Minimum Makespan Task Scheduling Algorithm in Cloud Computing

N. Sasikaladevi

*School of Computing, SASTRA University,
Tamilnadu, India
sasi@cse.sastra.edu*

Abstract

Cloud computing provides powerful and on-demand business environment. It is built on top of virtualized data centers. Virtualization provides the flexible infrastructure for cloud. But, managing the resources and scheduling the tasks in virtualized data center is a challenging task. This paper proposes a minimum makespan task scheduling framework named MMSF and minimum makespan task scheduling algorithm named as MMA. This algorithm is developed with two objectives. First, minimizing the total makespan and maximizing the virtual machine utilization. The task scheduling problem is formulated as multi-objective optimization problem. It is solved by using optimization techniques. Experiments shows that MMA outperforms the traditional task scheduling algorithms based on total makespan and virtual machine utilization.

Keywords: Cloud computing, Virtualization, Task Scheduling, Multi objective optimization

1. Introduction

Cloud is defined as "A large-scale distributed computing paradigm that is driven by economies of scale, in which a pool of abstract virtualized, dynamically-scalable, managed computing power, storage, platforms, and services are delivered on demand to external customers over the Internet" [1]. Data centers are getting popular in every enterprise [2-4]. Data centers are developed using virtualization technology with virtual machines called processing component of cloud [5-7]. Virtualization provides new challenges in cloud computing such as resource management and task scheduling [8, 9]. VMs share the physical resources such as processor, memory, etc. Effective use of VM makes efficient utilization of physical resources. Cloud computing research focuses on resource provisioning and VM allocation [6- 11]. All these work are used to increases the performance of data centers. Task level scheduling algorithms are needed to efficiently allocate the tasks to VM. This paper focuses on task scheduling algorithm with two objectives-minimum makespan and maximum utilization of available VMs.

The rest of the paper contains the related work on task scheduling in cloud computing environment in section 2. Section 3 describes the Minimum makespan scheduling framework (MMSF). Section 4 describes the minimum makespan scheduling algorithm (MMA). Section 5 gives the experimental results and section 6 ended with conclusion.

2. Related Work

Task scheduling in cloud computing is a emerging research trends. Most of the research work in task scheduling adopt different strategies like deadline based; maximize the resource utilization; workflow based model and so on. Different task scheduling algorithms for cloud computing are proposed by various researchers.

Thamari Selvi *et.al* [12] proposed a framework called CARE Resource Broker CRB).

CRB is a protocol to create, manage and allocate virtual resources in cloud environment. Maximum scheduling rate with CRB is 80%. Sen Su *et.al* [13] proposed a method to schedule large problem in cloud. Large program is divided into dependent task. The sequences of tasks are connected by DAG tree. This task scheduling algorithm focuses only on minimizing the makespan.

Young Choon Lee *et.al* [14] proposed profit driven scheduling for cloud services. Profit refers to service quality. Jiayin Li et.al[15] proposed optimization algorithm for scheduling of parallel tasks in cloud system to improve the performance. Parallel processing is applied in cloud computing. It considers the preemptable tasks. Baomin Xu *et.al* [16] introduced Berger model for job scheduling in cloud. It uses dual fairness constraint. The first constraint is to segregate the user task based on QoS and the second constraint is defining the fail utilization of resources.

Xiangzhen Kong *et.al* [17] proposed fuzzy prediction based task scheduling for cloud. It is a two objective optimization problem. The objectives are availability and responsiveness of the resources. Fuzzy prediction is used to model the uncertain workload. It provides the resource utilization rate 85% to 100%. Kai Li *et.al* [18] proposed a task allocation schema in cloud environment. It gives the solution for task scheduling by adjusting entropy function.

Amit *et.al* [19] proposed prioriy based task scheduling algorithm. Tasks are prioritized based on the size and VM are priorityized based on MIPS. Then FCFS scheduling is used to allocate the tasks to the VMs. Sudha *et.al* [20] proposed a two level scheduler for cloud computing environment. QoS based meta scheduler and backfill strategy based VM schedulers are proposed. Matthias *et.al* [21] proposed a task distribution method for virtual machines. Jiayin Li *et.al* [22] proposed Feedback based scheduling algorithm for preemptable jobs in cloud. MinMin algorithm is used to schedule the tasks.

Jiaylin Li *et.al* [23] proposed adaptive resource allocation technique for preemptable jobs. Jinhua Hu *et.al* [24] proposed a load balancing stategy for virual machines. Zhongni Zheng et.al[25] proposed gentic based cloud scheduling allproach. Load balancing algorithm for cloud data ener is proposed by Wenhong Tian *et.al* [26]. Job scheduling system for cloud is proposed by Kushal Dutta et.al[27].

Cloud virtual machine scheduling is proposed by Stelios Sotiriadis *et.al* [28]. Dynamic resource allocation strategy is proposed by Zhen xiao *et.al* [29]. A market oriented scheduling polices are proposed by Rajkumar buyya *et.al* [30]. In this model, scheduling policy considers user budget level and task dead line. Load balancing based task scheduling is proposed by Yiqiu Fang *et.al* [31]. It is based on the two level scheduling model. All these research work provides different solutions for task scheduling in cloud computing. In this paper, a multi objective scheduling algorithm is proposed. The objectives are minimizing the makespan and maximizing the VM Utilization rate.

3. Problem Statement

In cloud computing paradigm, scheduling the tasks with minimum makespan and minimum waiting time is considered as an optimization problem. In is very much needed to consider both makespan and resource (VM) utilization in order to define the task scheduling problem. Makespan is the total time taken by the resources to complete the executing of all tasks. Utilization of VM is defined as how well the resources are used in the cloud. Normally, Makespan is inversely proposal to Utilization rate. The efficient scheduling algorithm should schedule the tasks in such a a manner that makespan is minimum and VM utilization is maximum. In this paper, task scheduling is considered as a multi object optimization problem, and it is solved by using optimization technique.

Task scheduling problem is defined as a bi-criteria scheduling problem. Two objectives are minimizing the makespan and VM utilization rate. Tasks are considered as pre-emptable tasks. There is a need to reduce the number of pre-emption also. Multi objective

on multi machine scheduling is NP-hard problem. There is a need to use optimization techniques to solve the multi-objective scheduling.

Consider the multi objective scheduling problem,

- Let $M = \{1,2,...m\}$ where, m is the number of VMs
- Let $N = \{1,2,...n\}$ where, n is the number of tasks and n is an integer multiple of m. If this condition is not satisfied enough tasks are added with zero processing time.
- Let $P = \{p_1, p_2 ... p_n\}$ where, p_i is the processing time of ith task. It includes the initial set up time and the processing time, and that $p_1 \leq p_2 \leq \cdots \leq p_n$.

Each VM can process one task at a time. No two VM process the same task at a time. Tasks are considered as pre-emptable. All the tasks are having equal importance. The aim is to minimize the makespan and increasing the utilization rate of VMs.

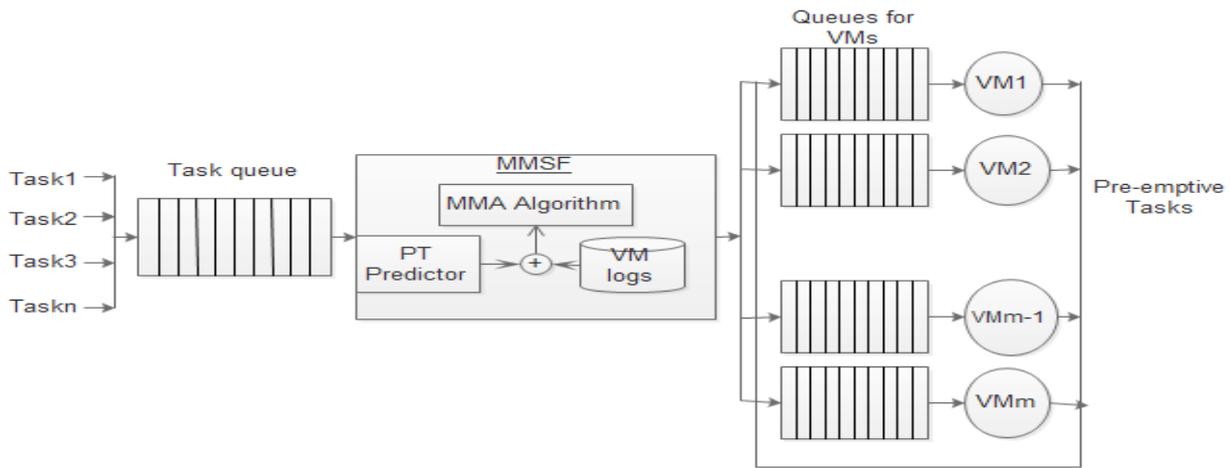## 4. Minimum Makespan Scheduling Framework

Cloud Computing provides services based on on-demand basis. Data centers utilization percentage is only between 5% to 20% [31,32]. Main issue in cloud computing is the under utilization of resources. Hence, maximizing the unitization of resources at same time minimizing the makespan is a challenging task. Resource management and task scheduling in cloud computing is different from the traditional computing paradigm. Scheduling can be centralized or distributed. In the centralized scheduling, the objective of the scheduling is to increase the overall performance of the system. Whereas distributed scheduler aims to increase the performance end user. Cloud computing needs centralized as well as distributed scheduler which makes the scheduling in cloud computing is complex.

Datacenters consists of several physical machines that may be homogenous or heterogeneous. Upon receiving the task request from users, the tasks are assigned to physical machine. In cloud computing, virtualization is an important technique. VMs are created dynamically by allocating one or more physical machines.

When users submit the task, the cloud should allocate the VMs to the task. Because of the elastic nature of the cloud, consumers tasks are allocated based on First Come First Serve (FCFS) method or Shortest Processing time First (SPF) method [34].

In this paper, the minimum makespan scheduling framework is proposed and it is shown in Figure 1. In this framework, upon receiving the tasks from different user, the tasks are kept in the task queue. In this framework, the tasks are considered as an equal priority and pre-emptable independent tasks. Number of tasks are n, number of VMs are m. 'n' tasks are scheduled on 'm' VMs with minimum makespan and minimum resource utilization rate.

MMAF is a centralized scheduler. It maintains the VMs detail. In this case, the VMs are considered to have equal number of physical machine. The physical machines are homogeneous. Hence, the VMs are the homogenous set of machine. MMA algorithm is proposed to schedule the tasks on set of VMs. MMA are a multi objective scheduling algorithm. Processing time of each task is predicted before applying the MMA scheduling algorithm.

Figure 1. Minimum Makespan Scheduling Framework

5. MMA Algorithm

The task Scheduling problem is considered as the Multi objectives optimization problem [36]. Bi-criteria scheduling algorithm proposed by Leung [37] is adopted in this algorithm. Objectives are minimizing the makespan and increasing the VM utilization rate. This is the NP-hard problem. In order to solve this problem, this is reduced by using Even-Odd parity technique.

Definition: Even-Odd Partition [35]

Let consider a finite set I and 2n integers aj, such that

$$\sum_{j=1}^{k} a_j = 2E$$

I is portioned into two set I_1 and I_2 such that

$$\sum_{a_j \in I_1} a_j = \sum_{a_j \in I_2} a_j = E$$

Where I1 and I2 each contains exactly one element form $\{a_{2i-1}, a_{2i}\}$?

MMA algorithm is a five step process. Following assumptions are made before applying this algorithm.

- $M = \{1,2,..m\}$ where, m is the number of VMs
- $N = \{1,2,...n\}$ where, n is the number of tasks and n is an integer multiple of m. If this condition is not satisfied enough tasks are added with zero processing time. n=r*m with r∈N
- $P = \{p_1, p_2 ... p_n\}$ where, p_i is the processing time of ith task. It includes the initial set up time and the processing time, and that $p_1 \leq p_2 \leq \cdots \leq p_n$.

Each VM can process one task at a time. No two VM process the same task at a time. Tasks are considered as pre-emptable. All the tasks are having equal importance.

Basic steps involved in MMA are,

Step1 : Arrange the tasks based on the non-deceasing order of processing time

Step 2: Allocate (r-1)*m tasks to m VMs based on Shortest Processing time First (SPT) algorithm.

Step 3: Calculate the last task completion time on VM j as

$$TC_j^M = \sum_{k=1}^{r-1} p_{j+(k-1)*m} \ \forall \ j = 1, \ldots m$$

Step 4: Calculate the Optimal Makespan[37]

$$MS_{opt} = MAX_{k=1,2,\ldots m} \sum_{l=1}^{k} TC_l^M + p_{n-l+1} \ /k$$

Step 5: Remaining m tasks are scheduled based on the following rule.
a) If the task completely fit onto the VM with least capacity, then it is scheduled on that machine.
b) If the task fit onto the VM with its capacity match with the MSopt, then schedule the task on that VM and don't consider that VM for further task allocation
c) If the above two conditions are false, then it is not possible to allocate the task completely onto the single VM. The task need to be preempted. Find a VMs such that
$$MS_{opt} - TC_k > p_j > MS_{opt} - TC_l$$
Then assign task j to VMk for $MS_{opt} - TC_k$ unit time and VMl for pj-$MS_{opt} + TC_k$ unit times.

Available VMs are allocated with tasks for MS_{opt} time units. A MMA algorithm is used to utilize all the VMs effectively. All the VMs are active and tasks are executed by m VMs in parallel. This algorithm reduces the total makespan by effectively allocating tasks to all VMs.

The time complexity of MMA algorithm is $O(n \log(n))$ times. The complete psedueo code of MMA is given in listing 1.

Listing 1: Pseudo code of MMA
Input:
n-number of tasks, m- number of VMs
P-Vector contains the processing time of n tasks.
Output: Optimized Schedule S
Procedure:

```
Arrange the task in the non-decreasing order of their processing time.
Set r=n/m, k=r-1
//Scheduling of first k tasks.
Set t=0;//Initialize the number of task scheduled
While (k>0)
{
        For( i=1;i<m;i++)
                Assign task t_{i+t} to VM_j, if the VM_j is busy, add t_{i+t} into the waiting queue for
VM_j
        k--;t=t+m;
}
//Minimum Makespan calculation
For (j=1;j<=m;j++)
{
        TC_j=0;//Initialize the last task completion in VM_j
        For(l=1;l<=(r-1);i++) TC_j=p_{j+(l-1)*m}
}
```

Set MS_{opt} =0; //initialize the minimum makespan
For(i=1;i<=m;i++)
        For(l=1;l<=k;l++) k+=c_l+p_{n-l+1},k=k/I; MS_{opt}=max(k, MS_{opt})
//Allocation of remaining r tasks.
For(k=n;k>=n-m+1;k++)
{
        // scheduling of task t_k
        For(j=1;j<=m;j++) If TC_j<MS_{opt} K=K+nk; nk++
        Set j1=1;
        For(i=2;i<nk;i++) If TC_{j1}<TC_j then J1=i
        If p_k<=MS_{opt}-TC_{j1} then
                Task t_k is completely scheduled on VM_{j1}. If the VM_{j1} is busy, add the t_k at the end of waiting queue for VM_{j1}
                TC_{j1}=TC_{j1}+p_k //Update the last task completion time of VM_{j1}
        Else
                //Preemption is needed. Selecting two VMs to schedule the task partially
                For(l=1;l<=j1-1;l++) If c_l+p_k==MS_{opt}then c1++; L=L+l;
                If(c1!=0)
                        j2=$L_{(c1--)}$
                        Assign task t_k to VM_{j2} If the VM_{j2} is busy, add the t_k at the end of waiting queue for VM_{j2}
                        TC_{j2}=TC_{j2}+p_k //Update the last task completion time of VM_{j2}
                Else
                For(i=1;i<=m;i++)
                        IfMS_{opt}-TC_i>p_k and MS_{opt}-TC_i>>0 then J2=I; flag$_{j2}$=1
                        If p_k<MS_{opt}-TC_i> && MS_{opt}-TC_i>>0 && p_k>0 then J3=I; flag$_{j3}$=1;
                        If(flag$_{j2}$!=0 && flag$_{j3}$!=0)
                                Allocate task t_k to VM_{j2} for MS_{optx}-TC_{j2} time units
                                TC_{j2}=MS_{opt}
        Allocate preempted task t_k to VM_{j3} for p_k-c_{max}+TC_{j2}> time units
                        TC_{j3} =p_k-MS_{optx}+TC_{j2}
}

6. Experiment on MMA

In order to evaluate MMA algorithm, simulation environment is created. A simulation tool named CloudSchedule in Java. Data centers are created with two hosts. Equal number of physical machines (Cores) is assigned to VM. It is assumed that the task arrival is based on Poisson distribution and processing time of task conforms to normal distribution. VM are chosen to represent the virtual processing element for real world cloud system.

In order to evaluate MMA, several tests are conducted with different sets of tasks and VMs. This algorithm is tested based on two conditions, one is normal load and the other one is heavy load. In the cloud computing paradigm, the service requires are not equal at all time. It is busty in nature.

MMA algorithm proved to be suitable in all conditions. Figure 2a shows the maximum makespan in normal load. Figure 2b shows the maximum makespan in heavy load. nT refers to number of tasks and nVM refers to number of allocated VMs to schedule the tasks. MMA is tested with varying nT/nVM ratios. Figure 2 shows that minimum makespan is achieved with the ratio of 5. Makespan is highly depends on ratio between nT and nVM.

MMA algorithm is compared with FCFS and SPF scheduling algorithms. Figure 3 shows that MMA provides better makespan as compared to FCFS and SPF. And it improves the VM utilization rate. It also shows that MMA outperforms FCFS and SPF in

normal load and heavy load conditions.

MMA provides the better resource utilization rate as compared to Xiangzhen [17] model. VM utilization achieved in Xiangzhen model is in the range 85% to 100%. MMA provides 96% to 100% VM utilization rate. As compared to Yiqiu Fang [31] model, MMA provides better resource utilization. Yiqiu Fang model provides the resource utilization in the range 60% to 65% where as MMA provides 96% to 100% resource utilization rate.
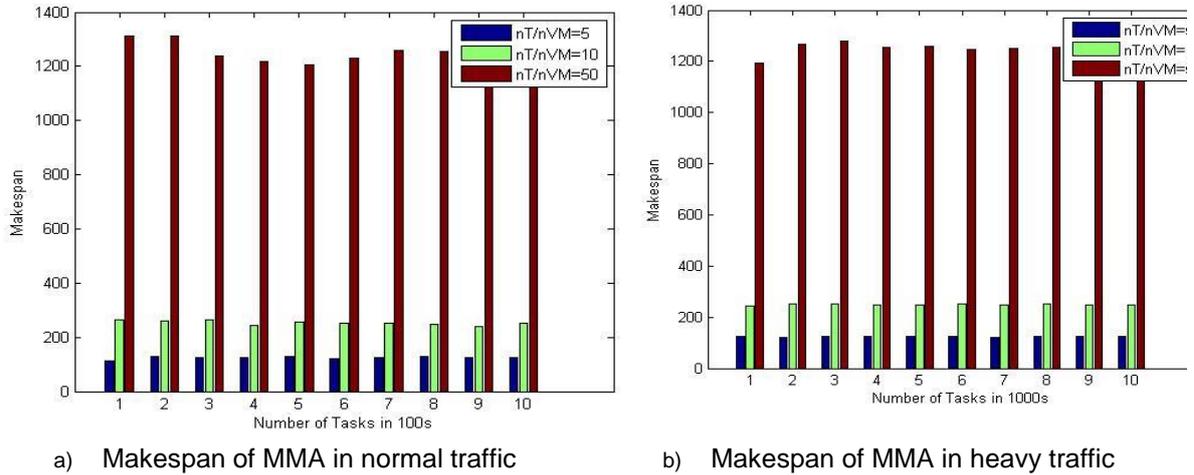

a)   Makespan of MMA in normal traffic   b)   Makespan of MMA in heavy traffic

Figure 2. Analysis of MMA on Different Traffic

Case 1: Task arrival rate is high (Number of tasks/number of VM =5)

a)   VM utilization of MMA, FCFS and SPF   b)   Makespan of MMA, FCFS and SPF
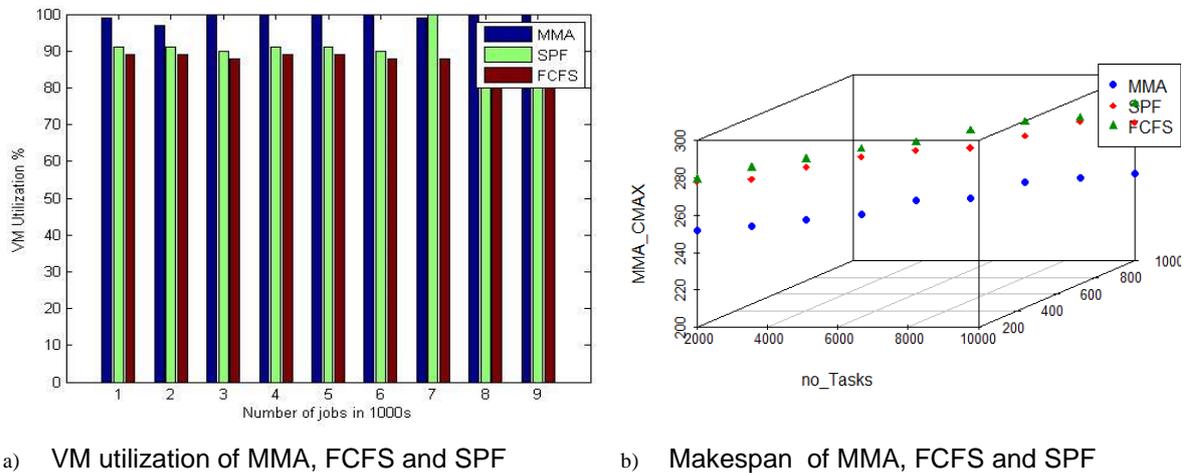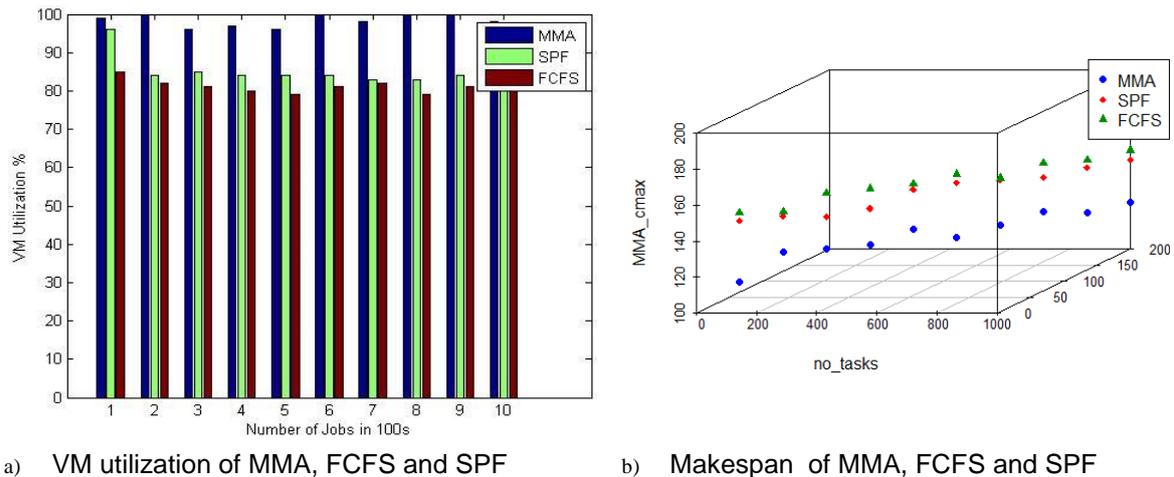
Figure 3. Comparison of MMA with FCFS and SPF based on total VM Utilization rate and Makespan in Heavy load

Case 2: Task arrival rate is normal (Number of tasks/number of VM =5)

a)   VM utilization of MMA, FCFS and SPF   b)   Makespan of MMA, FCFS and SPF

Figure 4.Comparison of MMA with FCFS and SPF based on Makespan and VM Utilization Rate in Normal Load

7. Conclusion

In this paper, Minimum makespan and maximum VM utilization task scheduling algorithm is developed. It is observed that proposed MMA outperforms the FCFS and SPF based on makspan and Utilization rate. One important overhead with MMA is the number of preemptions. In order to achieve 100% VM utilization, there is need to tolerate 2% of preemptions on the total submitted job. It is also observed that, MMA is a bi-criteria scheduling problem. In future, MMA can be improved further by considering other objectives related to cloud computing such as SLA based scheduling and energy efficient scheduling.

References

[1]   Foster, I., Zhao, Y., Raicu, I., and Lu, S, "Cloud computing and grid computing 360-degree compared", *In the Proceedings of the Grid Computing Environments Workshop*, pp. 1-10,2008.
[2]   Joseph D, Tavakoli A,StoicaI, "Apolicy-aware switching layer for data centers", *ACM SIGCOMM computer Communication Review,* Vol. 38, Issue. 4, pp. 51-62, 2008
[3]   Arregoces M, Portolani M, *Data center fundamentals*, Cisco Press, 2003.
[4]   Snevely R, *Enterprise data center design and methodology,* Prentice-Hall: Sun Microsystems, Inc.,2002.
[5]   Graupner S, Kotov V, Trinks H, "Resource-sharing and service deployment in virtual data centers*", In the Proceedings of the 22nd international conference on distributed computing systems*, pp.66-74, 2002.
[6]   Xu J, Zhao M, Fortes J, Carpenter R, Yousif M, "On the use of fuzzy modeling in virtualized data center management", *In the Proceedings of the 4th international conference on autonomic computing*, 2007.
[7]   Zhu X, "1000 Islands: an integrated approach to resource management for virtualized datacenters", *Cluster Computing,* Vol. 12, Issue 1, 2009
[8]   Marty M, Hill M, "Virtual hierarchies to support server consolidation*", In the Proceedings of the 34th annual international symposium on computer architecture*, pp.46-56, 2007.
[9]   Tinghuai Ma, Ya Chu, Licheng Zhao and Otgonbayar Ankhbayar, "Resource Allocation and Scheduling in Cloud Computing: Policy and Algorithm",IETE Technical Review, Vol. 31, Issue 1, pp 4-16, 2014.
[10]   Jie Song, Xuebing Liu, Zhiliang Zhu, Dazhe Zhao and Ge Yu,"A Novel Task Scheduling Approach for Reducing Energy Consumption of MapReduce Cluster", Vol. 31, Issue 1, pp.65-74, 2014.
[11]   Song S, Hwang K, Kwok Y, "Risk-resilient heuristics and genetic algorithms for security assured grid job scheduling", *IEEE Transactions on Computers,* Vol. 55, Issue 6, pp.703-19, 2006.
[12]   Thamarai Selvi Somasundaram, Balachandar R. Amarnath, R. Kumar, P. Balakrishnan, K. Rajendar, R. Rajiv, G. Kannan, G. Rajesh Britto, E. Mahendran, B. Madusudhanan, "CARE Resource Broker: A framework for scheduling and supporting virtual resource management*", Future Generation Computer Systems*, Vol. 26, , pp. 337-347, 2011.

[13] Sen Su, Jian Li, Qingjia Huang, Xiao Huang, Kai Shuang, Jie Wang, "Cost-efficient task scheduling for executing large programs in the cloud", *Parallel Computing* , Vol. 39, pp.177–188, 2013.

[14] Young Choon Lee, Chen Wang, Albert Y. Zomayaa, Bing Bing Zhoua, "Profit-driven scheduling for cloud services with data access awareness", *Journal on Parallel Distributed Computing*, Vol.72, pp.591–602,2012.

[15] Jiayin Li, Meikang Qiu, Zhong Ming, Gang Quanc, Xiao Qin, Zonghua Gu, "Online optimization for scheduling preemptable tasks on IaaS cloud systems", *Journal on Parallel Distributed Computing*, Vol.72, pp. 666–677, 2012,

[16] Baomin Xu, Chunyan Zhao, Enzhao Hu, Bin Hu, "Job scheduling algorithm based on Berger model in cloud environment", *Advances in Engineering Software*, Vol. 42, pp. 419–425, 2011.

[17] Xiangzhen Kong, Chuang Lin, Yixin Jiang, WeiYan, Xiaowen Chu, "Efficient dynamic task scheduling in virtualized datacenters with fuzzy prediction", *Journal of Network and Computer Applications*, Vol. 34, pp. 1068–1077, 2011.

[18] Kai Li, Yong Wang, Meilin Liu, "A Task Allocation Schema Based on Response Time Optimization in Cloud Computing"

[19] Amit Agarwal, Saloni Jain, "Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment", *International Journal of Computer Trends and Technology*, Vol. 9, pp. 344-349, 2014.

[20] Sudha Sadhasivam, Jayarani, Nagaveni, Vasanth Ram, "Design and Implementation of an efficient Two-level Scheduler for Cloud Computing Environment", *In the Proceeding of 2009 International Conference on Advances in Recent Technologies in Communication and Computing*, pp.884-886, 2009

[21] Matthias Schmidt, Niels Fallenbeck, Matthew Smith, Bernd Freisleben, "Efficient Distribution of Virtual Machines for Cloud Computing*", In the Proceeding of 18th Euromicro Conference on Parallel, Distributed and Network-based Processing*, pp.567-574, 2010.

[22] Jiayin Li Meikang, Qiu Jianwei Niu, Wenzhong Gao, Ziliang Zong Xiao Qin," Feedback Dynamic Algorithms for Preemptable Job Scheduling in Cloud Systems", *In the Proceeding of 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, pp.561-564, 2010

[23] Jiayin Li, Meikang Qiu, "Adaptive Resource Allocation for Preemptable Jobs in Cloud Systems", *IEEE Computer Society*, pp.31-37, 2010.

[24] Jinhua Hu, Jianhua Gu, "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment", *In the Proceeding of 3rd International Symposium on Parallel Architectures, Algorithms and Programming*, pp.89-96, 2010.

[25] Zhongni Zheng ,Rui Wang, Hai Zhong, Xuejie Zhang, "An Approach for Cloud Resource Scheduling Based on Parallel Genetic Algorithm", IEEE 2011, pp.444-447, 2011

[26] Wenhong Tian, Yong Zhao, Yuanliang Zhong, Minxian Xu, Chen Jing, "A Dynamic and integrated load balancing scheduling algorithm for cloud data centers", *In the Proceedings of IEEE CCIS*, pp.311-316, 2011.

[27] Kushal Dutta, Ramendu Bikash Guin, Sourav Banerjee, Sayan Chakrabarti, Utpal Biswas, "A Smart Job Scheduling System for Cloud computing Service Providers and Users: Modeling and Simulation", *In the Proceedings of 1st International conference on Recent Advances in Information Technology* , 2012.

[28] Stelios Sotiriadis, Nik Bessis, Fatos Xhafa, Nick Antonopoulos, "Cloud virtual machine scheduling: Modelling the cloud virtual machine instantiation", *In the Proceedings of 2012 Sixth International Conference on Complex, Intelligent, and Software Intensive Systems*, pp.233-240, *2012*

[29] Zhen Xiao, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment", IEEE Transactions on distributed and parallel systems, Vol. 24, Issue 6, 2013.

[30] Mohsen Amini Salehi and Rajkumar Buyya, "Adapting Market-Oriented Scheduling Policies for Cloud Computing*", Springer LNCS 6081*, pp. 351–362, 2010

[31] Yiqiu Fang, Fei Wang, and Junwei Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing*", In the Proceedings of WISM 2010*, LNCS 6318, pp. 271–277, 2010

[32] Siegele L, "Let it rise: A Special report on Corporate IT",2008.

[33] Rangan, Cookem Dhruv, "The Cloud Wars: 100 billion dollars at stake", Merrill Lynch, Tech. Rep., 2008.

[34] Morariu, Borangiu,"A genetic algorithm for workload scheduling in cloud based e-learning", In the proceedings of Internation workshop on cloud computing, pp.1-6, 2012

[35] Alessandro Agnetis, Jean charled, "*Multiagent scheduling: Models and Algorihtms*", Springer, 2014.

[36] Vincent T kindt, Jean Charles Billaut, *Multi-criteria scheduling algorithms: Theory, Applications and algorithms*, Springer, 2006

[37] Leung, Young, "Minimizing schedule length subject ot minimum flow time", *SIAM Journal on computing*, Vol. 18, Issue. 2, pp.314-326, 2010.

Author

N. Sasikaladevi completed Doctoral degree in computer science. She has published more than 20 papers in peer-reviewed international journals and IEEE conference proceedings. She authored two books titled "Object oriented Programming in C#.NET", "Programming in C#.NET" published by Prentice Hall. Her research interest is in Services computing; Cloud computing and Big Data analytics.