

Dynamic Multi-objective task scheduling in Cloud Computing based on Modified particle swarm optimization

A.I.Awad, N.A.El-Hefnawy and H.M.Abdel_kader

Dept. Operations Research and Decision Support, Faculty of Computers and Information, Menoufia University
Menoufia, Egypt
Asmaa.awad@ci.menofia.edu.eg

Dept. Operations Research and Decision Support, Faculty of Computers and Information, Menoufia University
Menoufia, Egypt
nancyabbas_1@hotmail.com

Dept. Information Systems, Faculty of Computers and Information, Menoufia University
Menoufia, Egypt
hatem6803@yahoo.com

Abstract

Task scheduling is one of the most important research topics in Cloud Computing environment. Dynamic Multi-objective task scheduling in Cloud Computing are proposed by using modified particle swarm optimization. This paper presents efficient allocation of tasks to available virtual machine in user level base on different parameters such as reliability, time, cost and load balancing of virtual machine. Agent used to create dynamic system. We propose mathematical model multi-objective Load Balancing Mutation particle swarm optimization (MLBMP SO) to schedule and allocate tasks to resource. MLBMP SO considers two objective functions to minimize round trip time and total cost. Reliability can be achieved in system by getting task failure to allocate and reschedule with available resource based on load of virtual machine. Experimental results demonstrated that MLBMP SO outperformed the other algorithms in time and cost.

Keywords: Cloud computing, partial swarm, load balancing, task scheduling, Particle swarm optimization.

1. Introduction

Computing is a model consisting of services that are used in a way similar to traditional utilities such as water, electricity, gas, and telephony. In computing model, users use services that need without consider to know how they are delivered services or where services are hosted. There are several computing models [2] such as cluster computing, Grid computing. Cloud computing is known as a provider of dynamic services using very large scalable and virtualized resources over the Internet. Cloud computing present services to users such as Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) [1]. SaaS presents applications running on a cloud infrastructure to users. PaaS created or acquired applications created using

programming languages, libraries, services, and tools supported to users. IaaS enable users to provision processing, storage, networks, other fundamental computing and enable to deploy and run arbitrary software, which can include operating systems and applications. Job scheduling is a nucleus and necessary issue in Cloud Computing [3]. Task Schedule is an NP-hard problem. There are two level of Task schedule in cloud computing, first level of schedule is user level that schedule task between service provider and user. Second level is system level that schedule management resource within datacenter. Job scheduling improve efficiency and performance of cloud computing also, improve utilization of Cloud computing resources proficiently. Job scheduling is search for optimal allocation tasks to resources with consider parameters such as execution time [13], response time[8], cost[10], load balancing[9], make span[7], profit[14], speed[15], success rate[12], resource utilization[11] and so on [4]. Most scheduling algorithms don't consider important parameters like reliability and availability.

The motivation of this paper is to establish Dynamic Multi-objective task schedule mechanism. JADE (Java Agent Development Framework) is a software framework fully implemented in Java language. JADE simplifies the implementation of multi-agent systems through a middleware that claims to comply with the FIPA (Foundation for Intelligent Physical Agents) specifications and through a set of tools [6]. JADE used to develop dynamic system through create agent with two behaviors'. The Proposed Schedule technique has been built on a heuristic algorithm using Multi-objective Load Balancing Mutation a particle swarm optimization (MLBMP SO). MLBMP SO minimized round trip time and minimized total cost with respect to other algorithms. MLBMP SO achieved reliability in system by getting failure allocation tasks and reschedule with available resource based on load of virtual

machine. MLBMP SO consider the following parameters execution time, transmission time; make span, round trip time, execution cost, transmission cost and load balancing between tasks and virtual machine. The rest of the paper is organized as follows: Section 2 JADE Agent Development Framework. In Section 3, describes Proposed Dynamic System Design. Section 4 presents our Task Scheduling Problem Formulation. Section 5 presents Dynamic task schedule using MLBMP SO. Section 6 presents an experimental evaluation of the performance our heuristic. Section 7 concludes the paper and discusses some future work.

2. Related work

In [7] present a new Cloud scheduler based on Ant Colony .The goal of our scheduler is to minimize the weighted flow time of a set of PSE jobs and minimizing Make span. In the ACO algorithm, the load is calculated on each host taking into account the CPU utilization made by all the VMs that are executing on each host. This metric is useful for an ant to choose the least loaded host to allocate its VM.

In [8] focuses on distributing the equally load for all the resources. In Round Robin algorithm, the broker allocates one VM to anode in a cyclic manner. The round robin scheduling in the cloud computing is very similar to the round robin scheduling used in the process scheduling. Result of Round Robin algorithm shows better response time and load balancing as compared to the other algorithms.

In [11] use Min-Min algorithm to scheduled short jobs first, until the machines are leisure to schedule and execute long jobs. The experimental results of improved Min-Min algorithm show it can increase resource utilization rate, long tasks can execute at reasonable time and meet users' requirements.

In [19] propose a new priority based job scheduling algorithm called PJSC. The proposed algorithm is based on the theory of Analytical Hierarchy Process (AHP). The PJSC algorithm provided a discussion about some issues such as complexity, consistency and finish time. Evaluation result of this algorithm has reasonable complexity, also it decrease finish time (Make span).

In [14] has proposed a differentiated scheduling algorithm with non-preemptive priority queuing model for activities performed by cloud user in the cloud computing environment. The Qos requirements of the cloud computing user and the maximum profits of the cloud computing service provider are achieved with this algorithm.

3. JADE Agent Development Framework

JADE (Java Agent Development Environment) is framework to create agent applications. JADE simplifies the implementation of multi-agent systems through a middle-ware that claims to comply with the FIPA specifications and through a set of tools. JADE is fully implemented in Java, which enable object serialization and remote method invocation (RMI). Agent communication is achieved through message passing. Each running of the JADE environment is called a Container as it can contain several agents [5]. The set of active containers is called a Platform. Main container holds three special agents Agent Management System (AMS), Directory Facilitator (DF) and RMA as shown in figure 1. The Agent Management System (AMS) ensures that each agent in the platform has a unique name and controls the authority in the platform. The Directory Facilitator (DF) provides a Yellow Pages service which each agent can find other agent. A behavior represents a task that an agent can execute. Agent can execute more than one behavior. We chose JADE to develop dynamic system. In Proposed system create one agent with two behaviors

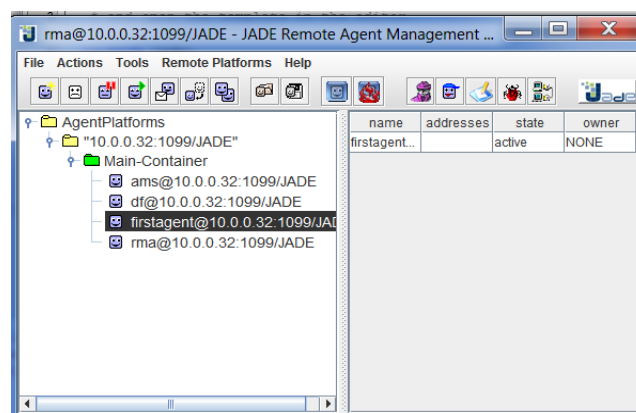


Fig.1 Jade remote agent management

4. Proposed Model Structure

Our model is to allocate tasks to virtual machines with considering reliability. The structure of our proposed model is shown in Fig.2. Java Agent Develop Environment (Jade) Tool used to create dynamic system. Proposed model consists of Agent with two behaviours.

First behaviour is responsible for

1. Task Buffer

There are millions of tasks need to execute in the cloud computing. Task buffer is responsible for collecting tasks from user. There are two stop criteria of receiving task from user and receive in next cycle. When the number of task reach to specific number or time reach to specific slot of time stop to receive other task.

2. Task and Resource Information

This phase collect the necessary information about Tasks arrived in cloud computing environment to execute. Those information such as Expected Execution Time (EET) , Expected Transmission Time (ETT), Resources-Required (RR) and Round Trip Time(RTT) . Also, this phase responsible collects information about resources in cloud computing environment. The resources in cloud computing are Datacenter, Hosts and virtual machines (VMs). Datacenter information is cost of processing, cost of memory, cost of storage, cost of BW, host list, VMs list and other information. Each Datacenter can contain more than one host and more than one VM. The information of hosts and VMs such as ram, mips, bandwidth, storage and other information. Information about tasks and machines are passing to next phase.

Second behaviour is responsible for

1. MLBMP SO

Multi-objective Load balancing mutation PSO used to schedule tasks to vms based on expected round trip time and total cost of execute task in vms. MLBMP SO used to schedule task to wm using PSO. But, PSO have two problems. First problem, tasks may failure to allocate to virtual machine. Second problem, task may allocate to more than one VM. MLBMP SO solve two problems by reschedule tasks not allocate and tasks allocate more than one. MLBMP SO take into account load balancing of each virtual machine. Solving these troubles achieve reliability, improve load balancing, users assert task executed without failure, minimize round trip time and improve other parameters.

2. Task Submission

This phase receives allocation plan from MLBMP SO. Then, the task submission allocates tasks to the resources based on plan generated by the MLBMP SO from previous phase.

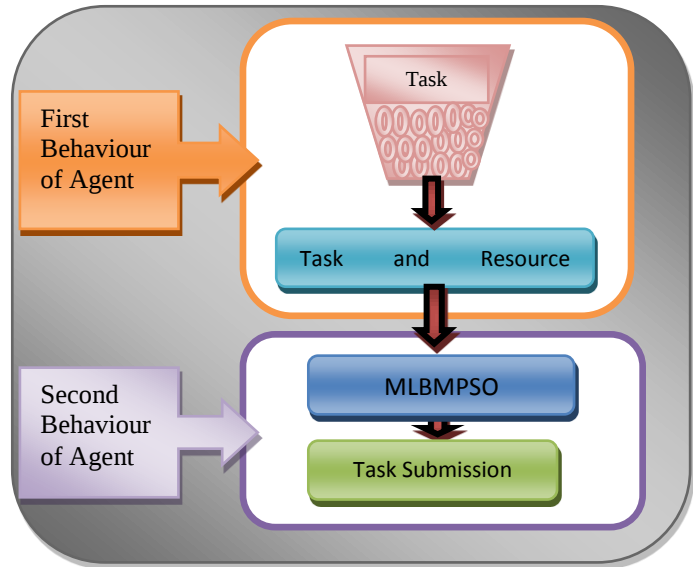


Fig.2: Proposed Model Structure

5. Task Scheduling Problem Formulation of proposed systems

There are n of tasks (t) and m of virtual machines (vms). Each task may assign to any vm. Proposed model use load balancing mutation PSO to distribute tasks to virtual machines. Model consists of two objective function and several constraints. First Objective function is to minimize Expected Round Trip Time ($ERTT_{ij}$) of task i in vm_j . Second Objective function is to minimize Expected Total Cost (ETC_{ij}) of task i in vm_j . The weighted sum approach used to solve multi-objective problem. The weighted sum strategy converts the multi-objective problem of minimizing the vector into a single objective problem by constructing a weighted sum of all the objectives. The RTT is the total time for the whole procedure involving the sending, the receiving and execution. $ERTT_{ij}$ is calculated by $(size_i / bandwidth_j) + delay + (length_i / mips_j) + delay$. $length_i$ is number of instruction of task i require to execute. $mips_j$ is number of Instructions executed by vm per second. ETC_{ij} equal $(length_i / mips_j) * Resource\ cost + (file\ size_i / bandwidth_j) * Cost / bandwidth_j$. Resource cost equal $(RAM\ of\ Virtual\ machine * Cost / memory) + (Size\ of\ Virtual\ machine * Cost / storage)$. x_{ij} equal to one or zero mean allocating task i to vm_j or not. First constraint allocate task to only one virtual machine (eq.2). Equation (3) and (4) restrict resource of all virtual machine to be less than or equal resource of datacenter. x_{ij} must assign positive number (5)

Nomenclature

N	The number of tasks
M	Number of virtual machines
X_{ij}	Decision variable of allocating task i to vm j or not
ERTT	Expected round trip time
mem_j	Memory allocate to vm j
cpu_j	Cpu allocate to vm j
Totalmem	Total memory of datacenter
Totalcpu	Total cpu of datacenter
W1	weight of first objective function
W2	weight of second objective function

Multi-objective task scheduling Mathematical Model

$$\begin{aligned} Min \ z = & W_1 * \sum_{i=0}^n \sum_{j=0}^m ERTT_{ij} * x_{ij} + \\ & W_2 * \sum_{i=0}^n \sum_{j=0}^m ETC_{ij} * x_{ij} \end{aligned} \quad (1)$$

Subject To:

$$\sum_{j=0}^m x_{ij} = 1 \ \forall \ i \quad (2)$$

$$\sum_{j=0}^m cpu_j \leq totalcpu \quad (3)$$

$$\sum_{j=0}^m mem_j \leq totalmem \quad (4)$$

$$x_{ij} \geq 0 \ \forall \ i, j \quad (5)$$

6. Multi-objective load balancing mutation Particle Swarm Optimization(MLBMP SO)

Obtaining an optimal schedule of tasks to resource with considering constraints of a bi-objective optimization problem are well-known problems in NP hard category[16]. The particle swarm optimization (PSO) algorithm is one of the heuristic techniques that used to obtain a feasible solution in reasonable time. pso proposed

by Kennedy and Eberhart [17] . Initially, the PSO algorithm generates a set of particles randomly in the D dimensional search space. Particles defined as a potential solution to a problem. Each particle is represented by a D-dimensional vector X_i where i ranges from 1 to d represent as $(x_{i1}, x_{i2}, \dots, x_{id})$. Velocity of each particle defined as $V_i = (v_{i1}, v_{i2}, \dots, v_{id})$. Each particle is updated its position and its velocity according to equations 6, 7. In the iteration t, the velocity $v_i(t)$ has been update based on $v_i(t-1)$ is the velocity of the pervious iteration, r_1, r_2 mean a uniform random variables between 0 and 1 this two random values are generated independently, c_1, c_2 are a positive constant constants called acceleration coefficients, and w is the inertia weight. It also remembers the candidate solution of best fitness value it has achieved thus far during the search (individual best position (pbest)). Also, the PSO algorithm maintains candidate solution of the best fitness value achieved among all particles in the swarm (global best position (gbest)). Equation (7) updates each particle's position using the computed $v_i(t)$. Tasks allocated to vms using pso to achieve proposed mathematical model. In Task allocation using pso has some problem .such as, some task fail to allocate to vm or task allocate to more than one vm and premature convergence. Solving previous problem in Particle Swarm Optimization add Load balancing mutation to pso as show in Fig. 3. Load balancing mutation improved reliability, availability, minimize round trip time and minimize cost .The idea of Multi-objective Load balancing mutation Particle Swarm Optimization (MLBMP SO) reschedule the failure tasks to the available (VM) and reschedule tasks that allocate to more than VM with take into account load of each vm. LBM guarantee all vm executed number of tasks appropriate with their load of vm. In LBM, First Determine failure tasks and calculate *load of virtual machines as load of vmi= (resource of vmi /total resource)*N*. Then sort tasks based on resource needed and sort vms based on load. Last Reschedule failure tasks to vm based on load of each vm as in algorithm 1.

$$v_i^{k+1} = w v_i^k + c_1 r_1 \times (pbest_i - x_i^k) + c_2 r_2 \times (gbest - x_i^k) \quad (6)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (7)$$

v_i^k	Velocity of particle i at iteration k
v_i^{k+1}	Velocity of particle i at iteration k+1
w	inertia weight
c_i	acceleration coefficients; j = 1, 2
r_i	random number between 0 and 1; i = 1, 2
x_i^k	Current position of particle i at iteration k
x_i^{k+1}	Current position of particle i at iteration k+1
pbest _i	best position of particle i
gbest	position of best particle in a population

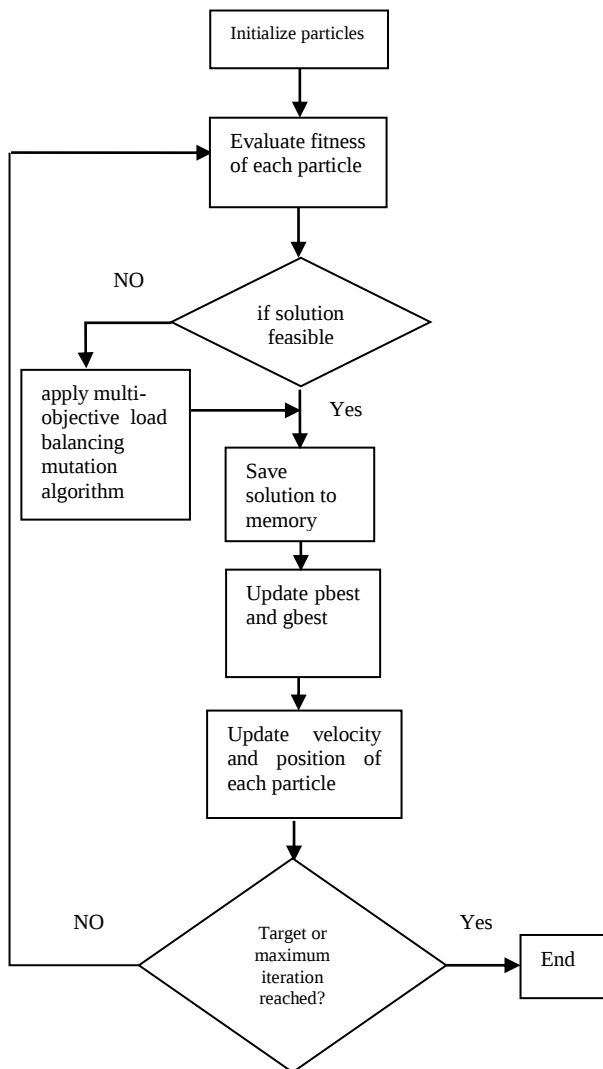


Fig.3: MLBMPSO Algorithm

Algorithm 1: Load Balancing Mutation Algorithm

Get best solution of pso

For all task { t_i } $\in T$ **do**

Determine unallocated tasks

Determine tasks allocated to more than one vm (wrong tasks)

End for

For all virtual machine { vm_i } $\in VM$ **do**

Determine current tasks allocated to vm_i (current load vm)

Determine real load of vm_i (real load vm)

End for

Sort vm based on real load

Sort wrong tasks based on resource needed

For all sorted virtual machine { svm_i } $\in VM$ **do**

For all sorted task { st_i } $\in T$ **do**

If real load $vm >$ current load vm

Schedule task from wrong tasks

Remove task from sorted tasks list

Current load $vm++$

Else

Break; // Exit to get next vm because this vm

Take load based on resource

End if

End for

End for

7. SIMULATION RESULT AND Evaluation

In this section, present data, the experiment setup and the results.

7.1 Data and Implementation

Cloudsim used to experiment proposed algorithm (MLBMPSO) and compared with, other algorithm. The experiments are implemented with 6 Datacenters with 50 VMs and 1000 tasks. The parameters of cloud simulation are shown in Table1.

7.2 Experiments and Results

In evaluating of scheduling heuristic, each task is independent to other task. The average execution time, average cost, averages round trip time and average make span are parameters used in comparison between two algorithms. These comparisons obtained after 15 independent experiments done. We compared between Multi-objective load balancing mutation pso, other algorithm [18].The result of comparisons between two algorithms based on different parameters when other

algorithm group based on time show in fig.4-7. In Figure 8-11 show comparisons between two algorithms when other algorithm group based on cost. The conclusions show that MLBMP SO is best algorithm which improve availability, reliability and consider load balancing between virtual machines. Also, minimize round trip time, execution time, make pan and cost.

Table1 1 : parameters of cloud simulation

Parameters	Value
Tasks(cloudlets)	
Length of task	1000-20000
number of task	1000
file Size	1-500
output Size	1-500
Virtual Machine	
number of VMs	50
MIPS	500-2000
VM memory(RAM)	256-2048
Bandwidth	500-1000
Datacenter	
Number of Datacenter	6
Number of Host	3-6

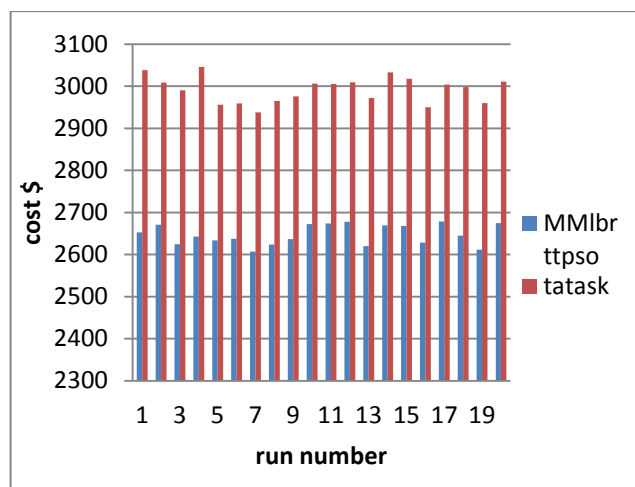


Fig.4: Average Cost

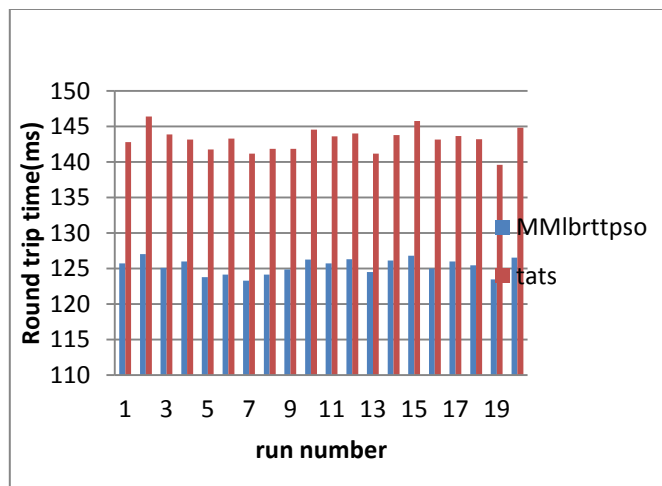


Fig.5: Average Round Trip Time

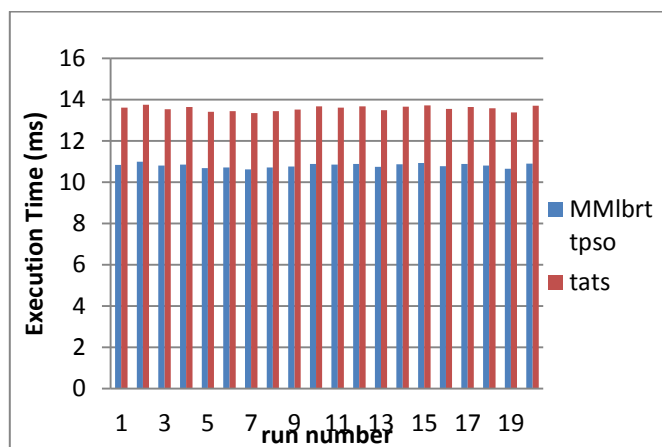


Fig.6: Average Execution Time

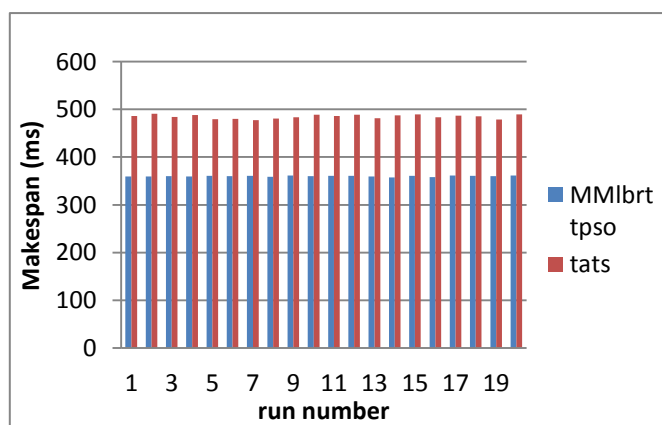


Fig.7: Average Makespan

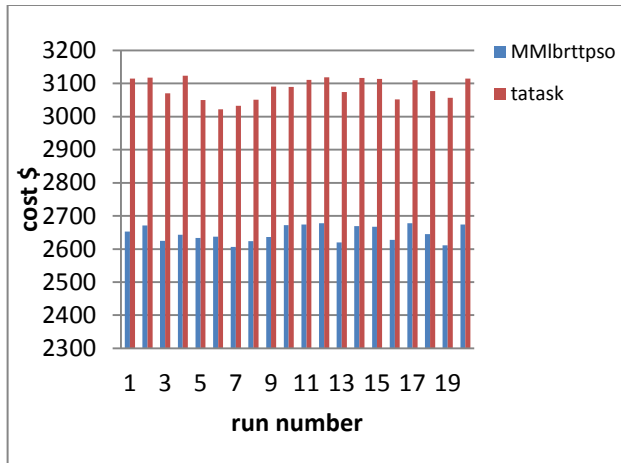


Fig.8: Average Cost

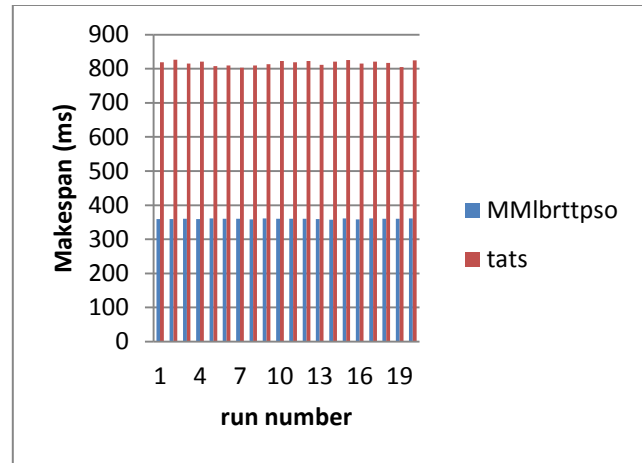


Fig.11: Average Makespan

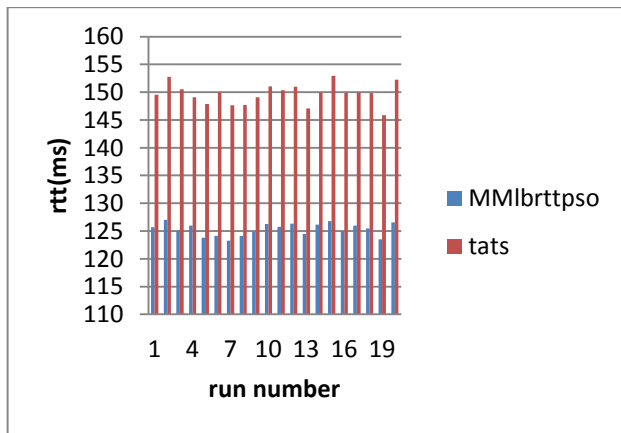


Fig.9: Average Round Trip Time

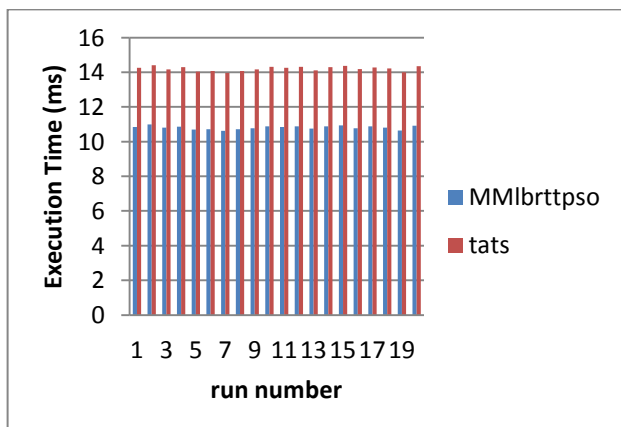


Fig.10: Average Execution Time

8. Conclusions

Task scheduling is one of the most important issues that effect in performance of Cloud Computing environment. There are many task scheduling algorithms which not consider important parameters such as availability and reliability. In this paper, we present Dynamic Multi-objective task a scheduled based on load balancing Mutation Particle Swarm Optimization (MLBMP SO). MLBMP SO improves the Reliability of cloud computing and consider availability of resources compared to other algorithms. MLBMP SO used to minimize total cost, minimize round trip time, improves task completion time, improve execution cost, good distribution of tasks onto resources, achieve load balancing between tasks and virtual machine and minimize the complexity in cloud computing environment. In addition, proposed algorithm consider the load balancing when schedule tasks to available and reschedule failure tasks to achieve reliability. It can be used to allocate any number of tasks and resources.

References

- [1] P. Mell, and T. Grance. (2011). The NIST Definition of Cloud Computing .NIST[Online]. Available: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>
- [2] R.Buyya ,C.S. Yeo, S.Venugopal ,J.Broberg and I.Brandic “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility” ,Future Generation Computer Systems , 2009 ,vol.25,no.6,pp.599–616.
- [3] Paul, M. et al. “Task scheduling in cloud computing using credit based assignment problem”, International journal of Comput. Sci. Eng, 2011, pp. 26-30.
- [4] A.I.Awad, N.A.El-Hefnawy and H.M.Abdel kader ,” Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments“ , in International

- Conference on Communication, Management and Information Technology (ICCMIT 2015),2015.
- [5] G.Caire, JADE Tutorial: JADE Programming for Beginners, <http://sharon.cselt.it/projects/jade/>
 - [6] P. Angin and B. Bhargava, "An Agent-based Optimization Framework for Mobile-Cloud Computing", Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2013, vol. 4, pp. 1-17.
 - [7] C.Mateos, E.Pacini and C. G.Garino,, "An ACO-inspired algorithm for minimizing weighted flowtime in cloud-based parameter sweep experiments", Advances in Engineering Software, 2013, vol.56,pp. 38–50.
 - [8] P.Samal and P.Mishra, "Analysis of variants in Round Robin Algorithms for load balancing in Cloud Computing", International Journal of Computer Science and Information Technologies, 2013, Vol. 4, no.3, pp. 416-419.
 - [9] B.Mondala, K.Dasgupta and P.Duttatb,," 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, 2012, vol.4, PP. 783 – 789,.
 - [10] S.Pandey, L.Wu, S.M.Guru2 and R.Buyya,," 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), 2010, pp. 400 - 407.
 - [11] G.Liu, J.Li and J.Xu, "An Improved Min-Min Algorithm in Cloud Computing", Proceedings of the International Conference of Modern Computer Science and Applications, 2012, vol .191,pp.47-52.
 - [12] W.Wang, G.Zeng, D.Tang and J.Yao,"Cloud-DLS: Dynamic trusted scheduling for Cloud computing", Expert Systems with Applications , 2012,vol.39,no.3, PP. 2321–2329,.
 - [13] B. Xu, C.Zhao, E.Hu and B.Hu,,"Job scheduling algorithm based on Berger model in cloud environment", Advances in Engineering Software, 2011 ,vol. 42,no.7, PP. 419–425.
 - [14] S.Ambike, D.Bhansali, J.Kshirsagar and J.Bansiwal " An Optimistic Differentiated Job Scheduling System for Cloud Computing" International Journal of Engineering Research and Applications (IJERA) , 2012, Vol. 2, no. 2, pp.1212-1214.
 - [15] H.Zhong, K.Tao and X.Zhang ,," An Approach to Optimized Resource Scheduling Algorithm for Open-source Cloud Systems ", in Fifth Annual China Grid Conference, ,on 16-18 July ,2010 , pp. 124 - 129.
 - [16] A. S. Ajeena Beegom and M. S. Rajasree,," A Particle Swarm Optimization Based Pareto Optimal Task Scheduling in Cloud Computing ",in Proceedings of 5th International Conference, ICSI 2014, October 17-20, 2014 , pp 79-86.
 - [17]Eberhart R , Kennedy J, "A New Optimizer Using Particle Swarm Theory", 6th International Symposium. Micro Machine and Human Science, Nagoya;1995,p. 39-43.
 - [18] M.Choudhary and S.K.Peddoju "A Dynamic Optimization Algorithm for Task Scheduling in Cloud Environment" ,International Journal of Engineering Research and Applications (IJERA) , , 2012 Vol. 2, no.3, pp.2564-2568.
 - [19] S.Ghanbaria and M.Othmana,," A Priority based Job Scheduling Algorithm in Cloud Computing",in International

Conference on Advances Science and Contemporary Engineering , 2012 ,vol.50, and PP. 778 – 785.