A Simulation of Priority Based Earliest Deadline First Scheduling for Cloud Computing System

[1]Garima Gupta, [2]Vimal Kr.Kumawat, [3]P R Laxmi, [4]Dharmendra Singh [5]Vinesh Jain, [6]Ravinder Singh
Department of Computer Science
Government Engineering College,
Ajmer, India
[1]guptagarima09@gmail.com, [2]vimalkumar1001@gmail.com, [3]laxmigweca@gmail.com, [4]dharmendra.singh234@gmail.com,
[5]vineshjain128o@gmail.com, [6]ravikaviya@gmail.com

***Abstract* – Cloud is a new trend in the computing. It is the combination of distributed, parallel and grid computing. It is a pay on demand computing. Basically cloud is a shared pool of resources. The resources are shared among users who are geographically distributed. As the area of cloud is increasing, several issues are also increasing. These issues are related to security, resource allocation, performance and cost. The paper has its main focus on the resource allocation problem and to solve the problem we are discussing a task allocation algorithm. The algorithm that we will discuss in this paper uses the two task scheduling algorithm one is priority based and other is earliest deadline first scheduling algorithm. The tasks are allocated on the basis of their priority and the task having higher priority get scheduled first. The simulation is performed on the CloudSim toolkit. CloudSim is a JAVA based simulator. The proposed work will results in higher performance and also improves memory utilization.**

Keywords –cloud computing, priority based, earliest deadline first, CloudSim

I. INTRODUCTION

Cloud computing can be defined as a computing environment. From NIST definition "cloud computing is a model that enables convenient and on demand network access to a shared pool of computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction." Resources in the cloud are permanently stored at the severs and cached provisionally for the clients. Cloud is a utility based computing that makes the software attractive and provide the resources in the pay-per- use basis.

All data in cloud is stored online. This storage model of cloud is known as online network storage model. The data is stored in a virtualized pool of storage. This pool of virtualized storage is generally driven by a third party vendor. Infrastructure as a Service (IaaS) provides the storage, networks, processing and computational resources to the clients[2]. IaaS clients or users submit their programs and the related data, and at the vendor's side all the computation processing is done. Resource allocation and task scheduling is a new area of research. The main goal of cloud computing is to provide resource sharing among its users. But while sharing the resources several conflicts are arises between the resources like isomerism, different user demands and dynamic nature of resources.

These conflicts make scheduling more complex [3]. So resource scheduling is an important research area in the cloud computing. Utilization rate of resource is a major challenge in scheduling and it depends on the scheduling mechanism used. Scheduling is a decision based policy. It deploys resources to the different application during a specific period of time. An optimized schedule considers some factors like cost, performance, priority and profit. Scheduling policies are generally focused on the allocation and management of resources for guaranteed resource utilization rate. The main aim of a scheduling policy is to satisfy the users need. Scheduling policies improve the completion time of the application, also increase theprofit of service party, and reduce cost of the applications. Resources in cloud can be accessed at any given time and from any location via internet.

Data centers in cloud consume a huge amount of energy. So energy aware computing systems are needed in cloud computing that consumes less amount of energy [4]. Resource demands for different jobs vary every time. SLA is another fundamental issue in job scheduling systems. In the cloud computing system, there are some application that deals with light-weight tasks. Scheduling these fine grained tasks is not only costly but also these tasks consumes extra amount of waiting time and turnaround time by comparing with the coarse-grained task allocation to resource.

Total turnaround time of a task includes scheduling time, transmission time and execution time of each task. A huge amount of fine-grained tasks will spend maximum of their time on scheduling and transmission. The total turnaround time of the fine-grained task can be reduced by combining these fine-grained tasks with the coarse-grained tasks in the entire scheduling process [3].

In this paper, we present a priority based earliest deadline first algorithm with task migration. Our algorithm gives priority to the task and allocates the resources according to their priority and migrate the resources whenever they miss their deadline. Thus the algorithm improves the overall response time and provides a better performance. In the rest of the paper we will discuss the

978-1-4799-3486-7/14/$31.00

simulation tool CloudSim that is a Java based simulation toolkit.

II. RELATED WORK

Architecture has been proposed in [5] that uses a control theory that is based on the feedbacks. The architecture manages the virtualized resources adaptively based on their VM. In the proposed architecture the hardware resources that are available collected into a common place that is shared such that all the required resources can be accessed by hosted applications as they are required to fulfil the application's Service level objective. The architecture uses an adaptive manager that has multi input multi output resource manager and it also have CPU controller, I/O controller, memory controller. The main goal is to maximize the utilization for multiple virtualized resources so that Service level objective of an application can be achieved by using the control inputs.

Walsh in [6] has proposed a layered architecture which considers the utility functions, and adopted an autonomous and dynamic allocation of resources that have some global arbiter and local agents. Local agents calculate the utilities for given current workloads and for the number of resources, for every AE and then the calculation results are moved to the global arbiter. Global arbiter then calculates the resource's near optimal configuration which are the results transferred from the local agents. Assignment of new resources to the AEs can make a new configuration and the computation is either done in an event triggered way or by the fixed control intervals.

A task model is proposed that considers both penalty and profit while executing the task [7] . The goal of this profit penalty aware task scheduling is to increase the profit obtained in a timely manner for the online service system. Here, tasks are associated with a pair of unimodal time functions that represents the accrued profit of the system. Hence in the middle of the execution of a task, information about both penalty and profit cannot be presented the utility function simultaneously.

[8] Proposes a approach for the management of dynamic and autonomous resources in computing the clouds. The architecture proposed in the paper has some autonomous node agents. Here architecture uses independent tasks. These node agents perform the independent tasks of the resource management.

In [9] a job scheduling method for the queue system is proposed that is history based. The scheduling method finds an appropriate machine to automatically allocate the jobs. To allocate the jobs first it finds the estimated time of a job. A solution using binary integer for the scheduling problem is proposed in [10] the solution also calculates the computating costs of this technique for the key parameters of the problem.

Xian and Peide proposed a scheduling system architecture for the cloud storage environment [11] . This architecture considers multiple criteria and shows a multi faced nature of a scheduler in cloud. The main aim of paper is to support the mass storage of stream media in cloud storage and provide the continuous access of the desired data to its end users while virtual machines are migrating.

An adaptive resource allocation scheduling algorithm is proposed in [12] for the cloud system having pre-empt able tasks. Main purpose of the proposed algorithm is to update the task execution based on adaptive resource allocation. The task scheduling uses two algorithms basically (i) adaptive min min scheduling algorithm and (ii) adaptive list scheduling algorithm. The proposed algorithm schedules the tasks statically. This is also called offline scheduling. A frequency is set for the evaluation of the resources. A continuous approach is used for re-evaluation of the resource allocation that is static. The procedure that is used here for the re-evaluation is an adaptive procedure that works online. After every process of re-evaluation scheduler of each process calculate again the new finish time of the tasks related to that process.

A new approach for resource allocation that makes multiple decisions distributive is proposed by Yazir and Mathew in their paper [13]. This resource allocation approach is dynamic in nature. In this paper author describe the whole method in two steps, in the first step architecture is considered that is distributive in nature. The architecture contains a Autonomous node agent and also assumes that the management of resources can be divided into the two tasks that are independent. Every independent task is executed by the node agent. To execute the task node agent perform three activities (i) VMplacement, in this activity a physical machine to run the given VM is found once it is found than the virtual machine can be given to that physical machine. (ii) Monitoring, in this activity the node agent monitors the all resources that a VM hosted. (iii) VMselection, in this activity a VM migration will take place. The whole activity can be completed by using either of two steps, in the first step a VM will be migrated to other Physical machine if local accommodation is not possible also in this step processes also migrated on the other PM. Another method is the PROMETHEE where decisions are carried parallel by the node agent. The approach used in the paper is generally feasible in large data centers.

Chieu T.C., Mohindra A., KarveA. A.,Segal proposed a method for solving the dynamic scaling in web application in their paper[14]. It describes an architecture where web applications are scaled dynamically. The proposed method is used for the virtualized cloud environment and it uses a threshold value. There is a load balancer that works from the front end and a no of virtual machines for the web application in the basic architecture of the method. To manage the user request and routing operations at HTTP load balancer is used here. To run the web application an Apache web server is used and a Linux based virtual machine that installs the web server in it. The virtual machine can be started an ended in the middle using a provisioning subsystem according to the need. A dynamic scaling algorithm is used to control this provisioning subsystem. This algorithm uses the threshold value for the web applications.

M R Rawtani has proposed a new idea for knowledge management in cloud computing [15]. The knowledge management consist of a large amount of information the intelligence concepts contains 4 fundamental principles such that data, knowledge, wisdom and information. Knowledge can be defined as a combination of context experience etc. It consists of a high value for making the decision. Knowledge in cloud computing achieves efficient storage, elasticity, scalable services, high performance computing, reliable access, multi-tenant environment, security and fault tolerance. In the real time systems, the scheduling algorithms are generally priority driven. The tasks are assigned priorities according to the constraints and the task having higher priority is considers as a most urgent task [16]. In the paper a non-pre-emptive scheduling algorithm is used. Initially tasks are entered into the ready queue and then each task is scheduled non-pre-emptively. If a task is executing than it cannot be removed until it completes its execution. If a new task arrives with the higher priority than it cannot be execute until the currently executing task completes. This reduces the overall system performance.

R Santosh proposed a architecture that migrates the task on the other virtual machine when they are about to miss their deadline [2] . The proposed algorithm in their paper uses a pre-emption based scheduling. The tasks are put into the ready queue and when a new higher priority task arrives the task that is executing currently is placed again in the ready queue and waits for the completion of the higher priority task. If a task is about to miss its deadline than it is migrated to the other virtual machine. The procedure is good enough but it increases the execution time of the pre-empted tasks.

In this paper we are considering the pre-empt-able task that are scheduled on the priority basis and we will migrate the currently executing task directly to other virtual machine when there is a high priority task arrives in the middle of the execution. Our work will reduce the execution time of the pre-empted task and provides a better performance.

III. PROPOSED WORK

Various scheduling algorithms have been developed that schedule the tasks according to the various parameters like bandwidth, completion time, memory requirements etc. Earliest deadline first algorithm is one of the algorithms for the scheduling of jobs. The basic earliest deadline first algorithm schedule the task according to the deadline the task having earliest deadline will be scheduled first than the next task having earliest deadline will be scheduled. The basic EDF algorithm is non pre-emptive algorithm i.e task cannot pre-empt each other while scheduling. We are proposing an algorithm that integrates two algorithms one is prioritization and other is EDF. We are also using the concept of waiting time for the pre-empted jobs. Waiting time of a job is the time to complete its execution after pre-emption. The proposed algorithm is aims to reduce the total execution time of the pre-empt-able tasks.

A. Objectives

The general objective of the proposed work is as follows:

- We are trying to allocate the resources as per user demands.
- Assign each task a priority based on their parameters and arrange them in the descending order of their priority.
- Schedule the task according to their priority such that higher priority tasks are scheduled first than second higher priority task and so on.
- Migrating the tasks on other virtual machine if they are pre-empted in the middle of the execution.

B. Proposed Method

Step 1: In this step we initialize the input tasks and the resources. We input the task with some parameter like arrival time, deadline, burst time etc. Resources in this step will be arranged in the decreasing order of their mips. They are arranged in this way so that the processing time of the task can be reduced.

Step 2: In this step we prioritize the tasks according to their deadline. The task having earliest deadline will be scheduled first. These prioritized tasks will be placed in a ready queue. From the ready queue the tasks are processed one by one. All the incoming tasks are placed in the ready queue. If a new higher priority task arrived than it will be placed at the front of the queue and it will be scheduled first. The resource having higher mips will be processed first.

Step 3: This step adopts EDF algorithm. The algorithm is used here to reduce the waiting time of the higher priority tasks. The task having higher priority and low deadline is scheduled. To decrease the waiting time of the tasks the tasks are arranged according to the priority. Then the task having earlier deadline is scheduled first, and then the task having second earlier deadline is scheduled. The process completes when there is no other task in the ready queue. If in the middle of the scheduling a higher priority task arrives than it will pre-empt the currently executing task and the pre-empted task will follow the next step.

Step 4: if a task is pre-empted in the middle of the execution than it will be placed in the waiting queue from there it will be migrated on the other virtual machine. The tasks in the waiting queue are also arranged according to the priority. And the task having earlier deadline is scheduled first on the other virtual machine. Adding a waiting queue in the scheduling process will reduce the waiting time of the pre-empted tasks and gives a better performance.

IV. EXPERIMENTAL SETUP

The proposed work is simulated on the CloudSim platform. CloudSim implements the generic application provisioning techniques which can be extensible with minimum effort. CloudSim supports both modelling and

simulation for single and inter-networked clouds. CloudSim enables seamless modelling, experiments and simulation of the cloud computing systems and application provisioning environments [17]. CloudSim is used here to simulate the results. The simulation is performed for different set of users and resources. In this simulation some cloud resources are created and they are accessed by the tasks.

In this simulation some cloud resources are created and they are accessed by the tasks. The major factors that are considerable in this simulation are deadline, memory, and no of virtual machines. The tasks are submitted to the data enter broker. The tasks in CloudSim are packaged in the cloudlet, or we can say cloudlet is the set of all tasks. The cloudlet in CloudSim contains size of job input and output and the memory requirements in MIPS for the job. In the table 1 we have presented some parameter lists that are used in our simulation.

TABLE I. RANGE of PARAMETERS

Parameter	Minimum	Maximum
No of cloudlets	10	100
No of virtual machine in the cloud	18	150
Memory in a VM (in MB)	30	2048
Bandwidth	500	1050
Input size	200	2000
Output size	300	3000

The simulation is performed on a varying no of tasks. And the results are compared with the non pre-emptive earliest deadline first algorithm. We are comparing the two algorithms on the basis of their efficiency and average waiting time.

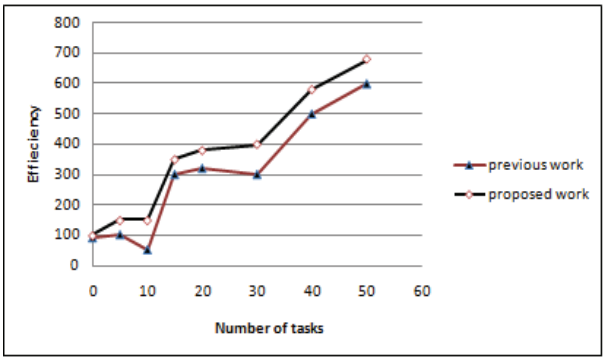

Fig1. Efficiency with different no of tasks

The figure 1 shows the relationship between different no of tasks and efficiency of the proposed work and previous work. Figure 2 shows the analysis of average waiting time in previous work and proposed work.

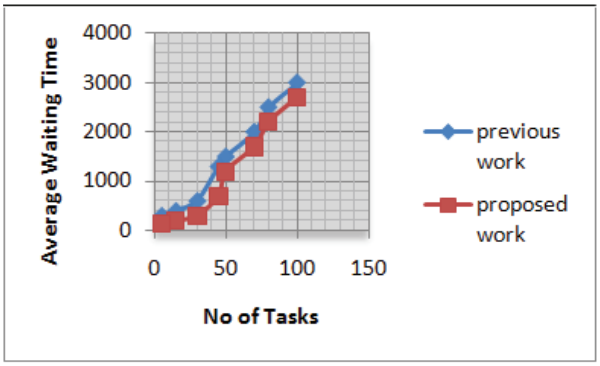

Fig 2. Average waiting time with different no of tasks.

V. CONCLUSION AND FUTURE WORK

Paper proposed a scheduling method for the tasks. In the existing method, tasks are scheduled pre-emptively and the pre-empted tasks are placed in the ready queue. Thus there is only one queue for the incoming tasks and pre-empted tasks. This approach increases the waiting time of the pre-empted tasks. In the proposed method we have introduce a waiting queue also that will process the pre-empted tasks. The tasks that are pre-empted are placed in the waiting queue and migrated on the other virtual machine. The task in this waiting queue is also arranged according to the priority and processed accordingly. Thus the given approach will reduce the execution time of the pre-empted tasks and they can be scheduled efficiently.

REFERENCES

[1] YashpalsinhJadeja, KiritModi, "Cloud Computing - Concepts, Architecture and Challenges" 2012 International Conference on Computing, Electronics and Electrical Technologies [ICCEET].

[2] R. Santhosh, T. Ravichandran, "Pre-emptive Scheduling of On-line Real Time Services With Task Migration for Cloud Computing" Proceedings of the 2013 Inter-national Conference on Pattern Recognition, Infor-matics and Mobile Engineering (PRIME) February 21-22

[3] JiaRu, Jacky Keung,"An Empirical Investigation on the Simulation of Priority and Shortest Job First Scheduling for Cloud-based Software Systems" 2013 22nd Australian Conference on Software Engineering.

[4] Jing Liu, Xing-GuoLuo, Xing-Ming Zhang, Fan Zhang and Bai-Nan Li, "Job Scheduling Model for Cloud Computing Based on Multi Objective Genetic Algorithm", IJCSI International Journal of Computer Science Issues, Vol. 10, Issue 1, No 3, January 2013.

[5] "Adaptive Management of Virtualized Resources in Cloud Computing Using Feedback Control," in First International Conference on Information Science and Engineering, April 2010, pp. 99-102.

[6] W. E. Walsh, G. Tesauro, J. O. Kephart, and R. Das, "Utility Functions in Autonomic Systems," in ICAC '04: Proceedings of the First International Conference on Autonomic Computing. IEEE Computer Society, pp. 70–77, 2004.

[7] Venkatesa Kumar, V. and S. Palaniswami," A dynamic resource allocation method for parallel data processing in cloud computing", Journal of Computer Science 8 (5): 780-788, 2012 ISSN 1549-3636 © 2012 Science Publications

[8] Y. O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in

Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing. Miami, Florida, USA: IEEE Computer Society, 2010, pp. 91 - 98.

[9] Y. MURATA, R. EGAWA, M. HIGASHIDA, and H. KOBAYASHI,"A history-based job scheduling mechanism for the vector computing cloud," in Proceedings of 2010 10th Annual International Symposium on Applications and the Intemet. Seoul, Korea: IEEE Computer Society, 2010, pp. 125-128.

[10] R. v. d. Bossche, K. Vanmechelen, and J. Broeckhove, "Cost-optimal scheduling in hybrid iaas clouds for deadline constrained workloads," in Proceedings of 2010 IEEE 3rd International Conference on Cloud Computing. Miami, Florida, USA: IEEE Computer Society, 2010, pp. 228 - 235.

[11] Xian Wu, PeideQian, “Towards the Scheduling of Access Requests In Cloud Storage”, The 8th International Conference on Computer Science & Education (ICCSE 2013) April 26-28, 2013. Colombo, Sri Lanka

[12] Jiayin Li, MeikangQiu, Jian-Wei Niu, Yu Chen, Zhong Ming, “Adaptive Resource Allocation for Preempt able Jobs in Cloud Systems,” in 10th International Conference on Intelligent System Design and Application, Jan. 2011, pp. 31-36.

[13] Yazir Y.O., Matthews C., Farahbod R., Neville S., Guitouni A., Ganti S., Coady Y., “Dynamic resource allocation based on distributed multiple criteria decisions in computing cloud,” in 3rd International Conference on Cloud Computing, Aug. 2010, pp. 91-98.

[14] Chieu T.C., Mohindra A.,KarveA.A.,Segal A., “Dynamic Scaling of Web Applications in a Virtualized Cloud Computing Environment,” in IEEE International Conference on e-BusinessEngineering, Dec. 2009, pp. 281-286.

[15] K. Dinesh, G. Poornima, K. Kiruthika, “Efficient Resources Allocation for Different Jobs in Cloud” in International Journal of Computer Applications (0975 – 8887) Volume 56– No.10, October 2012

[16] F. Casati and M. Shan (2001)., Definition, execution, analysis and optimization of composite e-service, IEEE Data Engineering.

[17] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. D. Rose, and R.Buyya, “CloudSim: A toolkit for modeling and simulation of cloudcomputing environments and evaluation of resource provisioningalgorithms,” In Software: Practice and Experience. Wiley Press, NewYork, USA, 2010.