel, Rutvik Mehta, Upendra Bhoi, "Enhanced Load Balanced Min-Min algorithm for static Meta Task Scheduling in Cloud Computing", ICRTC, Elsevier, 2015.
4. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar, Mondal, Santanu Dam, " A Genetic Algorithm based Load Balancing Strategy for Cloud Computing", CIMTA, Elsevier, 2013.
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, 2013.
6. Monir Abdullaha, Mohamed Othmanb, Cost Based Multi QoS Job Scheduling using Divisible Load Theory in Cloud Computing, International Conference on Computational Science, ICCS 2013.
7. Upendra Bhoi, Purvi N. Ramanuj, "Enhanced Max-min Task Scheduling Algorithm in Cloud Computing", International Journal of Application or Innovation in Engineering & Management, Volume 2, Issue 4, April 2013, pages 259-264.
8. Soumi Ghosh, Chandan Banerjee, "Priority Based Modified Throttled Algorithm in Cloud Computing", IEEE, 2016.
9. Mr. Shubham Sidana, Ms. Neha Tiwari, Mr. Anurag Gupta, Mr. Inall Singh Kushwaha, "NBST Algorithm: A load balancing algorithm in cloud computing", IEEE, 2016, pp. 1178-1181.
10. Pandaba Pradhan, Prafulla Ku. Behera, B.N.B. Ray, "Modified Round Robin Algorithm for Resource Allocation in Cloud Computing", International Conference on Computational Modeling and Security, ELSEVIER, 2016, pp. 878-890.
11. Kolodziej, J., Xhafa, F.: Modern approaches to modeling user requirements on resource and task allocation in hierarchical computational grids. Int. J. Appl. Math. Comput. Sci. 21(2), 243–257, 2011.
12. Zhao, C., Zhang, S., Liu, Q., Xie, J., Hu, J.: Independent tasks scheduling based on genetic algorithm in cloud computing. In: 5th International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1–4, 2009.
13. Cheng, B., "Hierarchical cloud service workflow scheduling optimization schema using heuristic generic algorithm", Przeglad Elektrotechniczny, 88(3):92-5, 2012.
14. Gomathi B,. Karthikeyan, K, "Task Scheduling algorithm Based on hybrid particle swarm optimization in cloud computing environment", Journal of Theoretical and Applied Information Technology, 33-38, 2013.
15. Hussain A Makasarwala and Prasun Hazari, "Using Genetic Algorithm for Load Balancing in Cloud Computing", IEEE, 2016.
16. Jigna Acharya, Manisha Mehta, Baljit Saini, "Particle Swarm Optimization Based Load Balancing in Cloud Computing", IEEE, 2016.
17. Hicham Ben Alla, Said Ben Alla, Abdellah Ezzati, "A novel architecture for task scheduling based on Dynamic Queues and Particle Swarm Optimization in cloud computing", IEEE, 2016.