Time and Resource Efficient Task Scheduling in Cloud Computing Environment

Vrushali Desale[1]

[1]*Department of Computer Engineering, Dr. Babasaheb Ambedkar Technical University, Raigad*

Abstract - *Cloud computing has become a new age technology that has got huge potentials in enterprises and markets. Cloud Computing has Large Scale Distributed Infrastructure which is accessible and scalable infrastructure. Cloud computing provides a pay as you go model in which the user has to pay for the services he uses. However one of the major challenges in cloud computing is related to optimizing the resources being allocated. Because of the uniqueness of the model, resource allocation should be performed with the objective of minimizing the costs associated with it. This optimized use of cloud can only be done by efficient and effective algorithm to select the best resources. In this paper, the Task Based allocation of resources is used to minimize the makespan of the cloud system and also to increase the resource utilization. The simulation is done using CloudSim and results show that TBA algorithm reduces the makespan, execution time and cost as compared to Random Algorithm and FCFS algorithm.*

Key Words: **Cloud Computing, Scheduling, Resource Allocation, Task Allocation, Make Span, Execution Time, TBA Algorithm, PM, VM.**

1. INTRODUCTION

Cloud computing emerges as a new computing paradigm which aims to provide reliable, customized and QoS (Quality of Service) guaranteed computing dynamic environments for end users. Distributed processing, parallel processing and grid computing together emerged as cloud computing [2]. A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and represented as one or more unified computing resources based on service level agreements established through the negotiation between the service providers and consumers. Cloud computing is a term that involves delivering hosted services over the Internet. By using the virtualization concept, cloud computing can also support heterogeneous resources and flexibility is achieved. Another important advantage of cloud computing is its scalability. Cloud computing has been under growing spotlight as a possible solution for providing a flexible on demand computing infrastructure for a number of applications. The services are broadly divided into three categories Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [3].

One of the goals in Cloud Computing Environment is to use the resources efficiently and gain maximum profit. Task Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users. So scheduling is the major issue in establishing Cloud computing systems. Job Scheduling of cloud computing refers to dispatch the computing tasks to resource pooling between different resource users according to certain rules of resource use under a given cloud circumstances. Resource management and job scheduling are the key technologies of cloud computing that plays a vital role in an efficient cloud resource management [4]. In cloud environment, huge number of tasks is executed simultaneously; an effective Task Scheduling is required to gain better performance of the cloud system. Various Cloud-based Task Scheduling algorithms are available that schedule the user's task to resources for execution. Due to the novelty of Cloud Computing, traditional scheduling algorithms cannot satisfy the cloud's needs, the researchers are trying to modify traditional algorithms that can fulfil the cloud requirements like rapid elasticity, resource pooling and on-demand self-service[5].

The scheduling in context of cloud means choose the best suitable resource for task execution or to allocate machines to tasks in such a way that the completion time (makespan) is minimized. Generally, in scheduling algorithms list of tasks is constructed by giving priority toevery task. Tasks are choosed according to priorities and assigned to a processor which fulfill a predefined objective function. Task allocation as shown in Fig.1 of user requests to the cloud resource can optimize various parameters like energy consumption, makespan, throughput, etc. Makespan is the required time to complete all tasks. This task allocation or mapping problem is a well-known NP-Hard optimization problem. The purpose of task allocation is to minimize the makespan of the cloud system, minimize the cost and also to increase the resource utilization [6].

2. RELATED WORK

Task scheduling is to choose the best suitable resource for task execution. It also involves allocating the tasks to the machines in such a way that the completion time (makespan) is minimized. In [7], authors mainly focus on preamble task. The author proposed an adaptive resource allocation algorithm which adjusts the cloud resources adaptively on the basis of actual task executions update. For this, they are using two algorithms: ALS (Adaptive List Scheduling) and AMMS (Adaptive Min- Min Scheduling) algorithm which uses static task scheduling paradigm to allocate the cloud resources statically. The online adaptive procedure is also used for re-evaluating the remaining span for execution and updates the task execution list. In [8], dynamic resource allocation on various distributed multiple criteria is used. The author proposed a PROMETHEE method that is used to take the follow decisions:

(1) VM placement: Aim is to find the best suitable Physical machine (PM) which is capable of hosting particular Virtual machine can run without interfering, try to assign that VM to PM.

(2) Monitoring: In this phase algorithm, has to monitor total utilization of resources by hosted VM.

(3) VM Selection: if the current physical machine is not capable of running particular VM then try to migrate that VM to some another Physical machine.

In [9], the objective is to maximize the total expected profit with the help of considering SLA multidimensional resource allocation scheme for multi-tier services. Basically, the author is considering three dimensions of a server:

(1) Processing power

(2) Memory usage

(3) Bandwidth.

Here author is giving more concentration on SLA; they are defining two types of SLA:

(1) Gold SLA: response time is guaranteed and if cloud server provider is violating the constraint then they have to pay penalty.

(2) Bronze SLA: some limited amount of time response time may be the delay, then this case no penalty has to pay.

In [10], the author introduces the concept of "skewness" which is to identify the unevenness utilization of a server. Author's main aim is to minimize the skewness with the help of this we can maximize the utilization of a server. They designed a Load prediction algorithm, which is used to identify the current workload of a VM. The main aim of authors is to achieve the following two things:

(1) Overload Avoidance: Load prediction algorithm is used to identify the current workload of Physical machines. If any physical machine is overloaded, then migrate the VM to another available of PMs.

(2) Green Computing: if any physical machine is under loaded then try to transfer that VM, is running on that PM, to some another PM and turned off that PM, with the help of this number of PM is minimized.

Priority based dynamic resource allocation in cloud computing:

In [11] Chandrasekhar et al. proposed an algorithm to reduce the makespan of VMs with the help of Priority Based scheduling algorithm (PBSA).Honey bee behavior inspired load balancing of tasks in cloud computing environments:

In [12] Krishna et.al proposed a basic model to reduce the load of the overloaded VMs considering priority as well. Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non preemptive Dependent Tasks:

In [13] Objective is to assign the computational tasks to the most suitable virtual machines from the dynamic pool of the VMs by considering the requirements of each task and the load of the Vms.

3. SYSTEM MODELS

To make the allocation of resources on basis of task, the algorithm is fed with sorted Tasks and hence the name Sorted Task Based Allocation Algorithm. The results for the sorted tasks show better performance. Allocation of tasks in the cloud is an NP-hard optimization problem [14]. Load balancing of non-preemptive independent tasks on VMs is an important aspect of task allocation in the cloud [16]. The main objective of this problem is to balance the load and speed up the execution of applications and hence minimizing the required time to complete all the tasks of the VMs. This clearly defines the makespan to be the required time to complete all tasks. The goal here is to maximize the resource utilization and reduce the makespan.

Objectives of the problem:

(1) Reducing the makespan

(2) Utilizing the resources to maximum extent

(3) Maximize the throughput

Our prime objective is to minimize the makespan by applying sorted TBA (Task Based Allocation) heuristic algorithm [15]. The proposed heuristic algorithm also tries to maximize the resource utilization, memory requirements and VMs requirements, etc. which may require during the execution of tasks [17].

3.1 System Architecture

The system designed here should be able to solve two problems namely reducing the makespan and maximizing the resource usage. Allocation of tasks in the cloud is an NP-hard optimization problem [14]. Load balancing of non-preemptive independent tasks on VMs is an important aspect of task allocation in the cloud [16].The system architecture for the proposed work is explained with the help of Fig.1, this is the model which forms the basis for the proposed algorithm. The model consists of various components like Physical Machines, Virtual Machines, Task

Scheduler, Task Manager and an Interface for the Task Creation and Submission. Adding further, all the PMs (PM1, PM2 and PMk) will have resources like memory, CPU time, and network bandwidth etc. Each PM has a VMM (Virtual Machine Manager), a supervisory program which is responsible for managing all the activities related to Guets Operating Systems. Over the VMM a finite number of the virtual machines (i.e., {VM1, VM2, VMm}) will run [18].

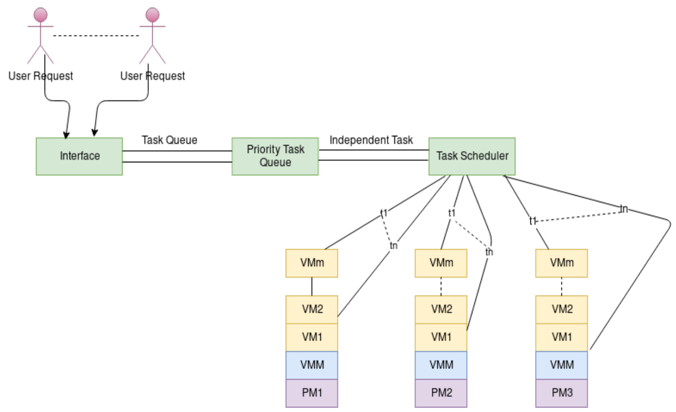

Fig -1: Cloud Computing Architecture

The cloud system model consist the following module:

1) Interface

2) Task Manager

3) Task Scheduler

(1) Interface: It is nothing but a GUI (graphical user interface), a web-page where cloud user can give the requests. These tasks are submitted to the Task Queue.

(2) Task Manager: It will take a task from Task Queue, and the Task Queue is nothing but a priority Queue. Here, higher task length or task size is given higher priority. The basic logic here is to give the higher length of task to the VM with High Speed, which will ultimately help reducing the makespan.

(3) Task Scheduler: The role of Task Scheduler here is to allocate the Tasks from the queue to the VMs. Here the makespan is reduced with the heuristic approach applied in the Sorted TBA. After execution of the task, it is the responsibility of the task scheduler to release the VMs and corresponding resources, and updates the available number of VMs and resources.[1]

3.2 System Model

The objective here is to calculate the reduced makepsan as follows:

Makespan=max {CT ij iεT, i = 1, 2, ..., n and jεV M , j = 1, 2, ..., m}

Objective Function = min {makespan}

We define our problem is a 3-tuple

S= {PM, VM, TM} to represent the problem. PM is a set of Physical machines. VM is a set of virtual machines. TM is the task model that consist task-id, task-length and task-type. The PM contains all the resources i.e. network resources, computational resources, memory requirements. The System model is explained as below:

A. Physical Machine Model

A PM has its own hardware: Memory, network, processing and storage resources. On this hardware, the VMM is loaded [19]. It is a 6-tuples entity i.e.

P M k = {IDk, CPUk, MMk, SSk, BWk, VMMk } Where,

- IDk is the identification of k physical machine
- CPUk is the computation processing speed (in MIPS)of kth physical machine
- MMk is the capacity of main memory of k physical machine
- SSk is the capacity of secondary storage of k physical machine
- BWk is the bandwidth capacity of k physical machine
- VMMk is the Virtual Machine Monitor(VMM) running on the kth physical machine

B. Virtual Machine Model

We have m number of virtual machines running on ith physical machine [19],

VM i = {VMi1, VMi2, ..., VMim } i.e. A virtual machine can be modeled as 5-tuples,

VMij = {IDij , CPUij, MMij, SSij, BWij } Where,

- IDij is the identification of j th virtual machine running on ith PM.
- CPUij is the computation processing speed (in MIPS) of jth virtual machine running on ith PM
- MMij is the capacity of main memory of jth VM on PMi
- SSij is the capacity of secondary storage of ith physical machine
- BWij is the bandwidth capacity of jth VM on Pmi

C. Task Model

A job is a collection of task in the cloud [19]. A task in the cloud is a service request which the cloud has to provide. We have n number of independent computing tasks running on some virtual machines T ij k = {T ij 1 , T ij 2 , ..., T ij n } i.e.

A task model can be modeled as 5-tuples,

Tijk = {IDij, Lkij , Rij} Where ,

- IDkij is the identification of kth task running on Vmij.
- Lij is the length of the kth task running on Vmij.
- Rkij represents the service-type 0 → CP U − intensive, 1→Data − intensive, 2→Communication − intensive. running on Vmij

3.3 Algorithm

3.3.1 ETC Matrix Generation Algorithm

The key here is to give tasks with higher length to high speed VMs. This logic will make sure that the task is executed in minimum time. It will return higher task length first, then we are calculating ETC (Expected Time to Complete) matrix by calling the ETC generation algorithm. This algorithm generates what time each task will take on all the available VMs and will return a Matrix for the same.[1]

Algorithm 1: ETC generation algorithm

Input: No of VMs, List of VMs Speed, No of Task, List of Task Size

Output: ETC Matrix

```
begin
    for i = 0 to (No of T ask − 1) do
        for j = 0 to (No of V M s − 1) do
            ETC[i][j] = List of Task Size[i]/List of VMs Speed[j];
        end
    end
return ET C
end
```

3.3.2 Task Based Allocation Algorithm

Algorithm 2: Sorted Task Based Allocation (TBA) algorithm[2]

Input: ETC matrix, VM waiting list, No of Task, No of VMs

Output: makespan

```
begin
    for i = 0 to (No of T ask − 1) do
    ind = 1, max = ∞, makespan = 1;
    for j = 0 to (No of V M s − 1) do
        if wttime of VM + ETC[i][j]<max then
            ind = index of currentVM ;
            max=wttime of VM + ETC[i][j]
        end
    end
    task assigned[i] = ind;
    wttime of VM + = ETC[i][ind] ;
    end
return makespan = max{wttime of VM }
end
```

3.3.3 Flow Chart of the System

Fig. 2 shows the flow chart of proposed algorithm. Firstly, it will accept the task, and then, it will store into the Max-Heap queue. Task manager picks the task from the queue and it will send the task to the task Scheduler. Task scheduler checks the availability of VMs in the system. If VM is not available then create the new VM instance. If VM is available then choose appropriate VM and assigned that task to that VM. After execution of the task, it is the responsibility of the task scheduler to release the VMs and corresponding resources, updates the available no of VMs and resources.[1]

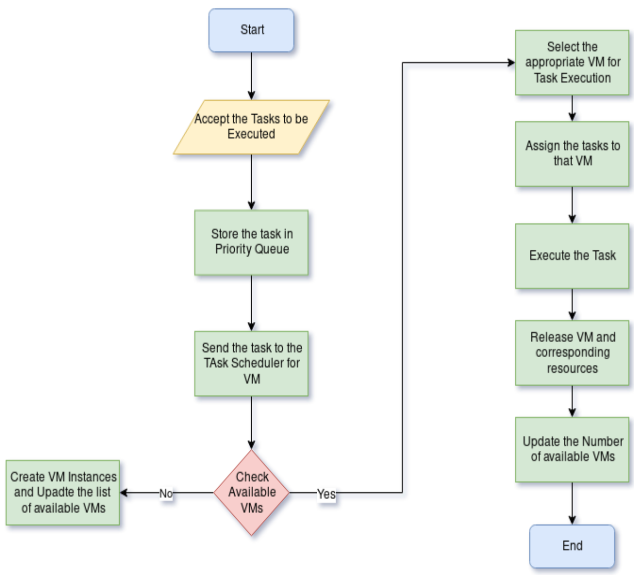

Fig-2: Flow Chart of the System

4. SIMULATION ON CLOUD SIM

CloudSim is a new open source toolkit developed using java that generalized, and advanced simulation framework allows simulation of Cloud computing and application services. CloudSim is a simulation tool for creating cloud computing environment and used as the simulator in solving the workflow scheduling problem. CloudSim allows us to create a data center with a set of hosts and number of virtual machines as resources. Each task of a workflow can be assigned to appropriate virtual machine once it's all parent tasks get executed.

4.1 Simulation Description

Virtual Machine- It is implemented virtual software of a computer that executes application programs same as a physical machine.

Cloudlet- Cloudlet is input job or set of tasks to be executed in cloud environment for. Cloudlet has its own unique Cloudlet_id, and Cloudlet_length.

The result analysis was conducted on HP PC with 2.0 GHz Intel i3 CPU and 4 GB of memory running windows 8 and CloudSim. There will be a single Data Center created. Here two simulations are done, One with fixed size of 80 virtual machines and varying tasks from 100 to 600 while for the other simulation the number of tasks are fixed to 400 an f virtual machines vary from 40 to 160. The simulation outputs for three performance parameters namely makespan, cost and execution time.[1]

5. GRAPHS

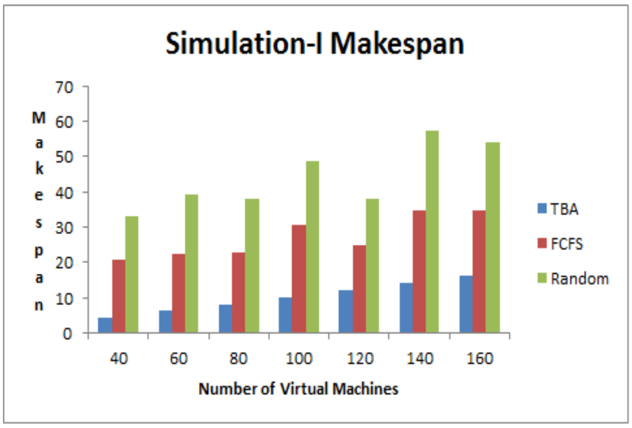

Chart -1: Simulation I for Makespan

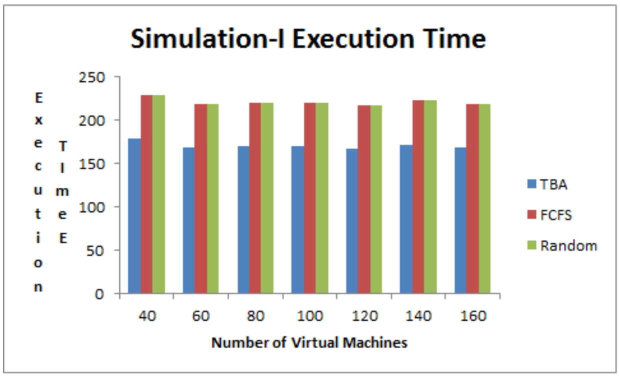

Chart -2: Simulation I for Execution Time

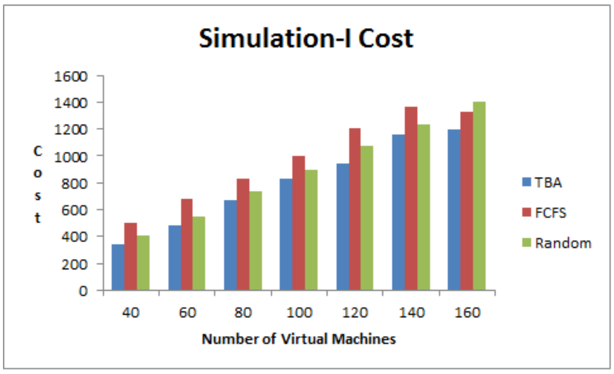

Chart -3: Simulation I for cost

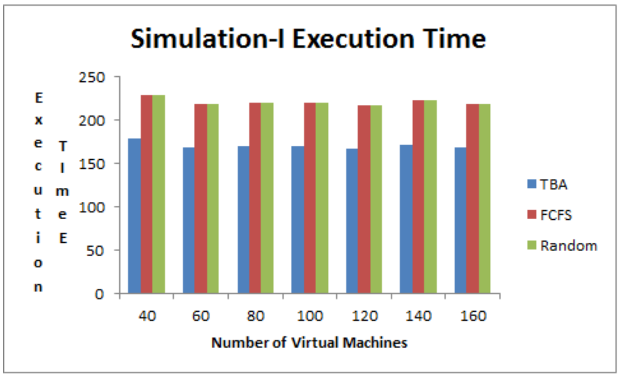

Fig- 6 Simulation-I for Execution Time

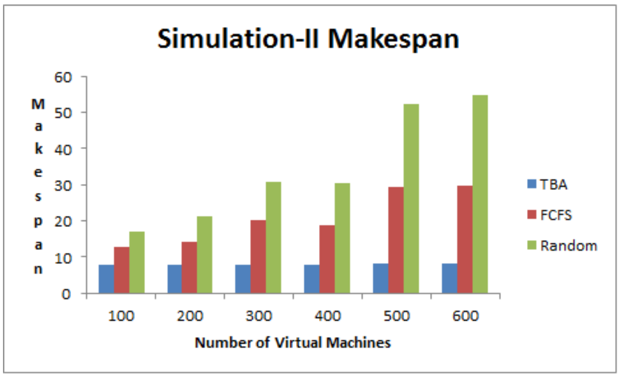

Chart -4: Simulation II for Makespan

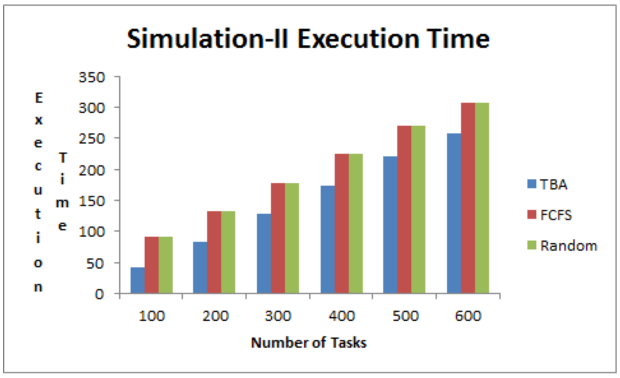

Chart -5: Simulation II for Execution Time

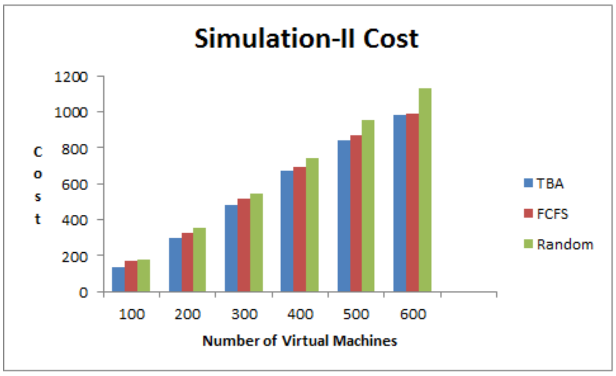

Chart -6: Simulation II for Cost

Fig- 9 Simulation-I for Execution Time

6. CONCLUSION

The Sorted Task Based Algorithm uses a heuristic approach for dynamic task allocation in heterogeneous cloud computing environment. The system model consists of PM model, VM model, and task model. The sorted TBA algorithm, gives higher priority to higher task size/task length by assigning them to VMs with high speed. This helps to improvise the performance than unsorted tasks. The ETC (Expected Time to Complete) Matrix is used to implement TBA. The sorted TBA algorithm has improved makespan for various performance parameters like makespan, cost and execution time.

REFERENCES

[1] Sambit Kumar Mishra, Md Akram Khan, Bibhudatta Sahoo, Sanjay Kumar Jena, "Time Efficient Task Allocation in Cloud Computing Environment", 2nd International Conference for Convergence in Technology (I2CT), pp. 715-720, April 2017.

[2] V.Vinothina, Sr.Lecturer, Dr.R.Sridaran, Dean Dr.PadmavathiGanapathi, "A Survey on Resource Allocation Strategies in Cloud Computing",(IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 3, No.6, 2012.

[3] S. L. S. Xavier, "A Survey of Various Workflow Scheduling Algorithms in Cloud Environment," International Journal of Scientific and Research Publications, Vol3 ,No.2, pp. 1-3, February 2013.

[4] K. Ravindran, "Task Allocation and Scheduling of Concurrent Applications to Multiprocessor Systems," [Online]. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2007/EECS-2007-149.html. [Accessed 15 October 2019].

[5] D. J. Abadi, "Data Management in the Cloud: Limitations and Opportunities," Data Engineering, IEEE Computer Society, vol. 32, no. 1, 2009.

[6] P. P. P. S. S. B. S. a. S. K. J. S. K. Mishra, "Improving Energy Usage in Cloud Computing Using DVFS," 2016. [Online]. Available: http://dspace.nitrkl.ac.in/dspace/bitstream/2080/2598/1/2016_ICACIE_SKMishra_Improving.pdf. [Accessed 15 October 2018].

[7] W. S. Q. C. Z. Xiao, "Dynamic Resource Allocation using Virtual Machines for Cloud Computing Environment," IEEE Transactions on Parallel and Distributed Systems, vol. 24, no. 6, p. 1117, 2013.

[8] M.P. HadiGoudarzi, "Multi-dimensional SLA-based Resource Allocation for Multi-tier Cloud Computing Systems," in 2011 IEEE International Conference, Washington, DC, USA, 2011.

[9] C.M.R.F. YagizOnatYazi, "Dynamic Resource Allocation in Computing Clouds using Distributed Multiple Criteria Decision Analysis," in 2010 IEEE 3rd International Conference, Miami, FL, USA, 2010.

[10] S. W. a. C. Q. Z. Xiao, "Dynamic resource allocation using virtual machines for cloud computing environment," IEEE Transactions on parallel andd distributed systems, pp. pp. 1107-1117, 2013.

[11] R. B. W. C. S. Pawar, "Priority Based Dynamic Resource Allocation in Cloud Computing," in Cloud and Services Computing (ISCOS), 2012 International Symposium, Manglore, India, 2012.

[12] P. V. Krishna, "Honey bee behavior inspired load balancing of tasks in cloud computing environments," Applied Soft Computing, pp. pp. 2292-2303, 2013.

[13] a. V. U. C. Devi, "Load Balancing in Cloud Computing Environment Using Improved Weighted Round Robin Algorithm for Non preemptive Dependent Tasks," The Scientific World Journal, 2016.

[14] S. Kumar, and R. H. Goudar,"Cloud Computing-Research Issues, Challenges, Architecture, Platforms and Applications: A Survey." International Journal of Future Computer and Communication, 1(4), pp. 356, 2012.

[15] K. S. Sahoo, B. Sahoo, R. Dash, M. Tiwary, and S. Sahoo,"Network Virtualization: Network Resource Management in Cloud." Resource Management and Efficiency in Cloud Computing Environments, pp. 239, 2016.

[16] S. Sahoo, S. Nawaz, S. K. Mishra, and B. Sahoo,"Execution of real time task on cloud environment." In 2015 Annual IEEE India Conference (INDICON), pp. 1-5, December 2015.

[17] D. Puthal, S. Nepal, R. Ranjan, and J. Chen, "DLSeF: A Dynamic Key Length based Efficient Real-Time Security Verification Model for Big Data Stream." ACM Transactions on Embedded Computing Systems, Vol,16(2), pp.51, 2016.

[18] M. Tiwary, K. S. Sahoo, B. Sahoo, and R. Misra, "CPS: a dynamic and distributed pricing policy in cyber foraging systems for fixed state cloudlets." Computing, pp. 1-17, 2016.

[19] S. K. Mishra, R. Deswal, S. Sahoo, and B. Sahoo, "Improving energy consumption in cloud," In 2015 Annual IEEE India Conference (INDICON), pp. 1-6, December 2015.

BIOGRAPHIES

Mtech 2nd Year Student at DBATU, Raigad, Maharashtra