Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 65 (2015) 920 – 929

Procedia Computer Science

International Conference on Communication, Management and Information Technology (ICCMIT 2015)

Enhanced Particle Swarm Optimization For Task Scheduling In Cloud Computing Environments

A.I.Awad[a], N.A.El-Hefnawy[b], H.M.Abdel_kader[c]

[a,b] *Dept. Operations Research and Decision Support*
Faculty of Computers and Information, Menoufia University
Menoufia, Egypt
[c]*Dept. Information Systems*
Faculty of Computers and Information, Menoufia University
Menoufia, Egypt

Abstract

The most important requirement in cloud computing environment is the task scheduling which plays the key role of efficiency of the whole cloud computing facilities. Task scheduling in cloud computing means that to allocate best suitable resources for the task to be execute with the consideration of different parameters like time, cost, scalability, make span, reliability, availability, throughput, resource utilization and so on. The proposed algorithm considers reliability and availability. Most scheduling algorithms do not consider reliability and availability of the cloud computing environment because the complexity to achieve these parameters. We propose mathematical model using Load Balancing Mutation (balancing) a particle swarm optimization (LBMPSO) based schedule and allocation for cloud computing that takes into account reliability, execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and virtual machine .LBMPSO can play a role in achieving reliability of cloud computing environment by considering the resources available and reschedule task that failure to allocate. Our approach LBMPSO compared with standard PSO, random algorithm and Longest Cloudlet to Fastest Processor (LCFP) algorithm to show that LBMPSO can save in make span, execution time, round trip time, transmission cost.

Peer-review under responsibility of Universal Society for Applied Research

Keywords: Cloud computing ; particle swarm optimization; scheduling strategy; load balancing; virtual machine.

*Corresponding author. Tel.: +02-012-874-371-20; fax: +02-048-222-3694.
E-mail address: asmaa.awad@ci.menofia.edu.eg , nancyabbas_1@hotmail.com , hatem6803@yahoo.com

1. Introduction

Cloud computing is defined by the National Institute of Standards and Technology (NIST) as “a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g.,

Peer-review under responsibility of Universal Society for Applied Research
doi:10.1016/j.procs.2015.09.064

networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."[1]. Cloud computing is a new kind of shared infrastructure which can attach huge pools of systems, provides users with a variety of storage and computing resources via the internet [6]. The major potential of the cloud comes from ability to provide anything as service "Xaas". "Xaas" means one or more of these services such that Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) [4]. Although cloud computing providing different services, there are several problems in cloud computing. Among those problems, task scheduling is one of the major problems. Task Scheduling problems, "which are concerned with searching for optimal (or near-optimal) real-time and predictive schedules subject to a number of constraints" [2]. Task scheduling means optimal usage of the available resources. The high performance of cloud computing environment greatly depends on efficiency resource scheduling. The Most existing scheduling algorithms consider various parameters like time [5], cost [6], make span [3], speed[10], scheduling success rate[12], resource utilization[13] and so on[14, 15]. However, there are important parameters like reliability and availability should be considered [2]. Task scheduling in cloud computing is NP-hard problem, PSO as one of the heuristic algorithms has been applied in solving scheduling problem and other NP-hard problems [6]. The main purpose of this paper is to enhance the overall performance of task scheduling problem in cloud computing environment. The Proposed technique has been built on a heuristic algorithm using Load Balancing Mutation a particle swarm optimization (LBMPSO). This algorithm is proposed to achieve reliability in task scheduling. LBMPSO takes into account execution time, transmission time, make span, round trip time, transmission cost and load balancing between tasks and virtual machine. The rest of the paper is organized as follows: Section 2 presents related work. In Section 3, describes Proposed System Design. Section 4 presents our Task Scheduling Problem Formulation. Section 5 presents load balancing mutation Particle Swarm Optimization (LBMPSO). Section 6 presents an experimental evaluation of the performance our heuristic. Section 7 concludes the paper and discusses some future work.

2. Related WORK

There are many of research worked in resource scheduling to improve efficiency in cloud computing. Most of these researches improve the cost, waiting time, make span, resource utilization, execution time and round trip time. But, not consider other important parameters such as reliability, availability, scheduling success rate, speed and scalability. The complexity is reasoning to not consider these parameters.

In [6] presented a Quality of service (QoS)-based Genetic Hybrid Particle Swarm Optimization (GHPSO) to schedule applications to cloud resources. In GHPSO, crossover and mutation of genetic algorithm is embedded into the particle swarm optimization algorithm (PSO). The simulation results show that the GHPSO achieves better performance than standard particle swarm algorithm used in minimize costs within a given execution time.

In [7] formulated a model for task scheduling and propose a particle swarm optimization (PSO) algorithm which is based on small position value rule to minimize the cost of the processing. . By virtue of comparing PSO algorithm with the PSO algorithm embedded in crossover and mutation and in the local research, the experiment results show the PSO algorithm not only converges faster but also runs faster than the other two algorithms in a large scale. The experiment results prove that the PSO algorithm is more suitable to cloud computing.

In [8] presented a particle swarm optimization (PSO) based heuristic to schedule applications to cloud resources that takes into account both computation cost and data transmission cost. It is used for workflow application by varying its computation and communication costs. The experimental results show that PSO can achieve cost savings and good distribution of workload onto resources.

In [9] found a solution that meets the user-preferred Quality of Service (QoS) parameters. The work presented focuses on scheduling cloud workflows. With this algorithm, a significant improvement in CPU utilization is achieved.

In [10] proposed an optimized scheduling algorithm to achieve the optimization or sub-optimization for cloud scheduling. In this algorithm an Improved Genetic Algorithm (IGA) is used for the automated scheduling policy. The tests illustrate that the speed of the IGA almost twice the traditional GA scheduling method and the utilization rate of resources always higher than the open-source IaaS cloud systems.

In [11] improved cost-based scheduling algorithm for making efficient mapping of tasks to available resources in cloud. This scheduling algorithm measures both resource cost and computation performance, it also improves the computation/communication ratio by grouping the user tasks according to a particular cloud resource's processing capability and sends the grouped jobs to the resource.

3. Proposed Model Structure

The inspiration of our model is to allocate tasks to virtual machines with considering reliability. The structure of our proposed model is shown in Fig.1. Proposed model consists of five phases. Model phase's concepts are:

- *Task Buffer*

There are millions of users require to execute tasks in the cloud computing. Task buffer is responsible for collecting tasks from user.

- *Task Information*

This phase provides the necessary information of Tasks arrived into cloud computing environment for execution. Those information such as Expected Execution Time (EET), Expected Transmission Time(ETT), Resources-Required (RR) and Round Trip Time(RTT) .

- *Resource Information*

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 host list, VMs list, storage list and cost of memory, cost of BW and other information. Each host can contain more than one VM. The information of hosts and VMs such as ram, mips, bandwidth and other information. These information machines are passing to next phase.

- *LBMPSO*

Load balancing mutation PSO used to reschedule tasks that failure to schedule. PSO have two problems. First problem, tasks may failure to allocate to virtual machine. Second problem, task may allocate to more than one VM. In this phase solve the problems by reschedule wrong tasks and take in account load balancing of virtual machine. Solving these problems help to achieve reliability, users assert task executed without failure, minimize execution, minimize round trip time and improve other parameters.

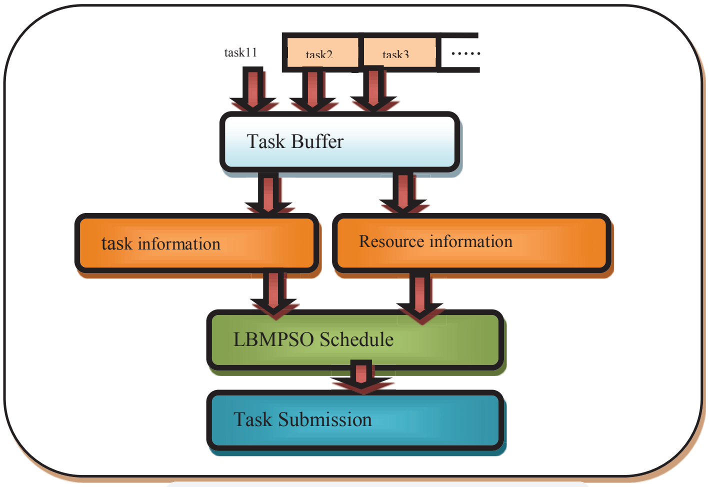


Fig.1: Proposed Model Structure

- *Task Submission*

This phase responsible receives allocation plan from previous phase. Then, allocates each task to virtual machines based on plan.

4. Task Scheduling Problem Formulation of proposed systems

There are several tasks (t) and several virtual machines (vms). There are n tasks and m number of virtual machines. Each task may allocate to any vm. Fig.2 shows mapping of Tasks to virtual machines. Each task must schedule to only one virtual machine. PSO attempts to select optimal distribution of tasks to virtual machines for achieving objective. Three mathematical models proposed for task scheduling. Each model consists of objective function and several constraints. Objective function of first model is to minimize execution time based on expected execution time (EET_{ij}) of task i in vmj. Equation (1) used in calculate processing time as: EET(processing time) = $length_i$ / $mips_j$. lengthi is number of instruction of task i require to execute. mipsj is number of Instructions executed by vm per second. Second objective function is to minimize transmission time ($ETRT_{ij}$)(6). Expected transmission time ($ETRT_{ij}$) of task i to vm j responsible for achieving second objective function. $ETRT_{ij}$ equals file size / bandwidth. To minimize round trip time (RTT) (3) is achieved by third mathematical model. The RTT is the (latency) time for the whole procedure involving the sending and the receiving. $ERTT_{ij}$ is expected round trip time calculate by $ETRT_{ij}$ + delay + EET_{ij} + delay. x_{ij} is allocating task i to vm j or not . The value of x_{ij} may one or zero. Each model has the same constraints. Each Task allocate to only one virtual machine achieve by first constraint in (2). Equation (3) and (4) represent resource of all virtual machine less than or equal resource of datacenter. x_{ij} 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
EET	Expected execution time
ETRT	Expected transmission time
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

First Mathematical Model Based On Expected Execution Time

$$\text{Min } z = \sum_{i=0}^{n} \sum_{j=0}^{m} EET_{ij} * x_{ij} \qquad (1)$$

Subject To.

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

$$\sum_{j=0}^{m} cpu_j \le totalcpu \qquad (3)$$

$$\sum_{j=0}^{m} mem_j \le totalmem \qquad (4)$$

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

Objective Function of Second Mathematical Model Based on Expected Transmission Time

$$\text{Min } z = \sum_{i=0}^{n} \sum_{j=0}^{m} ETRT_{ij} * x_{ij} \qquad (6)$$

Objective Function of Third Mathematical Model Based on Expected Round Trip Time

$$\text{Min } z = \sum_{i=0}^{n} \sum_{j=0}^{m} ERRT_{ij} * x_{ij} \tag{7}$$

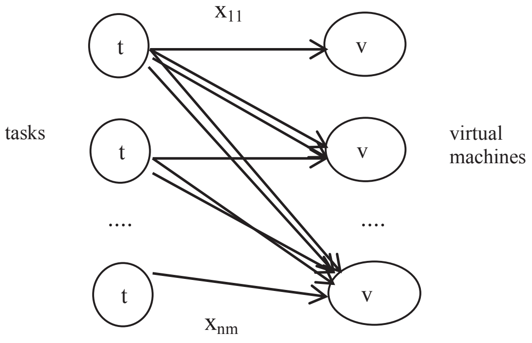


Fig.2: tasks mapping to virtual machine

5. load balancing mutation Particle Swarm Optimization (LBMPSO)

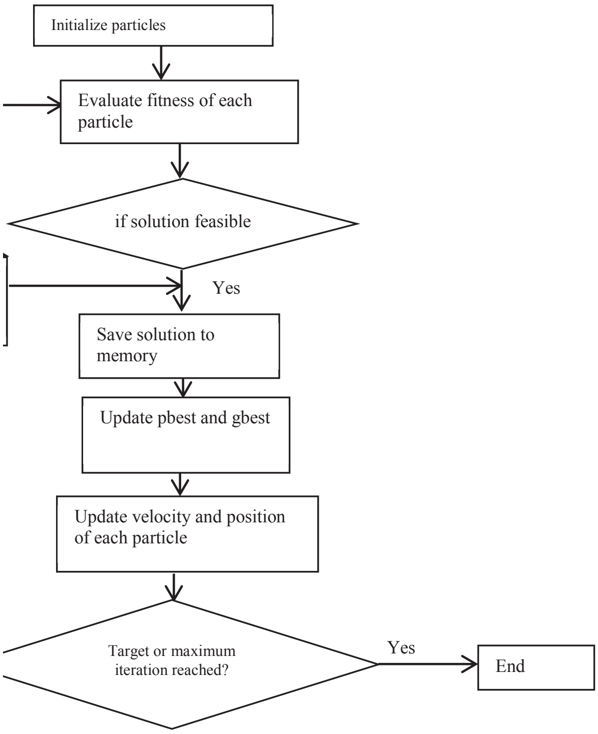
The particle swarm optimization (PSO) algorithm is a population-based search algorithm based on the simulation of the social behavior of birds within the flock and fish school proposed by Kennedy and Eberhart [16]. Let us define the notation adopted in this paper during the D-dimensional search space, each particle in this space defined as a potential solution to a problem, i.e. the i^{th} particle of the swarm represented as $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$, and its velocity defined as $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$. The update the particles at each generation are accomplished according to (16), (17). In the iteration t, the velocity $v_i(t)$ has been update to pull the particle i_{th} towards its own best position x_{pi} and the best position for all the particles x_g that has the best fitness value until the preceding generation. Also it is observe, the current velocity of each iteration t 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, c1, c2 are a positive constant, and w is the inertia weight. Equation (17) updates each particle's position in the solution hyperspace using the computed v_i (t) and the coefficients c and d that could be set to unity without loss of generality. Pso was used to allocate tasks to vms but, there are some problems. First problem, some task doesn't allocate to vm. Second problem some tasks allocate to more than one vm. Third problem is premature convergence. Load balancing mutation added to Particle Swarm Optimization to solve previous problem as show in Fig. 3. Load balancing mutation improved in other parameters such as minimize make span, minimize execution time, minimize round trip time and minimize cost. Also, achieve reliability and load balancing. The idea of Load balancing mutation Particle Swarm Optimization (LBMPSO) reschedule the failure tasks to the available (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 .Second calculate load of virtual machines as load of vmi=(resource of vm_i /total resource)*N. Third 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.

$$v_i^{k+1} = wv_i^k + c_1 r_1 \times \left(pbest_i - x_i^k \right) + c_2 r_2 \times \left(gbest - x_i^k \right) \tag{16}$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \tag{17}$$

Nomenclature

$\boldsymbol{v_i^k}$	velocity of particle i at iteration k
$\boldsymbol{v_i^{k+1}}$	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
$\boldsymbol{x_i^k}$	current position of particle i at iteration k
$\boldsymbol{x_i^{k+1}}$	current position of particle i at iteration k+1
$pbest_i$	best position of particle i
gbest	position of best particle in a population


Initialize particles
Evaluate fitness of each particle
if solution feasible
Yes
Save solution to memory
Update pbest and gbest
Update velocity and position of each particle
Target or maximum iteration reached?
Yes
End

.3: LBMPSO Algorithm

Algorithm 1: Load Balancing Mutation Algorithm

```
Get best solution of pso
 for all task { $t_i$ } Є T do
    Determine unallocated tasks
    Determine tasks allocated to more than one vm(wrong tasks)
  end for
  for all virtual machine { $vm_i$ } ϵ 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 {$svm_i$ } Є VM do
     for all sorted task {$st_i$ } Є 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
```

6. SIMULATION RESULT AND Evaluation

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

6.2 Data and Implementation

Cloudsim used to experiment proposed algorithm (LBMPSO) and compared with longest vm longest cloudlet algorithm, random algorithm, mutation pso without consider load balancing and standard pso. The experiments are implemented with 6 Datacenters with 50 VMs and 1000 tasks. The parameters of cloud simulation are shown in Table3.

6.2 Experiments and Results

We evaluated the scheduling heuristic using independent task to each other. The number of executions 15 represents the number of independent experiments done. The following experiments, the parameters the average execution time, average cost, average round trip time and average makes pan used in comparison between different algorithms. We compared between round trip time load balancing mutation pso, round trip time random, round trip time Longest Cloudlet to Fastest Processor ,round trip time pso ,round trip time mutation pso, transmission time load balancing mutation pso, transmission time random, transmission time Longest Cloudlet to Fastest Processor, transmission time pso, execution time mutation pso, time load balancing mutation pso, execution time random, execution time Longest Cloudlet to Fastest Processor, execution time pso and execution time mutation pso. The result of comparisons between different algorithms to improve execution time show in fig. 4 -7. In Fig. 9-12 show the comparison between different algorithms based on different parameters as second mathematical formula. The result is the best when take in account round trip time as show in Fig.14-17. The graph in Fig. 8, 13, 18 by average cost, average RTT, average ET and average makespan of algorithms obtained after 15 independent executions. The conclusions show that LBMPSO in third formula based on round trip time the best algorithm which minimizes round trip time, execution time, makes pan and cost as fig.19. Also, consider load balancing and achieve availability and reliability.

Table 1:Resource Parameters

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

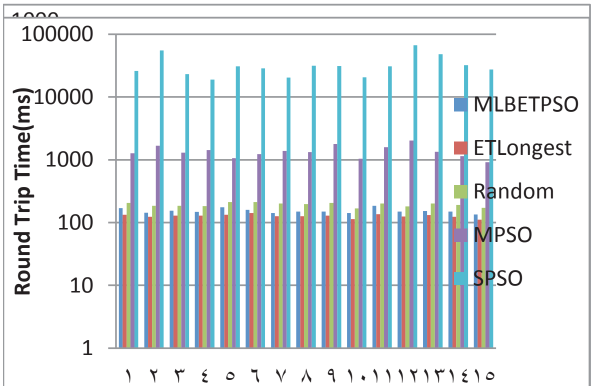


Fig.5: Average RTT based on Expected Execution Time

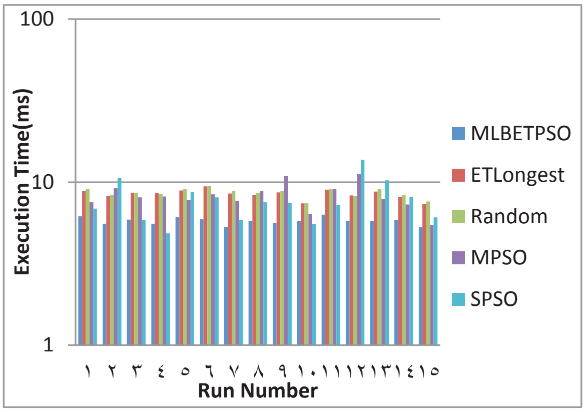


Fig.6: Average ET based on Expected Execution Time

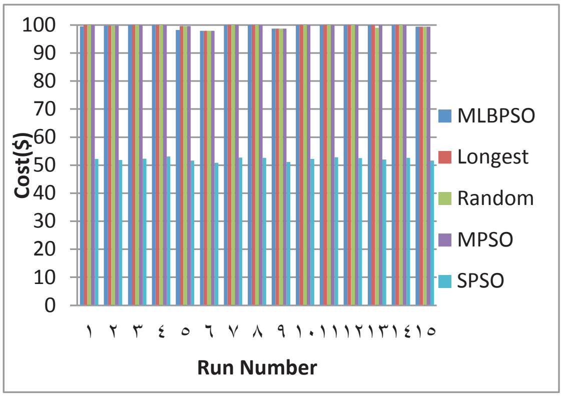


Fig.9: Average Cost based on Expected Transmission Time

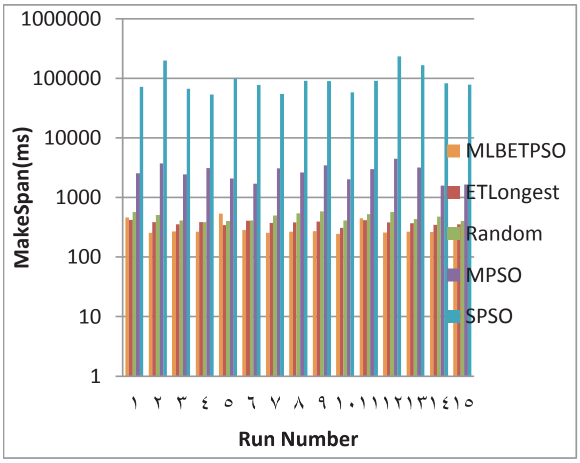


Fig.7: MakeSpan based on Expected Execution Time

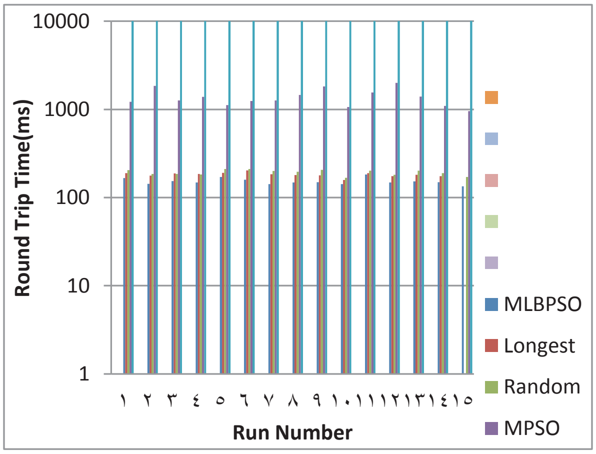


Fig.10: Average RTT based on Expected Transmission Time

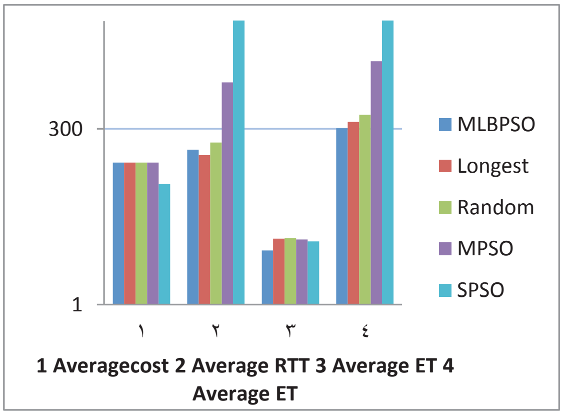


Fig. 8: All comparison based on Expected Execution Time

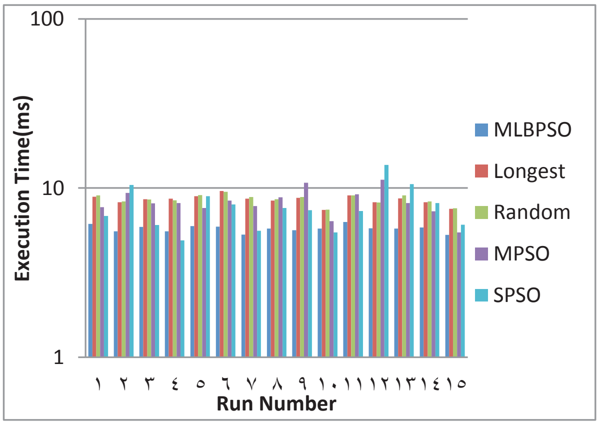


Fig.11: Average ET based on Expected Transmission Time

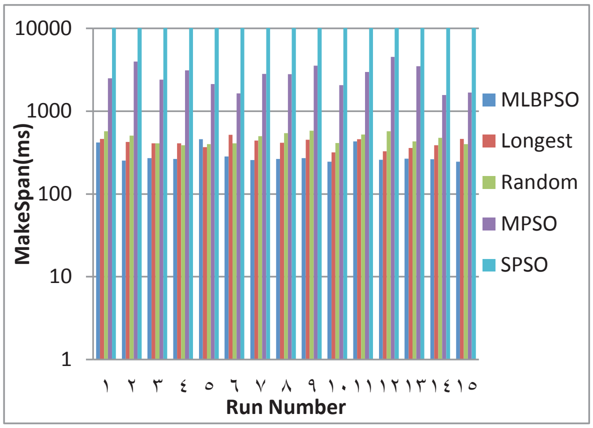


Fig.12: MakeSpan based on Expected Transmission Time

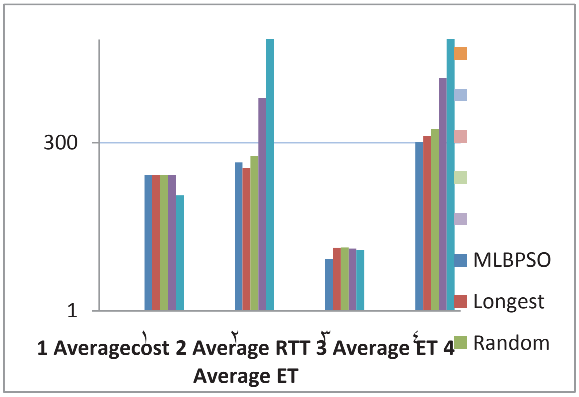


Fig.13: All comparison based on Expected Transmission Time

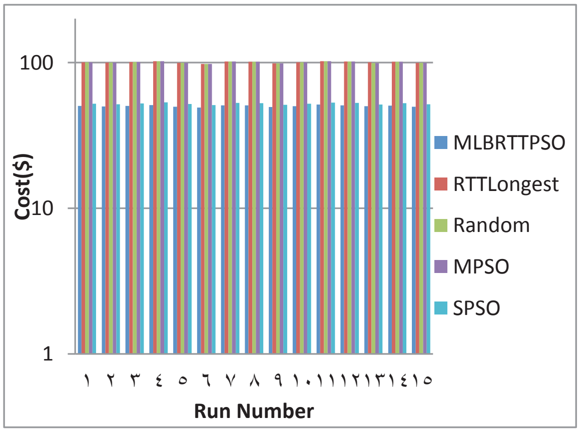


Fig.14: Average Cost based on Expected Round Trip Time

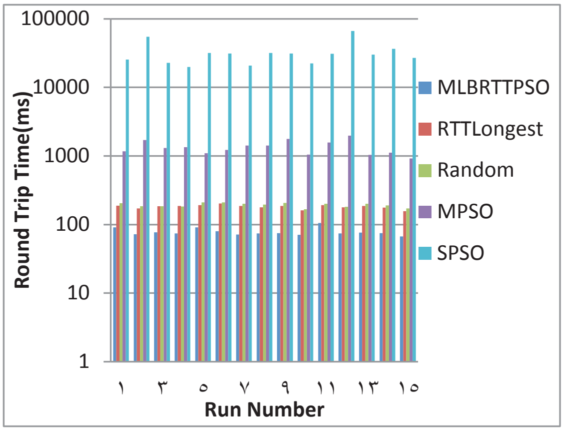


Fig.15: Average RTT based on Expected Round Trip Time

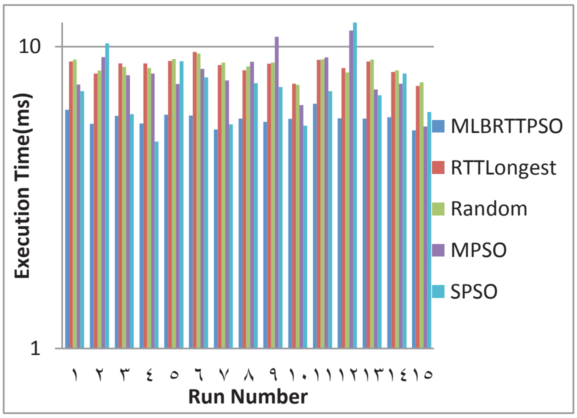


Fig.16: Average ET based on Expected Round Trip Time

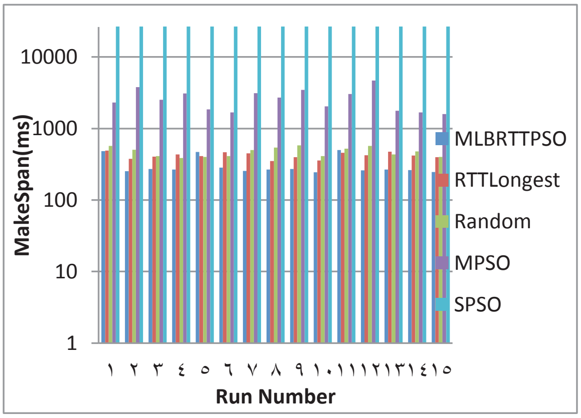


Fig.17: MakeSpan based on Expected Round Trip Time

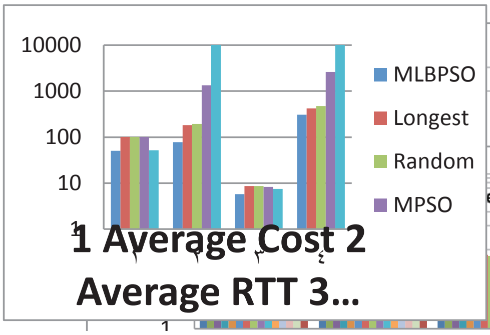


Fig.18: All comparison based on Expected Round Trip Time

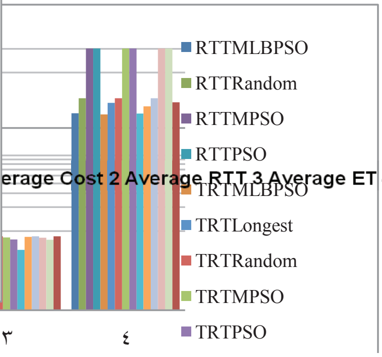


Fig.19: All Comparison

7. Conclusions and Future Work

In this paper, task a scheduled based on load balancing Mutation Particle Swarm Optimization (LBMPSO).LBMPSO used to minimize cost, minimize round trip time, minimize execution time, minimize transmission time, achieve load balancing between tasks and virtual machine, consider available resource and minimize the complexity in cloud computing environment. LBMPSO improves the Reliability of cloud computing and good distribution of tasks onto resources compared to other algorithms. We found that round trip time load balancing mutation PSO can achieve the best compared to other algorithms. In addition, proposed algorithm take in account the load balancing when distributing tasks to available resources, tasks assign as earlier as possible, finished as earlier as possible and reschedule failure tasks. It can be used for any number of tasks and resources.

References

1. Kundra V, Federal cloud computing strategy; 2011
2. Kumar S , Balasubramanie P,"*Dynamic Scheduling for Cloud Reliability using Transportation Problem*," Journal of Computer Science (8);2012, p. 1615-1626.
3. Raju R,Babukarthik R G, Chandramohan D, Dhavachelvan P, Vengattaraman T. "*Minimizing the makespan using Hybrid algorithm for cloud computing*", 3rd IEEE International Conference on Advance Computing Conference (IACC); 2013, p. 957 - 962.
4. N.Calheiros R,Ranjan R, Beloglazov A, A.F.De Rose C, Buyya R, "*CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms*," Software: Practice and Experience ,41;2011, p. 23-50.
5. Kumar P,Verma A, "*Independent Task Scheduling in Cloud Computing by Improved Genetic Algorithm*", International Journal of Advanced Research in Computer Science and Software Engineering (2: 5);2012 ,p.137-142.
6. Jun Xue S, Wu W, "*Scheduling Workflow in Cloud Computing Based on Hybrid Particle Swarm Algorithm*," TELKOMNIKA Indonesian Journal of Electrical Engineering (10:7); 2012, pp. 1560-1566.
7. Guo L,Zhao S,Shen S, Jiang C, "*Task Scheduling Optimization in Cloud Computing Based on Heuristic Algorithm*," Journal of Networks (7);2012, pp. 547-553.
8. Pandey S, Wu L, Guru S, Buyya R,"*A Particle Swarm Optimization-Based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments*",24th IEEE International Conference on Advanced Information Networking and Applications (AINA);2010.
9. Varalakshmi P, Ramaswamy A, Balasubramanian A,Vijaykumar P,"*An Optimal Workflow Based Scheduling and Resource Allocation in Cloud*",Advances in Computing and Communications, First International Conference, ACC;2011, pp. 411-420.
10. Zhong H, Tao K, Zhang X, "*An Approach to Optimized Resource Scheduling Algorithm for Open-Source Cloud Systems*" , in Fifth Annual China Grid Conference;2010.
11. Selvarani S, Sadhasivam G, "*Improved cost-based algorithm for task scheduling in cloud computing*", IEEE International Conference on Computational Intelligence and Computing Research (ICCIC); 2010.
12. Xu M, Cui L,Wang H,Bi Y, "*A Multiple QoS Constrained Scheduling Strategy of Multiple Workflows for Cloud Computing*", IEEE International Symposium on Parallel and Distributed Processing;2009.
13. Dinesh K , Poornima G,Kiruthika K,"*Efficient Resources Allocation for Different Jobs in Cloud*", International Journal of Computer Applications (56:10);2012.
14. A. Bala A,Chana I. "A *Survey of Various Workflow Scheduling Algorithms in Cloud Environment*". 2nd National Conference on Information and Communication Technology;2011,p. 26-30.
15. Kaur H,Singh M, "*Review of Various Scheduling Techniques in Cloud Computing*," international journal of networking and parallel computing (1:2);2012.
16. Eberhart R , Kennedy J, "*A New Optimizer Using Particle Swarm Theory*", 6th International Symposium. Micro Machine and Human Science, Nagoya;1995,p. 39-43.