INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH & DEVELOPMENT

ISSN 2278 – 0211 (Online)

Grouping Based Load Balancing in Cloud Computing

Dipti Ajmire
Professor, G.H. Raisoni College of Engineering, Nagpur, Maharashtra, India
Dr. Mohammad Atique
Associate Professor, S.G.B. Amravati University, Maharashtra, India

Abstract:
Cloud Computing is a new and inevitable technology in the fields of scientific, and engineering, and as well as in commercial, and industrial enterprises. In cloud computing there are many tasks that needs to be executed by the available resources to acquire high performance, reduce task completion time, minimize response time, utilization of resource usage and etc. However, user tasks developed for cloud might be small and of varying lengths according to their computational needs and other requirements. Process Migration is a technique whereby an active process is moved from one machine to another of possibly different architecture. This necessitates the capture of the process's current state of execution and recovering it on the destination machine in a manner understandable to it. My work mainly focuses on capture and recovery of the internal state of process, comprising of the execution and data state – activation history, and static and heap challenge to design an efficient scheduling strategy to achieve high performance in cloud computing. There are many existing algorithms for task scheduling but not reducing communication overhead time and computation time, and on the other hand maximizing resource utilization. The purpose of the study is to analyze and achieve better performance by taking new concept of grouping based task scheduling Therefore, this paper proposes "New Task Scheduling Grouping Base Model" with the objective of minimizing overhead time and computation time, thus reducing overall processing time of tasks.

Keywords: *Process migration, latency, cloudsim*

1. Introduction

Cloud computing always refers to Internet based development and utilization of computer technology, and hence, cloud computing can be described as a model of Internet-based computing. Actually, it is a style of computing, in which dynamically scalable (and mostly virtualized) resources are provided as a service over the Internet. [1]. A process of task scheduling is manages and maps the execution of mutually dependent tasks on the resources It allocates suitable tasks to resources so the execution is often completed to satisfy objective functions imposed by client. Scheduling is a critical problem in Cloud computing, because a cloud provider has to serve many users in Cloud computing system. So scheduling is the major issue in establishing Cloud computing systems. Suitable task scheduling will have important impact on the performance of the system. The general issue in scheduling tasks on distributed resources belongs to a category of issues called NP-hard issues. This type of problems it is difficult to find algorithms to build the optimal solution within polynomial time. Even if the task scheduling problem can be solved by using comprehensive search, the methods difficulty for solving task scheduling is very high. In Cloud environments, scheduling decision must be made in the minimum time possible, because there are many clients computing for resources, but at any time, time slots required by one customer could be taken by another customer. There are some major components in a task scheduling system: the task scheduling, data movement and fault management. Task scheduling find out resources and allocates tasks on proper resources to fulfil client requirements, whereas data movement manages data transfer between selected fault management and resources provides mechanisms for failure handling throughout execution. Additionally, the enactment engine provides opinion to a monitor therefore users can scan the task methodology standing through a task scheduling monitor [1].
The Task grouping is done found on a particular granularity size. Granularity size is the time within which a task is processed at the resources. It is used to measure the total amount of tasks that can be completed within a specified time in a particular resource. Relationship between the total number of tasks, processing requirements of those tasks, total number of available Cloud resources, processing capabilities of those resources and the granularity size should be determined in order to achieve the minimum.

2. Related Work

In this section, some of the delegate research works on Task scheduling in distributed computing systems and Cloud computing environment have been reviewed to explore the relevance of these works. Scheduling structure for Bandwidth-Aware Job Grouping-Based strategy that groups the jobs according to MIPS and Bandwidth of the resource, but the lack of the algorithm is first, groping strategies does not utilize resource sufficiently, and second, consideration of bandwidth strategy is not efficient to transfer the job[2].A Bandwidth-Aware Job Grouping-Based scheduling strategy, that groups the jobs according to the MIPS and bandwidth of resources, but short comings of the algorithm is first, the model sends group jobs to the resource whose network bandwidth has highest communication or transmission rate, but the algorithm does not ensure that resource having a sufficient bandwidth will be able to transfer the group jobs within required time[3].Constraint-Based Job and Resource scheduling (CBJRS) algorithm in which reducing the processing time, processing cost and enhance the resource utilization in comparison to other algorithms. Resources are arranged in hierarchical manner. So top of resource have maximum computational power. Job grouping strategy is use for job scheduling and it is based on the characteristics of resource. Using grouping strategy processing and communication time can be reduced.

Main factors in job grouping strategy that influences the way job grouping is performed to achieve the minimum job processing time and maximum resource utilization of the Grid resources. So the constraint based job grouping strategy give maximum utilization of the resources [4].GBJS algorithm in which reducing the communication time, processing time and enhance resource utilization. This is study presents and evaluates an extension from Computational-Communication to Computational-Communication-Memory based Grouping Job Scheduling strategy. GBJS provides real time grid computational and reduces the waiting time of the grouped jobs. Group base Job Scheduling is achieve better performance in terms of processing time. Job grouping method is gives better results for solving job scheduling problem and can achieve minimum communication time, better processing time and maximum utilization of the resources in grid computing [5]. This study focuses and evaluates an extension to dynamic grouping based task scheduling, which aims to reduce overall processing time, minimize communication overhead time and reduce cost of processing in cloud environment.

3. Proposed Scheduling Model

In the proposed model the scheduler accepts number of tasks and processing overhead of all the tasks. Resources are selected. The priority levels of the tasks are calculated. Tasks are sorted according to their priority, and they are placed in three different lists based on three levels of priority namely high priority, medium priority and low priority. Now task grouping algorithm is applied to the above lists in order to allocate the task-groups to different available resources.

3.1. Problem Formulation for Improved Grouping Based Load Balancing Algorithm (GBLBA)

To formulate the problem, define Ti, $i = \{1,2,3,\ldots n\}$ as n independent tasks permutation and Rj, $j = \{1,2,3,\ldots m\}$ as m computing resources with an objective of minimizing the completion time and minimizing the cost. The processing capacity of each resource is expressed in MIPS (Machine Instructions per second) and the size of each task in MI (Number of Machine Instructions). Suppose that the processing time Pi,j for task *i* computing on *j* resource is known. The completion time Ttot (x) represents the sum of the total computation time Texe(x) and total communication time Tcomm (x). Total computation time is calculated by adding the processing time of all the resources. The cost of every individual resource is different.

Ttot(x) = Texe(x) + Tcomm(x)

Terms used in the algorithm

n: Total number of task

m : Total number of Resources available

Gi : List of tasks submitted by the user

Rj : List of Resources available

MI : Million instructions or processing requirements of a user task

MIPS : Million instructions per second or processing capabilities of a resource

Tot-Jleng : Total processing requirements (MI) of a task group (in MI)

Tot-MIj : Total processing capability (MI) of jth resource

Rj-MIP : MIPS of jth Grid resource

Gi-MI : MI of ith task

Tot-GMI : Total length of all tasks (in MI)

Granularity Size : Granularity size (time in seconds) for task grouping

GJk : List of Grouped task

TargetRk : List of target resources of each grouped task

3.2. Improved GBLBA Algorithm

3.2.1. Algorithm for Arranging Tasks According to Their Priority Levels

- Step 1 : The tasks are received by the scheduler
- Step 2 : for all available tasks
- Step 3 : calculate their priority levels using
- Step 4 : Sort the tasks based on their priority
- Step 5: store the sorted tasks in three different lists by dividing the tasks into high, medium and low priority levels
- Step 6: If there is new task coming
- Step 7: Calculate its priority and then put it into an appropriate list

3.2.2. Task Grouping and Scheduling Algorithm

- Step 1 : The scheduler receives Number of tasks 'n' to be scheduled and Number of available Resources 'm'
- Step 2 : Scheduler receives the Resource-list R[ ]
- Step 3 : The tasks are submitted to the scheduler
- Step 4 : Set Tot-GMI (Sum of the length of all the tasks to zero
- Step 5 : Set the resource ID j to 1 and the index i to 1
- Step 6 : Get the MIPS of resource j
- Step 7 : Multiply the MIPS of jth resource with granularity size specified by the user
- Step 8 : Get the length (MI) of the task from the list
- Step 9 : If resource MIPS is less than task length
  - 9.1: The task cannot be allocated to the resource
  - 9.2: Get the MIPS of the next resource
  - 9.3: go to step 7

- Step 10: If resource MIPS is greater than task length
- Step 11: Execute steps 11.1 to 12 while Tot-GMI is less than or equal to resource MIPS and there exists ungrouped tasks in the list
  - 11.1: Add previous total length and current task length and assign to current total length (Tot-Jleng)
  - 11.2: Get the length of the next task

- Step 12 : If the total length is greater than resource MIPS
  - 12.1: subtract length of the last task from Tot-Jleng

- Step 13: If Tot-Jleng is not zero repeat steps 13.1 to 13.4
  - 13.1: Create a new task-group of length equal to Tot-Jleng
  - 13.2: Assign a unique ID to the newly created task-group
  - 13.3: Insert the task-group into a new task group list GJk
  - 13.4: Insert the allocated resource ID into the Target resource list TargetRk

- Step 14: Set Tot-GMI to zero
- Step 15: get the MIPS of the next resource
- Step 16: Multiply the MIPS of resource with granularity size specified by the user
- Step 17: Get the length (MI) of the task from the list
- Step 18: go to step 9
- Step 19: repeat the above until all the tasks in the list are grouped into task-groups
- Step 20: When all the tasks are grouped and assigned to a resource, send all the task groups to their corresponding resources GJk
- Step 21: After the execution of the task-groups by the assigned resources send them back to the Target resource list TargetRk

3.3. Flow Chart for Proposed Algorithm

To arrange tasks according to their priority levels

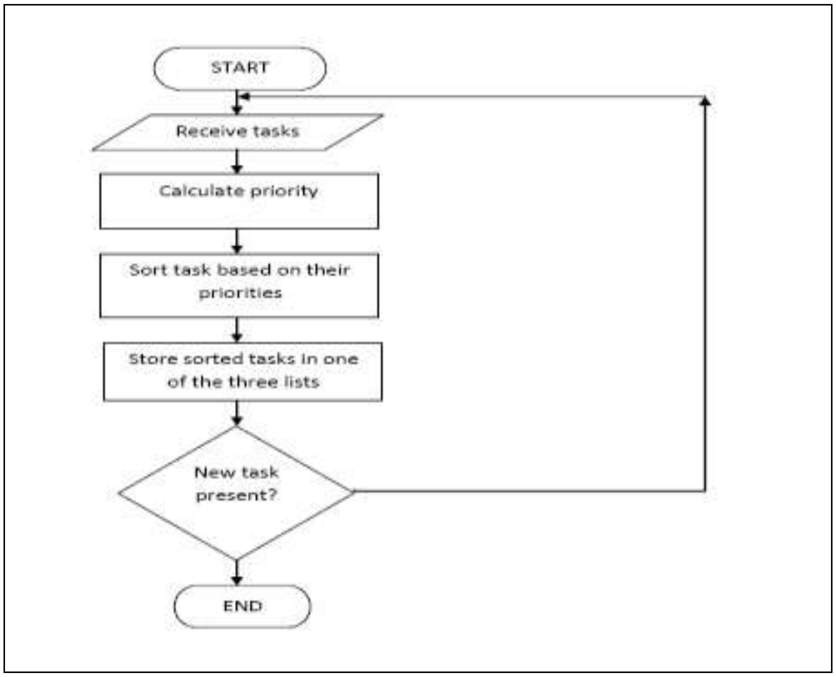

Figure 1: Flowchart to arrange tasks according to their priority levels

Task grouping and scheduling

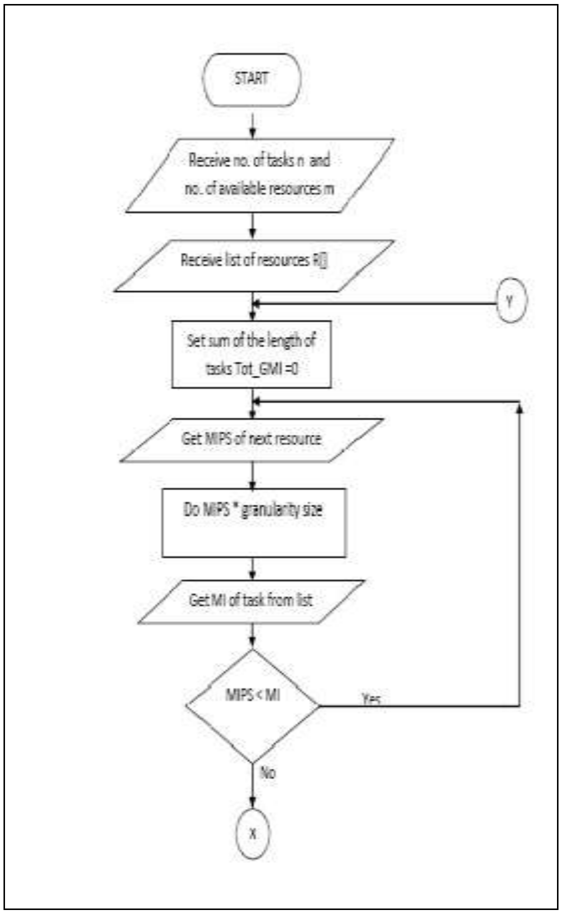

Figure 2: Flowchart to group tasks

5. Results

This work is carried out on CloudSim 2.1.1, programming is done in Java on machine having configuration, 2.10GHz Intel i3 processor, 2GB RAM. Datacenter with one host is created to have virtual machines on it. Customer can own n number of virtual machines on datacenter by paying charges for using resources of datacenter. The summary of results is stated below. The starting window is as follows.

Provider can provide infrastructure to the customer who wants to run his application using available resources of provider. Datacenter may have number of hosts in it which in turn can have number of virtual machines to process user tasks.

Care must be taken while assigning resources to customer that customer cannot be allotted more resource than what he permitted to.

After setting all above mentioned parameters when user tasks are run on VMs using either activity based or improved activity based algorithm following results are found.

User tasks are scheduled using activity based costing without grouping firstly and then executed after grouping, total execution time required to execute all tasks is as follows:

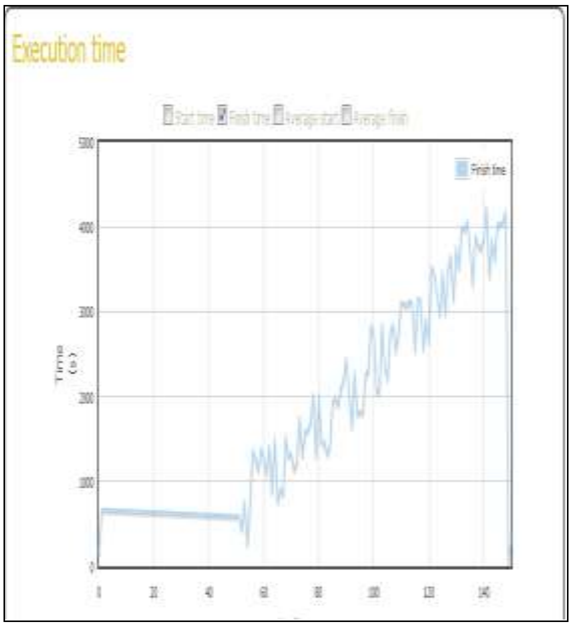

Figure 3: Execution time needed for existing algorithm

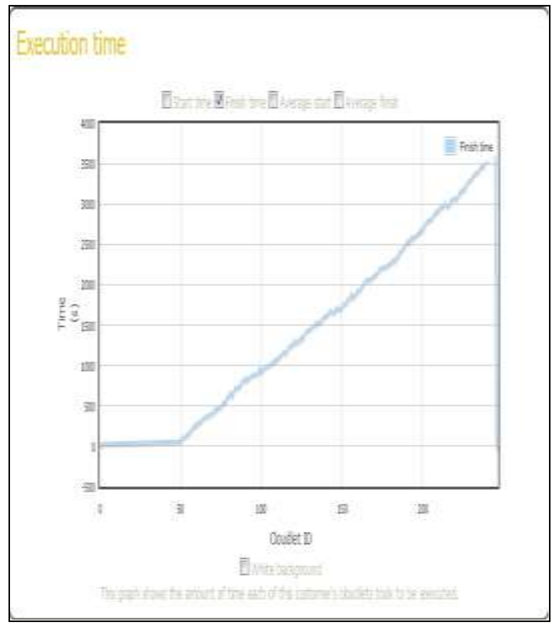

Figure 4: Execution time needed for proposed algorithm

Similarly the cost for executing user tasks using activity based algorithm without and with grouping tasks is,

Figure 5: Cost needed for existing algorithm

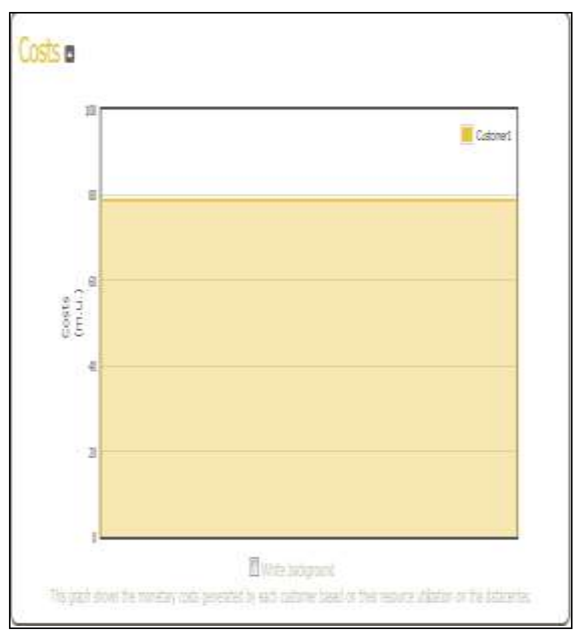

Figure 6: Cost needed for proposed algorithm

7. Conclusion and Future Work

Traditional way for task scheduling in cloud computing tended to use the direct tasks of users as the overhead application base. There are some problems with it. This problem leads to over-costed and over-priced in some high-volume simple tasks while under-costed and under-priced in low-volume complex ones. Group Based Load Balancing Algorithm is a way of measuring both the cost of the objects and the performances of activities and it can measure the cost more accurate than traditional ones in cloud computing.

Group Based Load Balancing Algorithm is a way of measuring both the cost of the resources and the computation performance. The Improved Group Based Load Balancing Algorithm method selects a set of resources to be used for computing. It groups tasks according to the processing capability of resources available. The coarse-grained tasks are processed in the selected resources, so that the Computation- Communication ratio is reduced. To reduce communication overhead the scheduling strategy should group a number of user jobs together according to a particular resource's processing capabilities, and send the grouped jobs to that resource. CloudSim can be used to carry out and simulate the tasks assignment algorithm, and task scheduling. It provides a series of core function for the establishment and simulation of heterogeneous distributed computing environment, particularly suitable for simulation and research of task scheduling on cloud.

This improved Group Based Load Balancing Algorithm algorithm takes the initial research on task scheduling in Cloud platform. However some issues remain open. Further improvement should be done to handle more complicated scenario involving dynamic factors such as dynamically changing cloud environment and other QoS attributes. The improvement of this algorithm should concentrate on discussing simultaneous instead of independent task scheduling in Cloud environment.

8. References

i. Berislav biocic, Drasko Tomic, Dario Ogrizovic, "Economics of the Cloud Computing", MIPRO 2011, May 23-27, 2011, Opatija, Croatia.
ii. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms",
iii. QI CAO, ZHI-BO WEI, WEN-MAO GONG, "An Optimized Algorithm for Task Scheduling Based On Activity Based Costing in Cloud Computing", 978-1-4244-2902-8/09/$25.00 ©2009 IEEE
iv. Michael A. Iverson, Fu¨sun O¨ zgu¨ner, and Lee C. Potter, "Statistical Prediction of Task Execution Times Through Analytic Benchmarking for Scheduling in a Heterogeneous Environment",
v. Zhi Yang, Changqin Yin, Yan Liu, "A Cost-based Resource Scheduling Paradigm in Cloud Computing", 2011 12th International Conference on Parallel and Distributed Computing, Applications and Technologies.
vi. Nguyen Doan Man, Eui-Nam Huh, "Cost and Efficiency-based Scheduling on a General Framework Combining between Cloud Computing and Local Thick Clients", 978-1-4673-2088-7/13/$31.00 ©2013 IEEE
vii. Anju Bala, Dr.Inderveer Chana, "A Survey of Various Workflow Scheduling Algorithms in Cloud Environment", 2nd National Conference on Information and Communication Technology (NCICT) 2011 Proceedings published in International Journal of Computer Applications® (IJCA)
viii. Nithiapidary Muthuvelu, Junyang Liu, Nay Lin Soe, Srikumar Venugopal, Anthony Sulistio and Rajkumar Buyya, "A Dynamic Job Grouping-Based Scheduling for Deploying Applications with Fine-Grained Tasks on Global Grids" . This paper appeared at the Australasian Workshp on Grid Computing and eresearch(AusGrid2005), Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 44.
ix. http://differencebetween.com
x. http://en.wikipedia.org/wiki/Cloud computing