Task Scheduling in Cloud Computing

[1]Abdul Razaque
Department of Electrical Engineering and Computer Science
Cleveland State University, OH, USA
[1]a.razaque@csuohio.edu

[2]Nikhileshwara Reddy Vennapusa
Department of Electrical Engineering and Computer Science
Cleveland State University, OH, USA
[2]n.vennapusa@vikes.csuohio.edu

[3]Nisargkumar Soni
Department of Electrical Engineering and Computer Science
Cleveland State University, OH, USA
[3]n.v.soni@vikes.csuohio.edu

[4]Guna Sree Janapati
Department of Electrical Engineering and Computer Science
Cleveland State University, OH, USA
[4]g.janapati@vikes.csuohio.edu

[5]khilesh Reddy Vangala
Department of Electrical Engineering and Computer Science
Cleveland State University, OH, USA
[5]a.vangala17@vikes.csuohio.edu

***Abstract*—Wireless Cloud computing delivers the data and computing resources through the internet, on a pay for usage basis. By using this, we can automatically update our software. We can use only the space required for the server, which reduces the carbon footprint. Task scheduling is the main problem in cloud computing which reduces the system performance. To improve system performance, there is need of an efficient task-scheduling algorithm. Existing task-scheduling algorithms focus on task-resource requirements, CPU memory, execution time and execution cost. However, these do not consider network bandwidth. In this paper, we introduce an efficient task-scheduling algorithm, which presents divisible task scheduling by considering network bandwidth. By this, we can allocate the workflow based on the availability of network bandwidth. Our proposed task-scheduling algorithm uses a nonlinear programming model for divisible task scheduling, which assigns the correct number of tasks to each virtual machine. Based on the allocation, we design an algorithm for divisible load scheduling by considering the network bandwidth.**

***Keywords*—Wireless Cloud Computing, carbon footprint, network bandwidth.**

I. INTRODUCTION

A Cloud is the collection of interconnected computer that are provided by one or more unified computing resources" [1]. In this growing market of business and organization, cloud computing is the alternative for their day-by-day increasing needs. A Cloud provider first constructs a computing system called cloud in this we have several virtual machines interconnected through this the provider processes the task of the users. "Cloud computing is not a well-behaved model for providing wanted, user- required, flexible access to a shared pool of configurable computing resources that can be quickly provided and released with low care effort or service are going to study the divisible task scheduling of high performance computing algorithms" [2]. Cloud-computing environment where multiple virtual machines (VMs) can share physical resources (CPU, memory, and bandwidth) on a single physical host and multiple VMs can share the bandwidth of a data center by using network virtualization. Because many users and applications essentially share system resources, a proper task-scheduling scheme is difficult to resource utilization and system performance. Many system parameters, such as processor power, memory space, and network bandwidth, affect the efficiency of task scheduling.

In addition, difference in computing sources in different nodes adds to the complexity of task scheduling. Furthermore, frequent data exchange among nodes, hosts, and clusters in data-intensive cloud applications makes the task-scheduling procedure extremely complicated. Most of these methods focus on allocating CPU and memory resources to various cloud-computing tasks, assuming that all physical nodes and VMs have unlimited network bandwidth. The multi-dimensional task-scheduling algorithm is based on the availability of CPU, memory, and VMs. This algorithm considers the limitation of resources, and provides resources according to task needs and resource loads.

However, this algorithm did not consider the bandwidth requirements of tasks, nor did it consider the dynamic change of their resource requirements. Here system resources like CPU, memory and bandwidth are used by many users, so it is little difficult to construct an efficient task scheduling algorithm. The efficiency of the algorithm is effected by many things like the processor power, speed, space and memory [3]. On behalf of this the heterogeneity of computing sources also causes damage to the efficiency of algorithm. The main two problems in computing are resource. Cloud computing provides three types of service models. "Software as a service, platform as a service and infrastructure as a service [4]". Generally, task scheduling is the main process in infrastructure as a service model.

While scheduling the task we consider virtual machines as scheduling machines. The main aim of task scheduling algorithms in cloud environment is to maintain the correct load on processors by considering the network bandwidth and increase their usage, efficiency and to reduce their task execution time. The reminder of the paper is organized as follows: Section-II gives an overview of existing approaches. Section-III discusses the problem identification and significance. Section IV presents the proposed task-scheduling in cloud computing. Section V discusses an implementation and results and finally entire paper is concluded in section VI.

II. RELATED WORK

We have so many existing algorithms which deals with task scheduling but these algorithms does not deals with network band width. Wang et al proposed a "multi-dimensional task scheduling algorithms according to the availability of CPU memory but does not consider about the network bandwidth and the main defect with this type of scheduling is it does not consider the dynamic change of resource requirement". [5]

"A two level task scheduling scheme which is based on load balancing algorithm". In this first, we assign user applications to virtual machines and in second level, we will provide proper host resource to virtual machine. However, this algorithm does not consider the network bandwidth and the usage of resources for each task "Non-linear programming model was proposed to reduce the cost of the data transfer and task execution time but this does not consider network band width". Therefore, in this algorithm even though it takes less time to execute each but the total time of execution of all tasks is more because of the time for transmission [6]

The dynamic scheme proposed by "Wang et al" "solve the minimum congestion which reduces the sharing of the same network link by multiple resources." [7] A double fitness genetic algorithm was proposed for programming framework; it reduces not only the task completion time but also the average conclusion time. Nevertheless, in this algorithm they considered the network bandwidth is unlimited they care only about the execution time.

In [8] enhanced cost-based algorithm for the task scheduler that measures both source costs and computing performance and improves the rate of calculations. After summarizing the works done by usual cloud computing algorithms, it is the turn of heuristic algorithms. ACO and Simulated. Annealing optimization methods together with the classic methods, we have demonstrated considerable access.

A fusion algorithm is combined with the proposed in two-stages. The first stage is the same genetic algorithm stops after the last iteration, where the final product was obtained as an input to the second algorithm (fusion). Some of the other studies considers the network bandwidth but this does not considers the limitations of CPU and memory resources.

None of the above existing algorithms considers the network bandwidth along with the resource requirements for task scheduling so due to insufficient bandwidth there is a lot of wastage in resources.

III. PROBLEM IDENTIFICATION AND SIGNIFICANCE

Generally, while scheduling the task we allocate the task to each VM based on the computing power and available resources of the VM. If the VM is capable of doing the task and resources required by the task are available, then we allocate the task to VM. In this while, one task is running we are allocating the next task to the same VM without checking whether the available bandwidth is sufficient for the task or not then, the task is in waiting position without allowing the next task to execute. Due to this, the total execution time of the entire task increases which cause the wastage of resources.

Therefore, we are checking the available bandwidth of the both virtual machine and the task required, and allocating the task based on bandwidth computing power and the resources of the virtual machine to decrease the wastage of resources and execution time

IV. TASK SCHEDULING IN CLOUD COMPUTING

In computing, scheduling is a method by which work specified by some means is assigned to resources that complete the work. It may be virtual computation elements such as threads& processors or data flows, which are in turn scheduled onto hardware resources such as processors.

A scheduler is carries out the scheduling activity. Schedulers allow multiple users to share system resources properly, or to achieve a good quality of service. Scheduling is fundamental to computation, and an internal part of the execution model of a computer system, the concept of scheduling makes it possible to have computer multitasking with a single CPU.

Preference is given to any one of the concerns mentioned above, depending upon the user's needs and objectives. Many parallel applications consist of multiple computational components. While execution-tasks depend on the of other tasks, others can be executed at the same time, which increases parallelism of the problem.

A. Non Linear Programming Model:

Task scheduling and maintaining of resources are the main problems in cloud computing. Mainly the task scheduling effects the high performance of the system. To reduce the wastage of resource we are considering the network bandwidth while constructing the algorithm, here we are proposing a non-linear programming model for task scheduling. In this we have n' computer hosts such as $\{H_1, H_2, \dots, H_n\}$ and each of these hosts has a virtual machine with its corresponding virtual machine monitors and here we adopt a "bounded bandwidth multipart communication model", which is used for heterogeneous distributed computing environment. In this we have m virtual machines with computing power, time needed to execute each task is cp_1 , cp_2... cp_m . Here we are considering the limited network bandwidth, by this non-linear programming method we can allocate the proper number of task to each virtual machine based available network bandwidth.

Algorithm 1: Execution time reduction

1. Initialization
   a. Ti: tasks
   b. Bw(Ti): Bandwidth of tasks
   c. Bw(VM): Bandwidth of Virtual Machine
2. N= number of tasks submitted
3. Bw ← each VM
4. Bw ← each task
5. Nonlinear programming model.
6. n for each VM
7. queue (t1,t2,t3,t4,………….,tn)
8. n1 for idle Vm
   No ←wait for idle
   Yes← Checks for Bw (available)
9. Execute the task

Waits for running to complete

In this method we have schedule method s0, which schedules the task to each virtual machine and here each virtual machine has a particular bandwidth let Bi and the outgoing bandwidth as B0. Here we have to take set of independent tasks which needs same execution time let it be 'g'. We have to check the scheduling node response to virtual machines simultaneously and the number of virtual machines connected to the virtual node must be less, we have to look that each virtual machine node has time to complete each task and the bandwidth is greater than 0. This solution gives the correct number of tasks to virtual machine. Here we considered band width required by each task is bi and the total band width is B0 and the bandwidth allowed to each virtual machine such that

bi<=Bi bi<B0 (1)

If we consider the number of tasks as 'M' then the number of tasks allowed to each VM (Vmm) is less than M

Vmm < M (2)

Virtual machine bandwidth must be greater than zero

VM (B.W) >>0 (3)

For better performance, we have to maintain the transmission time (Tt) of each task is less than execution time of task (ET)

Tt<<ET (4)

Scheduling node must communicate simultaneously with all nodes. The task scheduling process is shown in Fig.1.

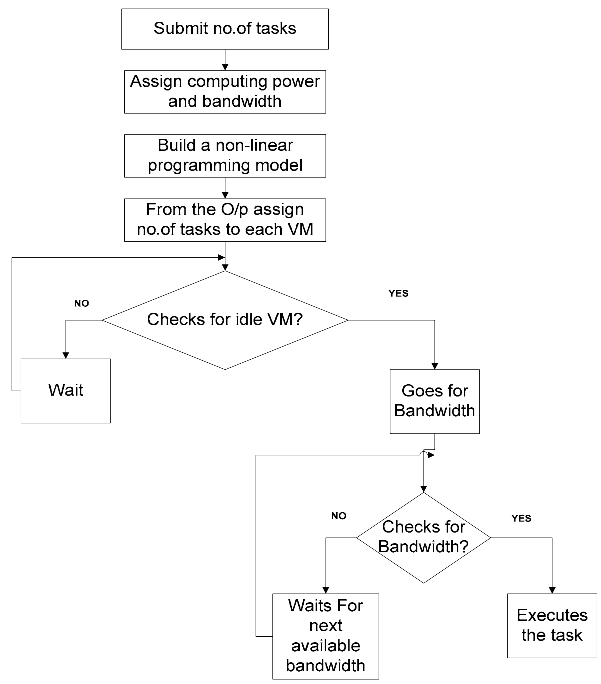

Fig 1: Showing task scheduling to VM's

Algorithm 1 explains the efficient scheduling of tasks while allocating the task this algorithm considers the computing power of the virtual machine, resources and the bandwidth availability, so it allocates the tasks in such a way that reduces the waiting time of tasks

- Submitting the tasks to virtual machine
- Assigning Bandwidth to virtual machine
- Assigning Bandwidth to tasks
- Performing non linear programming model
- From the result of the above model we are assigning tasks to each virtual machine
- Tasks are in queue
- Tasks checks for the idle virtual machine

- Then tasks checks for the required band width
- Executes the tasks

V. IMPLEMENTATION AND RESULTS

We are going to simulate our program in java. Here we consider four input tasks such as ΔTA, ΔTB, ΔTC , ΔTD and the allocated bandwidth to each virtual machine is ΔB1,ΔB2…When all the tasks are in queue and ready to execute. First the task ΔTA checks for the idle virtual machine and then checks for bandwidth and allows to execute

If ΔTA< ΔB1 (5)

Then Virtual machine allows task A to execute,Now we have

ΔB1- ΔTA= ΔX (6)

ΔX is the remaining available bandwidth while running the task ΔTA.

Now the second task checks the available band width

ΔX - ΔTB = ΔY (positive) (7)

Then it allows task ΔTB to execute along with the task ΔTA

ΔX-ΔTB = ΔZ (negative) (8)

Then it directly go and checks for the next task

ΔX - ΔTC = ΔM (positive) (9)

Then it executes the third task such that it allows execution by checking the available bandwidth so that it reduces the waiting time and consumption of resources

Then it directly goes and checks for the next task

ΔX - ΔTC = ΔM (positive) (10)

Then executes the third task such that it allows execution by checking the available bandwidth so that it reduces the waiting time and consumption of resources. We simulate our proposed task scheduling approach to reduce the execution time. We considered all the tasks are independent to each other and they require only one resource to complete the task

TABLE 1: Experimental Specifications

Parameters	Description
Platform	Eclipse
Code	Java
Operating System	OS X El Capitan
Version	10.11.1
RAM	8 GB
Processor	2.7 GHz intel corei5
Graphics	Intel Iris Graphics 6100 1536 MB

The simulation parameters are given in Table1 and the description of the parameters used and the platform to execute the proposed plan and the hardware requirements used. We used Java language to perform this, operating system is OSX El Capitan, and the version used is 10.11.1.

The processor we are using are 2.7GHz processor intel core i5 and we are writing code for this and we are specifying number of tasks and VMs and the power in different programs and we are calling all these programs in the main program.

In figure 2, we take the tasks on x-axis and the number of time on y-axis, and while considering without bandwidth, we observed that 180 tasks take 1425 milliseconds to execute that is bit longer time.

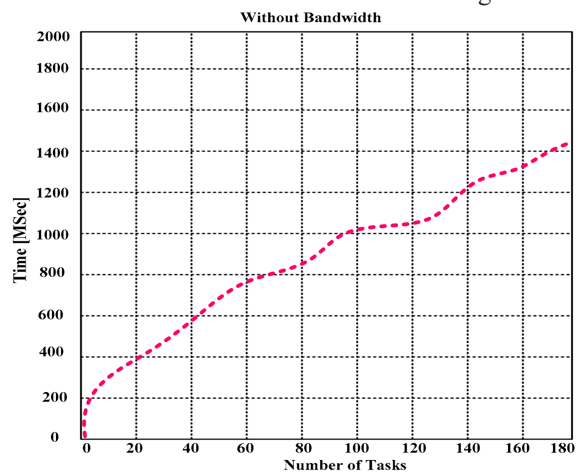

Fig 2: Showing experimentation results without bandwidth

In Figure 3, we observed that total tasks take 487 milliseconds that is almost 3 times less than without bandwidth. Our algorithm uses the parallel executing, task is stored in the queue and automatically executed, and it does not require waiting for previous task. It is validated that execution time is reduced as compared without bandwidth.

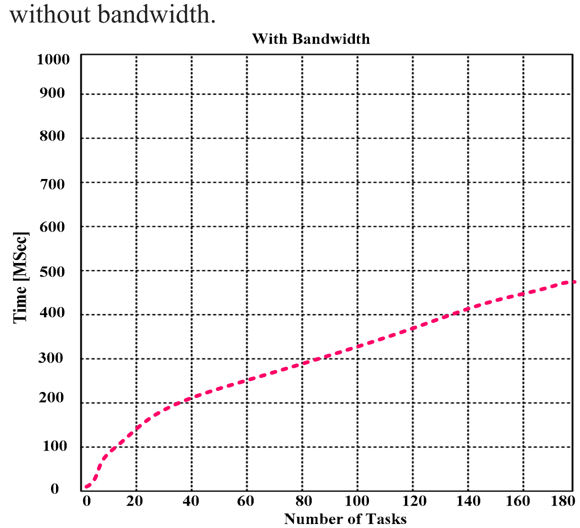

Fig 3: Showing experimentation results considering with bandwidth

VII. CONCLUSION & FUTURE WORK

In this paper, an efficient task-scheduling algorithm is proposed to eliminate the wait time. Our proposed algorithm reduces the total execution time and resource consumption. The proposed algorithm sets some conditions such as all the tasks are independent to each other. Furthermore, it automatically executes the tasks when scheduling for the execution. Based on the experimental results, we observed that our proposed algorithm is bandwidth friendly and takes less time for completion of tasks while without usage of our algorithm, tasks take a longer time to be completed. In future, we will focus on the dependency between the task and each task require more than one resource to complete the task.

REFERENCES

[1] Kaur, P.D., Chana, I. ―Unfolding the distributed computing paradigm, In: International Conference on Advances in Computer Engineering, pp. 339-342 (2010)

[2] Rizvi, Syed, Abdul Razaque, and Katie Cover. "Third-Party Auditor (TPA): A Potential Solution for Securing a Cloud Environment." *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on*. IEEE, 2015.

[3] Mei, L., Chan, W.K., Tse, T.H., ―A Tale of Clouds: Paradigm Comparisons and Some Thoughts on Research Issues‖, In: APSCC 2008, pp. 464-469 (2008)

[4] Silva, J.N., Veiga, L., Ferreira, P.: ―Heuristics for Resource Allocation on Utility Computation Infrastructures. In:6th ieee International Workshop on Middleware for Grid Computing, New York (2008)

[5] Wang L, von Laszewski G, Kunze M, Tao J. Schedule distributed virtual machines in a service oriented environment. Proceedings of the 2010 24th IEEE International Conference on Advanced Information Networking and Applications, 2010;230–236.

[6] Fang Y, Wang F, Ge J. A task scheduling algorithm based on load balancing in cloud computing. Web Information Systems and Mining, Lecture Notes in Computer Science 2010; 6318: 271–277

[7] Grounds NG, Antonio JK, Muehring JT. Cost-minimizing scheduling of workflows on a cloud of memory managed multicore machines . proceedings of the 1st International Conference on Cloud Computing, Lecture Notes in Computer Science, vol. 5931, 2009; 435–450.

[8] Shrestha HK, Grounds N, Madden J, Martin M, Antonio JK, Sachs J, Zuech J, Sanchez C. Scheduling workflows on a cluster of memory managed multicore machines. Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications, 2009

[9] Pandey S, Barker A, Gupta KK, Buyya R. Minimizing execution costs when using globally distributed cloud services. Proceedings of the 24th IEEE International Conference on Advanced Information Networking and Applications, 2010; 222– 229.

[10]. Wang C, Wang C, Yuan Y. Dynamic bandwidth allocation for preventing congestion in Datacenter networks. Proceedings of the 8th International Symposium on Neural Networks, Guilin, China, May 29–June 1, 2011.