British Journal of Mathematics & Computer Science
6(3): 247-256, 2015, Article no.BJMCS.2015.076
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

SCIENCEDOMAIN

Performance Evaluation of Two Stage Scheduling Algorithm in Cloud Computing

R. Krishnam Raju Indukuri[1*], P. Suresh Varma[2] and M. V. Rama Sundari[3]

[1]Department of CS, Padmasri Dr. BVRICE, Bhimavaram, A.P, India.
[2]Department of CS, Adikavi Nannaya University, Rajamundry, A.P, India.
[3]Department of IT, GIET, Rajamundry, A.P, India.

Article Information

DOI: 10.9734/BJMCS/2015/12641
Editor(s):
(1) Doina Bein, Applied Research Laboratory, Pennsylvania State University, USA.
(2) Paul Bracken, Department of Mathematics, The University of Texas-Pan American Edinburg, USA.
Reviewers:
(1) Anonymous, Turkey.
(2) Anonymous, India.
(3) Anonymous, China.
(4) Anonymous, China.
(5) Anonymous, China.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=733&id=6&aid=7611

Original Research Article

Received: 10 July 2014
Accepted: 04 December 2014
Published: 03 January 2015

Abstract

Cloud computing is an evolutionary approach that completely changes how computing services are produced, priced and delivered. Cloud computing allows to access services that reside in a distant datacenter, other than local computers. Resource provisioning is the key process in cloud computing. The Virtual Machine (VM) is a software implementation of a machine that executes programs like a physical machine. Two stage scheduling is a novel approach in cloud computing. In this case a job may request two virtual machines in sequence to complete their needs. This paper presents a novel two stage scheduling algorithm to schedule the given job requests in cloud environment by extending Johnson's Scheduling algorithm. Simulation results show that this algorithm reduces average waiting time and total elapsed time when compared to other scheduling algorithms.

Keywords: Cloud computing; two stage scheduling.

1 Introduction

Cloud computing is the model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service

**Corresponding author: irkbvrice@gmail.com, vermaps@yahoo.com;*

provider interaction. The cloud computing environment refers to the hardware and systems software in the datacenters that provide computing resources as services. Cloud computing provides on-demand computing resources for various applications. This paradigm provides a cost effective solution for running business applications by using Virtualization technologies, scalable distributed technologies and data management techniques with on demand Pay-As-You-Go (PAYG) pricing model. Cloud computing has three deployment models Public Cloud, Private Cloud and Hybrid Cloud. Cloud computing has three service models Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS).

Scheduling is the process of deciding how to commit resources between a variety of possible tasks. In cloud computing, a typical datacenter consists of computing machines connected by high speed network links. This environment is well suited for the computation of large, diverse group of tasks. Tasks belonging to different users are no longer distinguished one from the other. Scheduling algorithms in Cloud computing comes under the class of NP-complete. Heuristic is often applied as a suboptimal algorithm to obtain relatively good solutions. Scheduling heuristics can be static or dynamic. Static heuristic is suitable for the situation where the complete set of tasks is known prior to execution, while dynamic heuristic performs the scheduling when a task arrives.

A virtual machine (VM) is a software implemented abstraction of the underlying hardware, which is presented to the application layer of the system. Virtual machines may be based on specifications of a hypothetical computer or emulate the computer architecture and functions of a real world computer. Virtual Machines are considered an ideal vehicle for resource provisioning in Cloud Computing. Virtual Machine allocation for job requests in cloud computing is key operation in scheduling process. In Virtual Machine model, hardware, software, and availability can be provisioned for job requests. Additionally, Virtual Machines' capability in getting suspended, resumed, or migrated without major utilization loss has proved to be useful in resource management.

In this paper we proposed a novel two stage scheduling algorithm to schedule job requests in cloud computing by extending Johnson's Scheduling algorithm. The simulation results show the average waiting time and total elapsed time which are significantly reduced when compared to other scheduling algorithms.

2 Related Work

In Task Scheduling Samuel [1] presented an algorithm for job sequencing with resource constraints to achieve minimum total elapsed time using partial enumeration. This algorithm uses partial enumeration of mixed integer program. This algorithm employs maximum flow computation as a check for feasibility with respect to available resources. Richard. J [2] discussed a classic three machine scheduling model with optimal permutation of n items by considering total elapsed time as objective function. This scheduling algorithm uses Integer Linear Programming approach to solve the problem. Edward Ignall [3] proposed flow shop scheduling using branch and bound technique for n jobs and three machines considering mean completion time as objective function. This scheduling algorithm uses lower bound for makespan of all nodes emanating from a given node. Makespan is the time difference between the start and finish of a sequence of jobs.

In Workflow Scheduling Jia Yu and Rajkumar Buyya [4] presented a budget constrained scheduling of workflow applications on utility grids using genetic algorithms and proposed a budget constraint based scheduling using genetic algorithm, which minimizes execution time while meeting a specified budget for delivering results. Jia Yu, Rajkumar Buyya and Chen Khong Tham [5] further enhanced cost-based scheduling of scientific workflow Applications on Utility Grids with a novel algorithm which minimizes execution cost while meeting the deadline for delivering results.

Hamid and Radu [6] discussed a pricing model and truthfull mechanism for task scheduling by considering monetory cost and completion time as objective functions.

In the Virtual Machine Scheduling of Cloud Computing area Quyet Thang Nguyen [7] explained Virtual Machine allocation using mixed linear programming method in cloud computing for minimizing total execution time on each machine where each job needs to use a number of virtual machines during the given fixed time. Xiao cheng Liu [8] described a Priority-based Consolidation of parallel workloads in the Cloud. Zhen Xiao [9] discussed a Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment. In this approach the concept of skewness was introduced to measure the unevenness in the multi-dimensional resource allocation of a server. E. Ilavarasan and P. Thambidurai [10] identified the Scheduling of an application modeled by Directed Acyclic Graph (DAG). and explained a scheduling algorithm based on list scheduling, namely, low complexity Performance Effective Task Scheduling (PETS) algorithm for heterogeneous computing systems with less complexity, which provides effective results for applications represented by DAGs.

Scheduling policies which consider resource prices and also users budget and deadlines are called market oriented scheduling policies. Mohsen Amini, Rajkumar Buyya [11] discussed market oriented scheduling policies which aim to satisfy application deadlines. Navendu Jain [12] constructed a resource allocation algorithm which provides small approximation factor (<= 2.0) as the number of servers increasing. Nguyen and Nam [13] discussed Performance Constraint and power aware Allocation for User Request in Virtual Computing. S M Johnson [14] proposed a task scheduling algorithm for two stage problem and Wing [15] has implemented Johnson Scheduling algorithm with less time complexity. Tsai, Fang and Chou [16] discussed Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Jaiganesh and Kumar [17,18] discussed Fuzzy based Data center load Optimization in Cloud computing and also Optimization of Cloud Resource Service Adaptability using Genetic Algorithms.

3 Model

In resource allocation of cloud computing each job requests several Virtual Machines, and each virtual machine requires underlying physical resources. Each job request requires a series of different virtual machine to complete its task. For example a Job request in Cloud Computing initially requires one platform (Windows) and later may work on another platform(Linux). The present model considers a job request that requires a series of two virtual machines instances of two types of virtual machines.

Virtual machine allocation for each job request in the proposed model has been described in the Fig. 1.

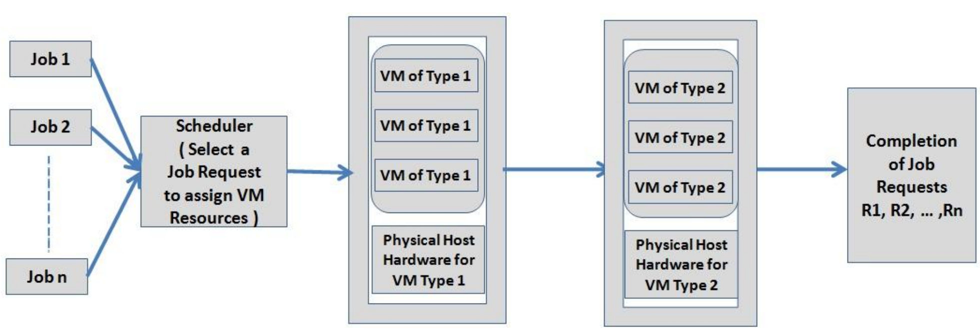

Fig. 1. Model of resource allocation

Let $r_1, r_2, \ldots, r_n$ be the set of job requests in the cloud computing at an instance of time. Each Job request r_i requires a Virtual Machine of type-1 (VM_1) for a time of T_{i1}, Virtual Machine of type-2 (VM_2) for an amount of T_{i2}. Each job request has to process its work on VM_1 and later on VM_2. Total elapsed time and average waiting time are the performance metrics for scheduling algorithm. An efficient scheduling algorithm is required to find a scheduling sequence among the n job requests on two virtual machines types to reduce Total Elapsed Time, Average Waiting Time and Average Turnaround Time. Idle time of Virtual Machines is also one of the performance evaluations.

Terminology and notations used in this paper is given below:

n : Number of jobs.
r_i : i^{th} Job Request.
T_{i1} : The time required on Virtual Machine of type-1 (VM_1) for job request r_i.
T_{i2} : The time required on Virtual Machine of type-2 (VM_2) for job request r_i.
p : Maximum no of virtual Machines can be created for a Virtual Machine Type.
s_i : Starting time of job request r_i in Virtual Machine of type-1.
c_i : Completion time of job request r_i in Virtual Machine of type-2.

4 Two Stage Scheduling in Cloud Computing

S M Johnson [14] proposed a scheduling algorithm for two stage problem. This scheduling algorithm seems to like the Shortest Job First scheduling but different in functionality. A typical two stage problem for the given n job request to process on Resource type-1 and then on Resource type-2, and assumed that only one instance of each Resource type are available. Wing [15] described an efficient implementation of Johnson algorithm. Johnson scheduling algorithm to schedule n job requests on two resource types can be expressed as follows.

Alg. Johnson (T[1..n][1..2])

```
begin
  i=1, j=n,
  Solution_seq = Empty;
  for each job request ri with shortest time period among all unprocessed jobs do
      if T[i][1] < T[i][2]  then
          add the job request ri to the Solution_seq at index i
          i = i + 1;
      else
          add the job request ri to the Solution_seq at index j
          j = j - 1;
      end if;
  end;
end;
```

Johnson Scheduling algorithm accepts n number of job requests, and each job request r_i to perform its operations on Resource type-1 for an amount of T_{i1} units time initially, later on Resource type-2 for an amount of T_{i2} units time. Solution vector is an array which stores the scheduling sequence of job requests. Johnson's algorithm recursively identifies a job request r_i with shortest time quantum T_{i1} or T_{i2} among unprocessed jobs. If the shortest time is T_{i1} of the job request r_i Resource type-1, then the job request r_i is added to the solution vector from front, otherwise if shortest time is T_{i2}, then the job is added in solution from the end. The job requests are added from front of solution vector since shortest time job on resource type-1 can be scheduled immediately, similarly the shortest time job on resource type-2 added at end.

Alg. Two-stage Scheduling (T[1..n][1..m], No_of_instances p)

```
1.  Begin
2.      Optimal_Seq = Johnson(T); // Calling Johnson Algorithm
3.      for i=1 to n do
4.          begin
5.          tval = i % p;
6.          append Optimal_Seq[i] to sub sequence S[tval];
7.      end;
8.     for i=1 to p do
9.     begin
10.        Calculate average waiting time, average turn around time and
11.        Total elapsed time for each subsequence S[i];
12.    end;
13. end;
```

Waiting Time of a job request is the time elapsed between the arrival time of job request and when the job request starts its work on Virtual Machine of type-1, plus the time elapsed between the time it completes its work on Virtual Machine of type-1 and starts its work on Virtual Machine of type-2. The Total Elapsed Time of the entire schedule is the time when all job requests completed their work on both virtual machines of type-1 and type-2 respectively. Total Elapsed time of this schedule is the c_k , where k is the last request in the schedule given by the scheduling algorithm. The performance metrics can be computed by the following computations for a given scheduling sequence.

Average Waiting Time(AWT), Average Turnaround Time(ATT), Total Elapsed Time (TET) of all job requests can be computed as follows

$$AWT = \sum_{i=1}^{n}\left((c_i - t_{i2}) - (s_i + t_{i1})\right) / n$$
$$ATT = \left(\sum_{i=1}^{n} c_i\right) / n$$
TET = c_k , where k is the last job request in schedule.

In Two-stage Scheduling algorithm p number of Virtual Machine Instances can be created for each virtual machine type on respective physical machines. Scheduling sequence can be divided in to p (p >=2) sub scheduling sequences. Each sub scheduling sequence can be processed with one instance of each virtual machine type. A scheduling sequence $S=\{r_1,r_2,r_3,\ldots.r_n\}$ can be divided into p sub sequences as $S_1=\{r_i$, for all i where i mod p = 1 $\}$, $S_2=\{r_i$, for all i where i mod p= 2 $\}$, …, $S_p=\{r_i$, for all i where i mod p = 0 $\}$ Each sub sequence has to schedule on a single VM instance among available multiple VM of each virtual machine type. For example the scheduling sequence which consists of 9 job request $\{r_1,r_2,r_3,r_4,r_5,r_6,r_7,r_8,r_9\}$ can be divided into 3 sub sequences as follows. $S_1=\{r_1,r_4,r_7\}$, $S_2=\{r_2,r_5,r_8\}$, $S_3=\{r_3,r_6,r_9\}$

5 Experimental Evaluation

A custom simulation environment has developed in JAVA to analyze the First Come First Server (FCFS) Scheduling and Two-stage Scheduling Algorithm with p instances for each virtual machine type for number of jobs 8 and 16. Gaussian distribution is used to generate process times for virtual machine for each job request. The following instance is considered where eight job requests on two virtual machine types such that only two(p=2) instances of virtual machine can be created for each virtual machine type. Each job request r_i and required process times on each type of virtual machines are shown in Table 1.

Table 1. Job requests and time required to process on VM_1 and VM_2

Job request (r_i)	Ti_1	Ti_2
1	25	65
2	13	37
3	82	50
4	7	65
5	10	17
6	87	99
7	33	83
8	57	54

The given job requests are divided into two sequences using FCFS's scheduling, and processed each sub sequence with an instance of each virtual machine type. The above sequence of eight job requests with two virtual machine types having two instances for each type, can be divided into two sub sequences S_1 = { 1 2 3 4 } and S_2 = { 5 6 7 8 }.

Initially, one can process r_0 on first instance of virtual machine type-1, r_4 on second instance of virtual machine type-1. Later each job request process it's operation on an instance of virtual machine type-2. Tables 2 and 3 show start time, turnaround time and waiting time of the given job request.

Table 2. FCFS Scheduling for sub sequence S_1

Job request (r_i)	Start time of r_i	Turn around time of r_i	Waiting time of r_i
1	*0*	*90*	*0*
2	*25*	*127*	*77*
3	*38*	*177*	*45*
4	*120*	*242*	*170*

Table 3. FCFS Scheduling for sub sequence S_2

Job request (r_i)	Start time of r_i	Turn around time of r_i	Waiting time of r_i
5	*0*	*27*	*0*
6	*10*	*196*	*10*
7	*97*	*279*	*163*
8	*130*	*333*	*222*

Table 4 shows the Total elapsed time, Average turnaround time and Average waiting time of FCFS Scheduling with two sub sequences. These performance evaluation metrics show the advantages of scheduling the job request with multiple instances of each virtual machine type.

Table 4. Performance Evaluation Metrics in FCFS scheduling with two Virtual Machine Instances for each Virtual Machine type

	S_1	S_2	Avg(S_1,S_2)
Total elapsed time	242	324	283
Average turnaround time	159	208	183
Average waiting time	73	99	86

Two-stage Scheduling for the given problem will results in a scheduling sequence {4 5 2 1 7 6 3 8}, such as the scheduling sequence is split into two sub sequences S_1 = {4 2 7 3} and S_2={5 1 6 8}. Table 5, shows the start time, turnaround time and waiting time of sub sequence S_1, and Table 6, shows the same metrics for sub sequence S_2.

Table 5. Two-stage scheduling for sub sequence S_1

Job request (r_i)	Start time of r_i	Turn around time of r_i	Waiting time of r_i
4	*0*	*72*	*0*
2	*7*	*109*	*59*
7	*20*	*192*	*76*
3	*53*	*242*	*110*

Table 6. Two-stage scheduling for sub sequence S_2

Job request (r_i)	Start time of r_i	Turn around time of r_i	Waiting time of r_i
5	*0*	*27*	*0*
1	*10*	*100*	*10*
6	*35*	*221*	*35*
8	*122*	*275*	*164*

Table 7, shows the Total elapsed time, Average turnaround time and Average waiting time of Two-stage Scheduling with two sub sequences. These performance evaluation metrics have minimum time compared to FCFS Scheduling with multiple instances.

Table 7. Performance Evaluation Metrics in Two-stage Scheduling with two Virtual Machine Instances for each Virtual Machine type

	S_1	S_2	Avg (S_1,S_2)
Total elapsed time	275	258	266.5
Average turnaround time	155	154	154.5
Average waiting time	61	52	56.5

Table 8, shows the three evaluation metrics of FCFS Scheduling, and Two-stage Scheduling. Total elapsed time in Extended-Scheduling with multiple VM instances is 218 which is 246.5 in FCFS Scheduling with multiple VM instances. Average waiting time of all job requests is 52 units of time in Two-stage Scheduling with multiple VM instances and 94.5 in FCFS Scheduling with multiple VM instances.

Table 8. Comparison of Performance Evaluation Metrics in FCFS and Two-stage Scheduling strategies (n=8)

	FCFS scheduling with multiple virtual machine instance (p=2)	Two-stage scheduling with multiple virtual machine instance (p=2)
Total elapsed time	283	266.5
Average turnaround time	183	154.5
Average waiting time	86	56.5

Fig. 2 depicts the comparison of FCFS and two stage scheduling algorithm with multiple VM instances with respect to the given three valuation metrics.

Similarly simulation has done for number of job 16(n=16), Tables 9 and 10, shows the simulation results.

Fig. 3 depicts the comparison of FCFS and Two stage scheduling algorithm with multiple VM instances with respect to the given three valuation metrics. Total Elapsed time has decreased by 20.09%, Average turnaround time decreased by 15.85%. and Average waiting time also decreased by 22.33%. Hence the two-stage scheduling algorithm gives better performance metrics when compared with FCFS for number of job request of 8 and also 16.

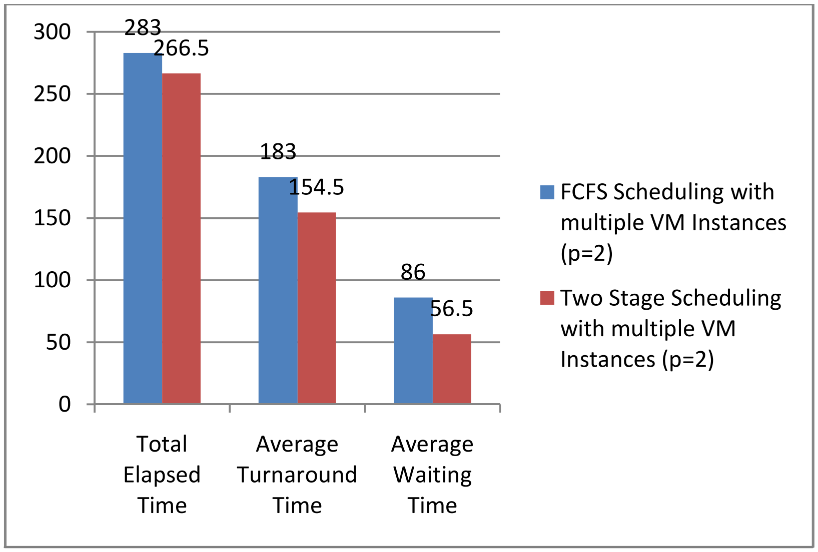

Fig. 2. Comparison of performance evaluation metrics in different scheduling strategies (n=8).

Table 9. Job Requests and Time required to process on VM_1 and VM_2

Job request (r_i)	**Ti_1**	**Ti_2**
1	59	39
2	19	76
3	44	96
4	14	0
5	35	99
6	84	60
7	80	15
8	2	17
9	*86*	*4*
10	*26*	*27*
11	*48*	*31*
12	*4*	*77*
13	*39*	*5*
14	*43*	*3*
15	*41*	*36*
16	*25*	*52*

Table 10. Comparison of performance evaluation metrics in FCFS and two-stage scheduling strategies (n=8)

	FCFS scheduling with multiple virtual machine instance (p=2)	**Two-stage scheduling with multiple virtual machine instance (p=2)**
Total elapsed time	418	333.5
Average turnaround time	277.5	233.5
Average waiting time	197	153

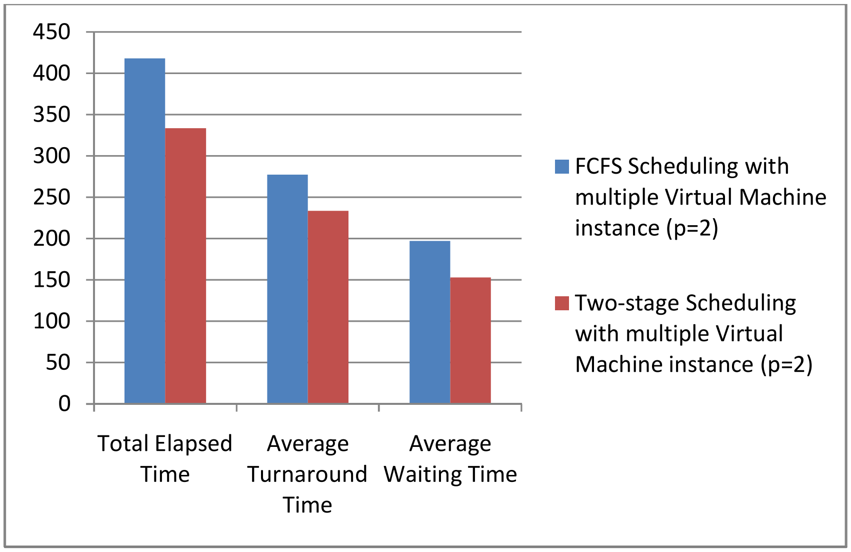

Fig. 3. Comparison of performance evaluation metrics in different scheduling strategies (n=8)

6 Conclusion

Scheduling the given n job requests in cloud computing on two types of virtual machines using Two-stage Scheduling algorithm gives better performance evaluation metrics. The total elapsed time and Average waiting time are reasonably decreased in Two-stage Scheduling Algorithm when compared to FCFS Scheduling algorithm. Experimental results show that Two-stage Scheduling algorithm has better scheduling parameters.

Competing Interests

Authors have declared that no competing interests exist.

References

[1] Samuel Gorenstein. An algorithm for project (Job) sequencing with resource constraints. Journal of Operations Research. 1970;20:835-850.

[2] Richard J. Giglio, Harvey M. Wagner. Approximate solutions to the three machine scheduling problem. Journal of Operations Research. 1964;12:305-324.

[3] Edward Ignall, Linus Schrage. Application of the branch and bound technique to some flow-shop scheduling problems. Cornell University; 1964.

[4] Jia Yu, Rajkumar Buyya. A budget constrained scheduling of workflow applications on utility grids using genetic algorithms; 2006.

[5] Jia Yu, Rajkumar Buyya, Chen Khong Tham. Cost-based scheduling of scientific workflow applications on utility grids; 2005.

[6] Hamid Mi Fard, Radu, Thomas. A truthful dynamic workflow scheduling mechanism for commercial multi-cloud environments. IEEE T Transaction on Parallel and Distributed Systems; 2012.

[7] Quyet Thang Nguyen, Quang-Hung, Nguyen Huynh Tuong, Van Hoai Tran, Nam Thoai. virtual machine allocation in cloud computing for minimizing total execution time on each machine, IEEE; 2013.

[8] Xiaocheng Liu, Chen Wang. Priority-based consolidation of parallel workloads in the cloud. IEEE Transactions; 2012.

[9] Zhen Xiao, Weijia Song, Qi Chen. Dynamic resource allocation using virtual machines for cloud computing environment. IEEE Transaction on Parallel and Distributed Systems; 2012.

[10] Ilavarasan E, Thambidurai P. Low complexity performance effective task scheduling algorithm for heterogeneous computing environments; 2007.

[11] Mohsen Amini, Rajkumar Buyya. Adapting market-oriented scheduling policies for cloud computing; 2009.

[12] Navendu Jain, Ishai Menache, Joseph Naor, Jonathan Yaniv. A truthful mechanism for value-based scheduling in cloud computing; 2010.

[13] Nguyen Nam. Performance constraint and power aware allocation for user request in virtual computing lab; 2011.

[14] Johnson SM. Optimal 2 and 3 stage production schedules with set up time included. Noval Research Logistics, Q1. 1954;61-88.

[15] Wing S. An efficent implementation of a two stage production scheduling algorithm. Journal of Operational Research. 1989;40 (11):1049-1052.

[16] Jinn-Tsong Tsai, Jia-Cen Fang, Jyh- Horng Chou. Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Journal Computers and Operations Research. 2013;40:3045-3055.

[17] Jaiganesh M, Vincent Antony Kumar. B3: Fuzzy based data center load optimization in cloud computing. Mathematical Problems in Engineering. 2013;1-11.

[18] Jaiganesh M, Vincent Antony Kumar. A optimization of cloud resource service adaptability using genetic algorithms. International Conference on IT and Intelligent Systems. 2013;66-70.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=733&id=6&aid=7611