ISSN 2321-807X
Volume 12 Number15
Journal of Advances in chemistry

Responsive Multi-objective Load Balancing Transformation Using Particle Swarm Optimization in Cloud Environment

VG.Ravindhren[1], Dr. S. Ravimaran[2]
[1]Lecturer (SG), Seshasayee Institute of Technology, Tiruchirappalli. Tamil Nadu
Affilated to Directorate of Technical Education, Chennai
Email: [1]ravindhren.vg@gmail.com
[2]Software System Group Lab, Department of Computer Science and Engineering
Affiliated to Anna University, Chennai, M.A.M. College of Engineering,
Tiruchirappalli, Tamil Nadu.India
Email: [2]ssg_ravimaran@mamce.org

ABSTRACT

Cloud computing is an emerging computing paradigm with a large collection of heterogeneous autonomous systems with flexible computational architecture which provides the customers with computing resources as a service over a network on their demand. A multi-objective nature is inherent in cloud resource scheduling, as the objectives of cloud providers, cloud users, and other stakeholders can be independent. Resource allocation among multiple clients has to be ensured as per service level agreements. Several techniques have been invented and tested by research community for generation of optimal schedules in cloud computing. To accomplish these goals and achieve high performance, it is important to design and develop a Responsive multi-objective load balancing Transformation algorithm (RMOLBT) based on abstraction in multi cloud environment. It is most challenging to schedule the tasks along with satisfying the user's Quality of Service requirements. This paper proposes a wide variety of task scheduling and resource utilization using Particle swarm Optimization (PSO) in cloud environment. The result obtained by RMOLBT was simulated by an open source cloudsim configured with test case specification. Finally, the results demonstrate the suitability of the proposed scheme that will increase throughput, reduce waiting time, reduction in missed process considerably and balances load among the physical machines in a Data centre in multi cloud environment.

Keywords

Cloud Computing, Resource Management, responsive scheduling, dynamic scheduling, Cloud optimization.

Academic Discipline And Sub-Disciplines

Information Communication and Engineering

SUBJECT CLASSIFICATION

Cloud Computing

TYPE (METHOD/APPROACH)

System Model and Experimental Analysis Approach

1. INTRODUCTION

Cloud computing is a new paradigm for enterprises that can effectively facilitate the execution of tasks. Task scheduling is an important issue which is greatly influencing the performance of cloud computing environment. The cloud service provider and clients have different objectives and requirements. In dynamic environment resource availability and load on resources is changed time to time. So, scheduling resources in Clouds is a complicated problem. Scheduling problems belong to a broad class of combinational optimization problems aiming at finding an optimal matching of tasks to different sets of resources.

Task scheduling algorithm is a method by which tasks are matched, or allocated to data center resources. Due to conflicting scheduling objectives generally no absolutely perfect scheduling algorithm exists. A good scheduler implements a suitable compromise, or applies combination of scheduling algorithms according to different applications. A problem can be solved in seconds, hours or even years depending on the algorithm applied. The efficiency of an algorithm is evaluated by the amount of time necessary to execute it. The execution time of an algorithm is stated as a time complexity function relating the input. There are several kinds of time complexity algorithms that appear in the literature.[1][2].

Load balancing in cloud computing provides an efficient solution to various issues residing in cloud computing environment setup and usage. Load balancing must take into account two major tasks, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment. Efficient provisioning of resources and scheduling of resources as well as tasks will ensure: (a). Resources are easily available on demand. (b). Resources are efficiently utilized under condition of high/low load. (c). Energy is saved in case of low load (i.e. when usage of cloud resources is below certain threshold). (d). Cost of using resources is reduced[3].

Load balancing can be achieved by first mapping the tasks to VM's and further all the VMs to host resources by the means of Task-Based System Load Balancing method. This algorithm achieves the system load balancing by transferring only extra tasks from an overloaded VM rather than migrating the entire overloaded VM. The loads are formulated as:

VM_load = Bandwidth+ RAM+ MIPS

Multi Objective is used to achieve efficient resource utilization, further optimizing the resources in terms of accuracy and efficiency. There are number of data centers in a cloud to serve customers demands. Data centers further consists of number of servers and in turn each server runs number of VMs. [4]. The execution capacity of different VMs varies to execute tasks with different QoS parameter. Cloud broker is an important entity in cloud computing as it sends request to the cloud service provider for the QoS of requested task and for list of VMs. The task's priority is allocated as per the QoS value of task in such a way that the lower QoS valued task is given high priority and vice versa. To assign QoS to VMs, Millions of instructions per second (MIPS) of VM are used in such a way that the VM with high MIPS is declared high QoS VM and similar is with low MIPS VM. The list of VMs possessed by cloud broker is updated after fixed time interval. Based on MIPS the list of VMs is sorted in descending order starting from high QoS VM to low QoS VM and non- dominated sorting is performed on the list to generate non-dominated task's set. These tasks are bounded to VMs sequentially and the process of allocation is repeated for all tasks. Multiple Objective resource utilization is a mathematical optimization with more than one objective functions to be achieved and mathematically can be expressed as:

$$\textit{Find } X = \{x_1, x_2, \ldots\ldots\ldots\ldots x_n\} \text{ Which } \mathit{Min}[f1(x), f2(x), \ldots\ldots\ldots fk(x)]$$

where k>=2 represents the number of objectives and X is the set of feasible solutions. A multi-objective optimization does not normally provide a sufficient solution that minimizes all objective functions at the same time. In mathematical form, a feasible solution x_1 dominates another solution x_2 if:

1. fi(x1) ≤ fi(x2) for all objectives i £ {1,2,……,k } and
2. fj(x1) <fj(x2) for at least one objective j £ {1,2,……,k }

Here solution x1 is a multi objective optimal solution as there does not exist better solution that dominates it.

Particle Swarm Optimization (PSO) has been found to be robust and is successfully applied in solving nonlinear, non differentiable multi-modal problems quickly. It is still in its infancy. Many research works have mentioned application of PSO in task scheduling. PSO is most successful meta heuristic algorithm for generations of optimal scheduling solutions. PSO scans over solution space during each iteration and accumulates global best and local best solutions. This section presents review of recent proposals which considered PSO in the field of task scheduling in cloud environment[5]. We proposed a mathematical model using a Load Balancing transformation Particle Swarm Optimization (LBTPSO) and considered reliability and availability as the objective parameters of proposals. LBTPSO used an algorithm to generate schedule and allocation for cloud computing environment. This algorithm considers available resources for generation of schedule and allocation patterns. Basic PSO suffers from free VMs, allocation of more than one task to same VM, allocation of same tasks to multiple VMs and premature convergence.

LBTPSO takes into account execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and achieved reliability in task scheduling. The idea of LBTPSO is to reschedule failure tasks to available VM. The performance of LBTPSO is efficient and the cost is the lowest as compared with other representative counterparts. The rest of the paper is organized as, section 2 contains a literature survey as related work about scheduling in cloud computing, section 3 describes about the system model of responsive multi objective Load balanced transformation using PSO Algorithm. Section 4 discusses details about experimental setup and experimental results of the proposed model and the paper concludes with conclusion in Section 5.

2. RELATED WORK

In [6], authors proposed a model for resource-task mapping which could reduce execution cost and also designed a PSO based heuristic to allocate tasks to resources. Both computation cost and data transmission cost are calculated by using the workflow application. Authors compared results of heuristic against "Best Resource Selection" (BRS) heuristic and found that PSO based task scheduling could result into three times cost savings. In [7], authors compared three popular heuristic approaches namely PSO, GAs and MPSO for efficient task scheduling in cloud environment. MPSO algorithm improved makespan characteristics when compared with PSO and GA. In [8], authors proposed a hybrid of particle swarm optimization and simulated annealing. CloudSim toolkit was used to implement and analyze the proposal. Population based metaheuristic (PBM) algorithm maximizes resource utilization and minimizes makespan and demonstrating improvements upwards of 53%. Hybrid meta-heuristic method proved to enhance the performance in CloudSim toolkit. In terms of resources utilization and makespan Hybrid PSO implementation in [9] tried to balance the load across the system and minimize the makespan. In [10], authors presented multi-objective PSO based optimization algorithm for dynamic environment of clouds and optimize energy and processing time. Proposed algorithm provides an optimal balance results for multiple objectives. Experimental results illustrated that proposed methods (TSPSO) out-performed BRS and RSA. In CA-PSO authors proposed Modified PSO by augmenting PSO with Cost-Aware Heuristic. CAPSO generates tasks schedules using a cost aware fitness function to quantify the cost of resource usage along with fitness function for time cost to minimize processing time. Authors preferred to optimize usage cost of resources if possible. In [11], a task-level scheduling algorithm Chaotic Particle Swarm Optimization (CPSO) is based on chaotic sequence and inertia weight factor. Chaotic sequence with high predictability improves the variety of solutions and its reliability assures a good global convergence. It can optimize the cost of whole scheduling and overcome the premature convergence of PSO algorithm to satisfy the market-oriented characteristic of cloud workflow.

3. PROPOSED WORK

Load balancing in cloud computing provides an efficient solution to various issues residing in cloud computing environment set-up and usage. Load balancing must take into account two major tasks, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment.

Efficient provisioning of resources and scheduling of resources as well as tasks will ensure:

a. Resources are easily available on demand.

b. Resources are efficiently utilized under condition of high/low load.

c. Energy is saved in case of low load (i.e. when usage of cloud resources is below certain threshold).

d. Cost of using resources is reduced.

We introduce the new method of Resposive multi-objective load balanced transformation using particle swarm optimization.It places user application(s) inside a cloud software as Services container, where Virtual Machines abstract operating systems from hardware, SaaS containers abstract application(s) from platform. The Figure 1 shows the system architecture.

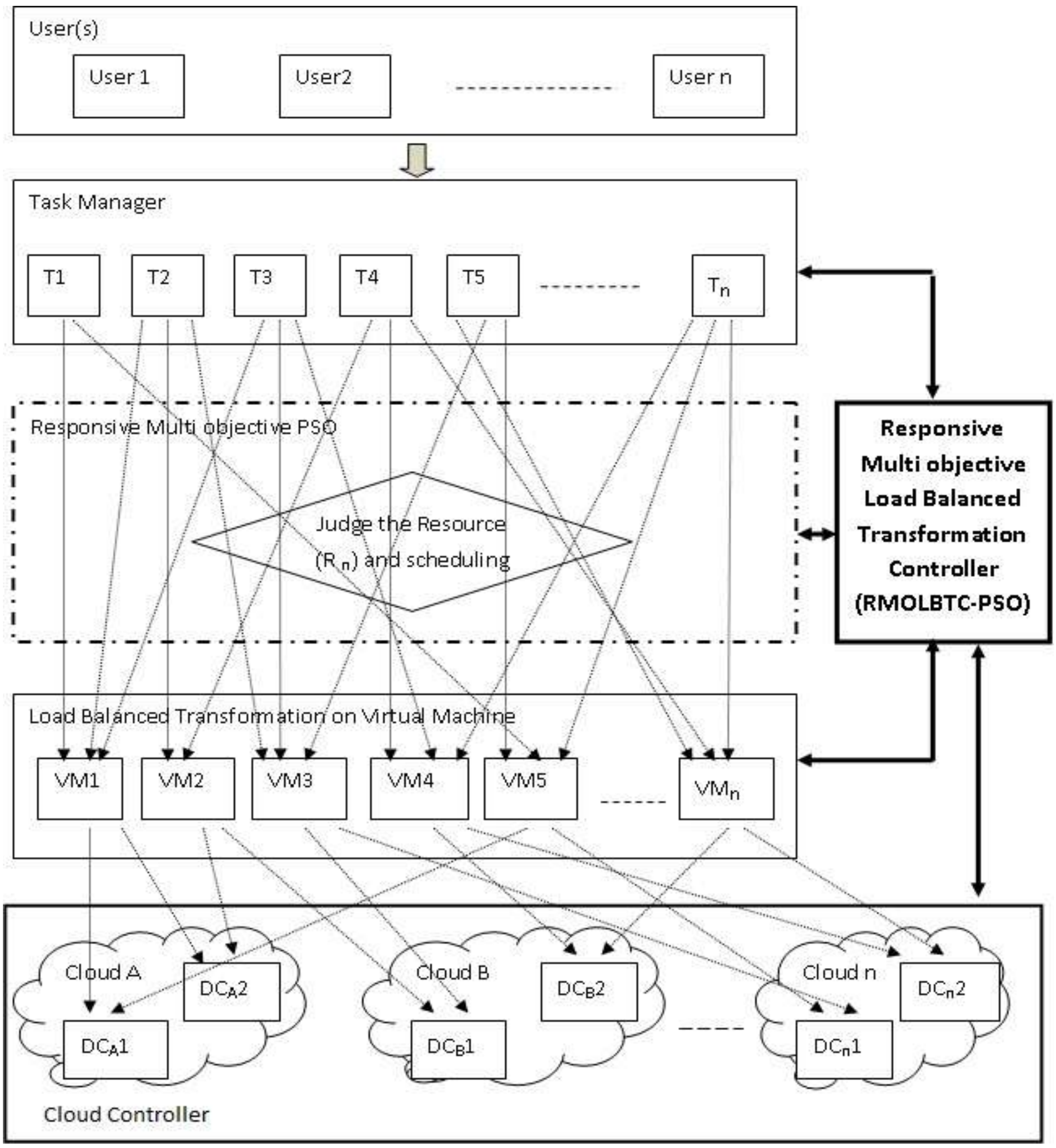

Fig 1: System Model for Responsive Multi objective Load Balance Transformation using PSO

The system architecture shows the model for which the propsed algorithms is implemented. Here the jobs are submitted by the user for computing process in cloud. As the jobs are submitted by user they arrive to the cloud and are queued in as user tasks. Our RMOLBT-PSO check the avalibility of resource and estimates the job size and checks for capacity of the VMs. Once the job size and the available resources size match, the job is scheduled immediately, VMs are allocated and the identified resources to the user tasks in queue. The impact of the proposed algorithm has an improvement in response time by the jobs are equally spread to complete the process by responsive multi objective load balanced abstraction and no virtual machine are ideal. The advantage is that the processing cost of cloud and the data transcation cost are reduced.

The proposed model carried out the following steps:

Step1: Users : The user submit and store the tasks and are queued. The user tasks are classified and recorded in RMOLBT controller. This is represented as n users as $t_1,t_2, \ldots t_n$ and there also the independent tasks represented, as $i_1,i_2,\ldots i_n$.

Step 2: Virtual Machine: The VMs responsible for collecting and recording information about the idle and engaged VMs which are represented as $VM_1,VM_2,..VM_n$

Step 3: Data Centers: The data centers are responsible for selecting available host in a data center, which meets the storage, memory and MIPS of VM deployment and are represented as DCx_1, $DCx_2 \ldots DCx_p$, where x means the data centers of respective cloud host and p means the available DC.

Step 4: Cloud Controller: It is responsible for controlling the multi-objective allocation of data centers of multi cloud environment . The federated cloud is also manipulated and controlled for users task.

Step 5: RMOLBT-PSO: It is responsible for the collection of user requirement, the user task submitted by the task manager which schedule the task by judging the available resource to assign by responsive multi-objective scheduling to a VM to complete the task by Load balanced Transformation, which perform as data abstraction of user tasks based on which the VM is allocated based on priority. The priority is assigned by storage, memory and million instriuction per seconds (MIPS) of VM , and stored as the status of the VM. After the priority are assigned the Particle swarm optimization algorithm is used to allot the Data centers in multi cloud environment with difference services which perform the user task abstraction of load balancing and tasks are transformed to the VM for computing process on muti-cloud data center. The proposed model can be formulated as follows.

$$\textit{User task} = t_1,t_2,\ldots t_n$$

a set of 'n' task to be scheduled with 'n' cloud data center (DC) in turn to be executed by 'n' virtual machines (VM). Consequently, each cloud data center can carry out a subset of user task and represented as follows.

$$VM_i = (\textit{user task}\{ Ti_1,Ti_2 \ldots Ti_n\})$$

The total processing time of all task are assigned to VM_i which make span with start time and execution time which are represented as follows.

$$Makespan(VM_j)=MAX(T_i).\textit{start time} + Ti.\textit{excute time}$$

Thus the task scheduling in cloud environment could be defined as searching of a set. It is represented as follows.

$$nTask=\{ VM_1,VM_2,\ldots VM_n\} \text{ and}$$

$$VM_{ij}=\{Ti_{1j},Ti_{2j},\ldots Ti_{nj} \} ; \text{ with } 0< =n \text{ which reduces makespan}(VM_i)$$

In order to evaluate the quality of user request process VM task is based on fitness used by particle swarm optimization algorithm. The fitness function is defined as

$$Fitness(VM_i)=\sum(fitness(T_i,VM_j)) ;\text{where } 1<=i<=n \text{ and}$$

$$Fitness(Ti,VMi)=Ti.timetoexecute ;\text{where Ti.timetoexecute is the execution time of user task of job 'i' need to run in } VM_i.$$

In each multi cloud contains 'N' individual solution where each one is represented by a set of datacenter. Each datacenter carried out a set of job tasks by PSO as follows.

$$\sum(\text{datacenter})\ VM(\text{task})=\{VM_1T,VM_2T,..VM_nT\}$$

3.1. Load Balance Abstraction Algorithms

The proposed load balanced abstraction algorithm eliminates the drawbacks of existing load balancing algorithm in cloud computing. In the proposed architecture when a new process arrives in the system it is sequenced at a small resource. The Load balanced abstraction algorithm selects the server to which to send client requests. The abstracted connections method selects the server with the abstracted active connections to ensure that the load of the active requests is balanced on the servers. The Load Balanced transformation gathers the load information from every node to evaluate the cloud resource allocation status. The first step is to find the weight of each node. The weight of node is related to various static and dynamic parameters. The Load Balanced transformation will choose the node that is either idle or normal in status based on Abstraction. Then, it will assign the request to that node for further processing. The weight results are input to the load status table created by the Load Balanced Abstraction. Each Balancer has a weight table, which is used to calculate the cloud resource allocation status based on following algorithm 1.

Algorithm 1: Request Allocation to Node

set counter=0;

Select node sequentially

If counter< max load capacity of node

Assign request to node && counter++;

else select next node

Check for node

If counter< max capacity of node

Assign request to node && counter++;

If counter == max capacity of node

Request assignment not possible;

wait for some nodes to become free;

end if

end

else

end if

The algrothim 2 describes priority concept leads to better utilization of the resources. Priority of a request is assigned depending upon its capacity and the load factor.

Algorithm 2: Priority

Input: N number of client Request

Output: Balance the Request

Algorithm check priority schedule

1. Get the available VM List //find the appropriate VM List from job scheduler

2. If priority1 is not assign to request

3. Priority1=max available VM

4. Else if priority1 is set

5. Turn ON priority1

6. Used VM List

7. If VM is not used VM List then

Add VM to used VM List

8. Deploy request on new VM

9. If priority N is not assign goto step1

10. Assign VM to ClientRequests

11. If assigned the requests then return Successful

12. End for

13. Return available VMList

The Priority based scheduling Algorithm is intended to be used by organizations need to implement small to medium sized local clouds. This algorithm should scale to larger sized clouds because one of the main contributions of the cluster controller is load balancing compute nodes. Each process is assigned as a priority. Process with highest priority is to be executed first and so on. Processes with same priority are executed on first come first serve basis. Priority can be decided based on memory, time or any other resource requirement shown in Table.1.

Table 1: Priority Value

Request	Arrival Time(ms)	Execute Time(ms)	Priority (P)	Service Time(ms)
R0	0	8	3	0
R1	1	4	2	4
R2	2	10	1	10
R3	3	7	4	7
R4	4	5	5	5

Algorithm 3: Load Balanced Abstraction Algorithm

```
abstraction_algorithm()
{
Inputs: Job-> x1,x2,…………xn
        Initialize all the available VM -> zero
                Old VM -> oVM1,oVM2,………………...,oVMn=0;
                New VM ->  nVM1,nVM2,………………….,nVMN=0;
        If (oVM1==1)
                        { nVM1+=1; }
                Else if
                        { nVM1+=1; }
                                Selected VM=abstract VM (nVM1,nVM2,………………… ,nVMN)
                                        If(Selected VM is free)
                                                { Selected VM=req(x1,x2,…………xn) }
                                        Else
                                                { abstract VM =exclude(selected VM) }
                                        goto
                                                selected VM
}
```

3.2. Multi objective Particle Swarm Optimization algorithm

The problem to map a user task , available resource and data center in cloud based on the multi-objective optimization method, so that the particle swarm optimization algorithm is a class to find not one “global best” solution to provide a benefits to both provider and user by cost reduce, makespan, power and energy consumption. In contrast to other, multi objective particle swarm optimization algorithm with load balanced transformation inherits the basic idea of PSO algorithm and Load balanced algorithm to decrease execution time of task with multi- cloud data center.

The pseudo code of the PSO procedure is as follows.

```
For each VMs
{
  Initialize VM
}
Do until maximum iterations or minimum error criteria
{
  For each VMs
  {
```

Calculate User Task fitness value

If the fitness value is better than pBest

{

Set pBest = current fitness value

}

If pBest is better than gBest

{

Set gBest = pBest

}

}

For each VMs

{

Calculate particle priority

Use gBest and priority to update particle User Task

}

Adopting the load balancing, scheduling and resource model into the responsive multi-objective PSO model are carried out by assuming all the tasks are executed based on our proposed algorithms to minimize the total execution time of user task as well as to achieve a well load balanced transformation across all VMs in multi-cloud.

4. EXPERIMENTAL RESULTS

We create a class by inherit of LBA and PSO algorithms to find the RMOLBT using PSO for task scheduling and optimized resource allocation in cloud environment by simulation using CloudSim. For measuring the efficiency and effectiveness load balanced abstraction, the simulation environments are required. CloudSim is the most efficient tool that can be used for modeling of Cloud. During the lifecycle of a Cloud, CloudSim allows VMs to be managed by hosts which in turn are managed by datacenters.The resource provisioning algorithm can be categorized into different classes based upon the environment, purpose and technique of proposed solution. The cloud setup in CloudSim Simulator is detailed in Table 3.

Table 3: Cloud Setup Configuration Details.

Sl.No	Entity	Quantity	Purpose
1	Data center	3	Data center having the physical hosts in the test environment
2	Number of hosts in DC	500 hosts (200 Nos-4-CoreHyperThreadedCorei3270, 200 Nos-4-Core HyperThreadedOpteron2218 and 100 Nos 8-Core HyperThreaded Corei3)	Number of physical hosts used in the experiment
3	Number of process elements	8/8/16	Number of executing elements in each of the hosts. The host has 8 or 16 processing elements
4	PE processing capacity	174/247/355 MIPS	Each host has any one of the processing capacities
5	Host ram capacity	16/32 GB	Each host has any of these RAM memories
6	Number of VM	10 to 100 with an increment of 10	Number of virtual machines used in the experiment
7	Number of PE to VM	3	Processing element allotted in each VM
8	VM's PE processing capacity	150/300/90/120/93/112/105/225	Virtual machine's processing capacity
9	VM RAM capacity	1920 MB	The RAM's memory capacity of the VM
10	VM manager	Xen	The operating system runs on the physical machine to manage the VMs. It provides the

			virtualization
11	Number of PE in Tasks	1	The job/task's maximum usable processing element
12	Task length/instructions	500000 to 200000000	Tasks length in million instructions. The heterogeneous job length test having the variations from the mentioned minimum to maximum

The Load balanced Abstraction algorithm proved that the task length delivers a faster completion time with weighted and priority in the heterogeneous resources (VMs) and homogenous tasks. The Figure 2 illustrates it.

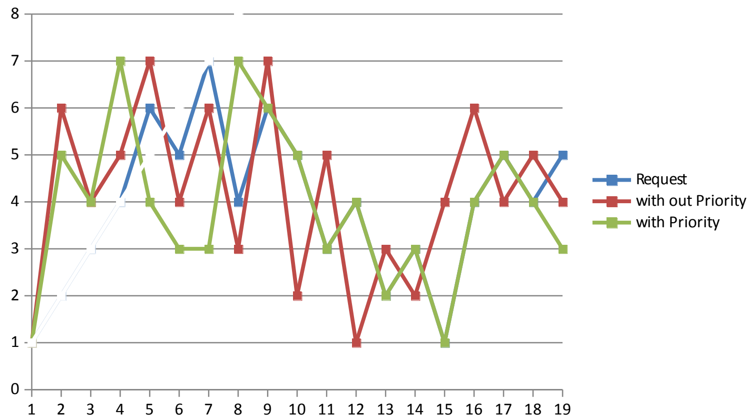

Fig 2: Execution of Task completion time

The task migrations are very minimal in the RMOLBT-PSO model due to extensive in existing scheduler algorithm as identified the most appropriate VM to each of the tasks which is illustrated in Figures 3.

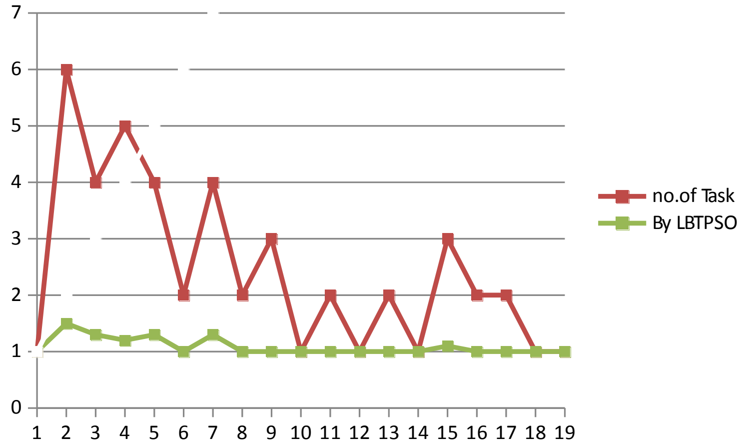

Fig 3: Number of task migrations using LBTPSO

The LBA algorithm makes a better resource utilization of the available resources in cloud is shown in the Figure 4.

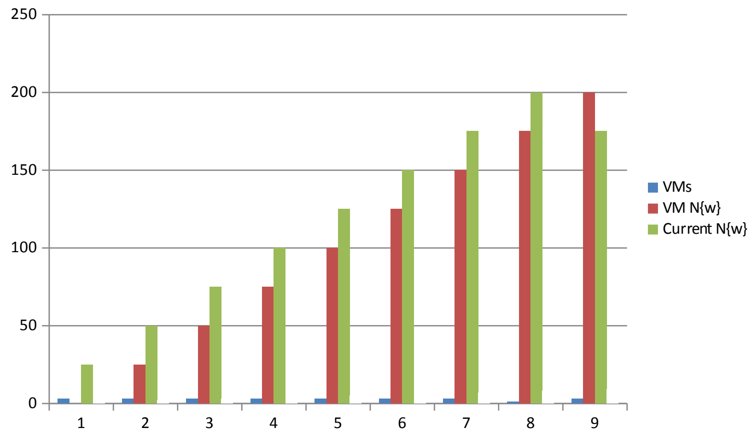

Fig 4: Resource Utilizations

5. CONCLUSION

We have presented a Responsive Multi-Objective Load Balancing Transformation Model using Particle Swarm Optimization in Cloud environment based on a multi-objective approach. In this model we take as objective functions the user task, connection cost, and energy consumption; with the aim of performing an efficient use of the capacity resources in the available Virtual Machines. Based on this model it was presented a responsive multi objective load balance Algorithm (RMOLBT). The results obtained by our proposal were satisfactory and provided a user task scheduling and resource allocation with less time execution and cost in their cloud. With this process they could get an efficient use of their cloud resources, reduce the connection costs, and extend the power efficiency. Due to we proposes a Particle swarm optimization algorithm, the model is opens up to incorporate additional parameters as objective functions; these parameters could be obtained from both the available access user task and the resource allocation. As future work we propose to continue this research, we want to introduce the concept of fairness in the load balance optimization and also include the concept of quality of experience (QoE) in the objective functions.

6. REFERENCE

1. J. Dean and S. Ghemawat., 2014, MapReduce: simplified data processing on large clusters, Sixth Symposium on Operating System Design and Implementation (OSDI’14). , p. 1-13.
2. Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic . 2009,Cloud Computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. Journal Future Generation Computer Systems. 25 : 6.
3. Liu, K. , 2009, Scheduling Algorithms for Instance Intensive Cloud Workflows. Ph.D. Thesis, Swinburne University of Technology, Australia.
4. A. Y. Zomaya, and Y. 2011, The Observations on using genetic algorithms for dynamic load-balancing. IEEE Transaction on Parallel and Distributed Systems; 12: 899-911
5. Yu, J., and Buyya, R. 2008, Workflow Scheduling Algorithms for Grid Computing,; In Xhafa F, Abraham A (eds), Metaheuristics for scheduling in distributed computing environments, Springer, Berlin.
6. P. Angin and B. Bhargava, 2013, “An Agent-based Optimization Framework for Mobile-Cloud Computing”,Journal of Wireless Mobile Networks,Ubiquitous Computing, and Dependable Applications, vol. 4,pp. 1-17.
7. C.Mateos, E.Pacini and C. G.Garino,, 2013 “An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments”, Advances in Engineering Software, vol.56,pp. 38–50.
8. P.Samal and P.Mishra, 2013, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing",International Journal of Computer Science and Information Technologies, Vol. 4, no.3, pp. 416-419.
9. B.Mondala, K.Dasgupta and P.Duttab, 2012,” Load Balancing in Cloud Computing using Stochastic Hill Climbing-A Soft Computing Approach”, in 2nd International Conference on Computer, Communication, Control and Information Technology , on February 25 – 26, vol.4, PP. 783 – 789,.
10. S.Pandey, L.Wu, S.M.Guru2 and R.Buyya, 2010,” A Particle Swarm Optimization-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments”,in IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 400 - 407.
11. G.Liu, J.Li and J.Xu, 2012, “An Improved Min-Min Algorithm in Cloud Computing”, Proceedings of the International Conference of Modern Computer Science and Applications, vol .191,pp.47-52.
12. W.Wang, G.Zeng, D.Tang and J.Yao, 2012,”Cloud-DLS: Dynamic trusted scheduling for Cloud computing”, ExpertSystems with Applications , vol.39,no.3, PP. 2321–2329,.
13. B. Xu, C.Zhao, E.Hu and B.Hu, 2011 ,”Job scheduling algorithm based on Berger model in cloud environment”, Advances in Engineering Software, vol. 42,no.7, PP. 419–425.
14. S.Ambike, D.Bhansali, J.Kshirsagar and J.Bansiwal 2012, “ An Optimistic Differentiated Job Scheduling System for Cloud Computing” International Journal of Engineering Research and Applications (IJERA) , Vol. 2, no. 2, pp.1212-1214.
15. H.Zhong, K.Tao and X.Zhang , 2010 ,” An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems “, in Fifth Annual China Grid Conference, ,on 16-18 July , pp. 124 - 129.

16. A. S. Ajeena Beegom and M. S. Rajasree, 2014 ," A Particle Swarm Optimization Based Pareto Optimal Task Scheduling in Cloud Computing ",in Proceedings of 5th International Conference, ICSI 2014, October 17-20, pp 79-86.
17. Eberhart R , Kennedy J, "A New Optimizer Using Particle Swarm Theory", 6th International Symposium. Micro Machine and Human Science, Nagoyap. 39-43.
18. M.Choudhary and S.K.Peddoju, 2012 "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment" ,International Journal of Engineering Research and Applications (IJERA), , Vol. 2, no.3, pp.2564-2568.
19. S.Ghanbaria and M.Othmana, 2004." A Priority based Job Scheduling Algorithm in Cloud Computing",in International K. Amin, G. von Laszewski, M. Hategan, N. J. Zaluzec, S. Hampton, and A. Rossi. Gridant: A client-controllable grid work.ow system. In *HICSS '04: Proceedings of the Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS'04) - Track 7*.
20. M. Armbrust, A. Fox, R. Grifth, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia. 2009.,Above the clouds: A berkeley view of cloud computing. Technical report, University of California at Berkeley.
21. R. Buyya, S. Pandey, and C. Vecchiola. 2009.Cloudbus toolkit for market-oriented cloud computing. In *CloudCom '09:Proceedings of the 1st International Conference on Cloud Computing*, volume 5931 of *LNCS*, pages 24–44. Springer, Germany,
22. J. Cao, S. A. Jarvis, S. Saini, and G. R. Nudd. Gridflow: Workflow management for grid computing. In *CCGRID '03: Proceedings of the 3rd International Symposium on Cluster Computing and the Grid*, pages 198–205, Washington, DC, USA.
23. E. Deelman, G. Singh, M.-H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, Laity, J. C. Jacob, and D. S. Katz. Pegasus: A framework for mapping complex scientific workflows onto distributed systems. *Sci. Program.*, 13(3):219–237.
24. N. Furmento, W. Lee, A. Mayer, S. Newhouse, and J. Darlington. Iceni: an open grid service architecture implemented with jini. In *Supercomputing '02: Proceedings of the 2002 ACM/IEEE conference on Supercomputing*.
25. Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers. 2007,Examining the challenges of scientific workflows. *Computer*, 40(12).
26. J. Kennedy and R. Eberhart. Particle swarm optimization. In *IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948.
27. J. Louchet, M. Guyon, M. J. Lesot, and A. Boumaza. , 2002,Dynamic flies: a new pattern recognition tool applied to stereo sequence processing. *Pattern Recognition Letters*, 23(1-3):335–345.
28. W. Z. Lu, H.-Y. Fan, A. Y. T. Leung, and J. C. K. Wong. Analysis of pollutant levels in central hong kong applyingneural network method with particle swarm optimization.*Environmental Monitoring and Assessment*, 79(3):217–230.
29. B. Lud¨ascher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, and Y. Zhao. , 2006Scientific workflow management and the kepler system: Research articles. *Concurrency and Computation: Practice & Experience*, 18(10):1039–1065.

Author' biography

V.G. Ravindhren received B.E. in Computer Science and Engineering from Mookambika College of Engineering, Pudukottai, India in 1990 and M.E. Computer Science and Engineering from then Regional Engineering College(now NIT) , Tiruchirapalli, India in 1995, and pursuing Ph.D from Anna University, Chennai since 2010 . He is Lecture(SG) in Department of Computer Engineering at Seshasayee Institute of Technology, Tiruchirappalli. His area of interests are Distributed System, Mobile Computing and Cloud Computing. He is a member of Computer Society of India(CSI) .

Dr.S. Ravimaran received B.E. in Computer Science and Engineering from National Institute of Technology, Tiruchirapalli, India in 1997 and M.E. Computer and Communication Engineering from Periyar Maniammai College of Technology, Vallam, Tanjore, Anna University, Chennai, Tamil Nadu, India in 2004 and Ph.D from Anna University, Chennai in 2013 . He is Principal and Professor in Department of Computer Science and Engineering at M.A.M. College of Engineering, Tiruchirappall. His area of interests is Distributed database, Mobile transaction, Grid Computing and Cloud Computing. He is a member of IEEE and IEEE Computer Society, ISTE, CSI and Institution of Engineers.