International Journal of Engineering Research and Development
E-ISSN: 2278-067X, p-ISSN: 2278-800X, www.ijerd.com
Volume 10, Issue 6 (June 2014), PP.32-36

A Vibrant Optimization for Job Forecast In Cloud

Vandana
M.Tech.(IT), 4th semesester University School of Information and Communication Technology
Guru Gobind Singh Indraprasth University Sector-16 C, Dwarka, New Delhi-110075

Abstract:- Cloud computing has emerged as a popular computing model to support on demand services. It is a style of computing where massively scalable resources are delivered as a service to external customers using Internet technologies. The aim is to make effective utilization of distributed resources and put them together in order to achieve higher throughput .Scheduling in cloud is the process of selecting resources where the tasks submitted by user are executed. In other words, scheduling is responsible for selection of best suitable resources for task execution, by taking some static and dynamic parameters and restrictions of tasks' into consideration.

In this paper entitled "a vibrant optimization for job forecast in cloud", a scheduling algorithm is proposed which addresses the major challenges of task scheduling in cloud. It overcomes the problems of high task execution cost, improper resource utilization and improves the task completion time

Keywords:- granularity size, resource allocation, scheduling

I. INTRODUCTION

Cloud computing is a very current topic and the term has gained a lot of attention in recent times. It can be defined in many ways due to the lack of consensus on a single definition. One such definition describes it as on demand pay-as-per-use model in which shared resources, information, software and other devices are provided according to the clients' requirement when needed [1]. According to Forrester [2] it can be thought of as a pool of abstracted, highly scalable, and managed compute infrastructure capable of hosting end-customer applications and billed by consumption. Human dependency on cloud is evident from the fact that today's most popular social networking, email, document sharing and online gaming sites are hosted on cloud. Google, Microsoft, IBM, Amazon, Yahoo and Apple among others are very active in this field. They provide cloud computing commercial solutions in one form or another and actively sponsor research centers to explore the underlying potential of cloud.

The idea of cloud computing is based on the fundamental principal of reusability of IT resources and capabilities. Resource sharing in cloud simplifies infrastructure planning. Rapid growth of cloud technology is attributed to the fact that it gives a great majority of people and small enterprises access to new applications, platforms and infrastructure anytime, from anywhere. Key advantages of this model are ease-of-use and cost-effectiveness [3]. It has the minimum requirement on the client devices, which makes it useful and most convenient. For network applications in the cloud model, data is only one, saved in the "cloud", on the other side. It can be aptly quoted that cloud computing provides users with almost infinite possibility using the Internet. Personal and individual devices of users are limited, but the potential of cloud computing is nearly limitless. Scheduling theory for cloud computing is receiving growing attention with increase in cloud popularity. In general, scheduling is the process of mapping tasks to available resources on the basis of tasks' characteristics and requirements. It is an important aspect in efficient working at cloud as various task parameters need to be taken into account for appropriate scheduling. The available resources should be utilized efficiently without affecting the service parameters of cloud.

Proposed framework

Scheduling Modules

Task Grouping

Grouping means collection of components on the basis of certain behavior or attribute. By task grouping in cloud it is meant that tasks of similar type can be grouped together and then scheduled collectively. We can say that it is a behavior that supports the creation of 'sets of tasks' by some form of commonality. Several kinds of task groupings are observable in current scenario. Some of the candidate grouping strategies can be [12]:

1. Grouping by deadline - When several urgent tasks must be completed before a specific time, regardless of whether there are any formal relationships between them, they are grouped by deadline
2. Grouping by location - When a number of tasks are grouped by the proximity of the locations they are carried out in, it called grouping by location.

3. Grouping by participant - For example when a number of tasks are grouped by the need to involve another participant in their planning or performance.
4. Grouping by type - Tasks that relate to a specific type are grouped together such as task requiring specific resource.

In the proposed framework tasks are grouped on the basis of constraint which can be deadline or minimum cost. Once the tasks are grouped, they can be judged for their priority and scheduled accordingly. Grouping, if employed to combine several tasks into a single task, reduces the cost-communication ratio. Dynamic grouping can be done on the basis of total number of jobs, processing requirements those jobs, total number of available resources, processing capability of those resources and granularity size to achieve minimum job execution time and cost and maximum resource utilization [13].

Prioritization

Priority determines the importance of the element with which it is associated. In terms of task scheduling, it determines the order of task scheduling based on the parameters undertaken for its computation. Each task can be mapped to one of the priority queues – low, medium, high on the basis of resource usage, profit earned by task and the cost of resources [11]. High priority jobs are scheduled first and low priority task are scheduled later as their execution can be delayed.

Activity based Costing approach can also be used to determine the priority of the incoming tasks. It measures both the cost of resource and computation performance. In order to measure direct cost of applications, every individual use of resources (like CPU cost, memory cost etc.) are measured. As the direct data of each individual resource cost is measured, more accurate cost and profit analysis is obtained.

In the present framework, the deadline based tasks are prioritized on the basis of task deadline. The tasks with shorter deadline need to be executed first. So they are given more priority in scheduling sequence. The task list is rearranged with tasks arranged in ascending order of deadline in order to execute the task with minimum time constraint first. The cost based tasks are prioritized on the basis of task profit in descending order. This is appreciable as tasks with higher profit can be executed on minimum cost based machine to give maximum profit.

Greedy Allocation

Greedy algorithm is suitable for dynamic heterogeneous resource environment connected to the scheduler through homogeneous communication environment [14].

Greedy approach is one of the approaches used to solve the job scheduling problem.

According to the greedy approach -

“A greedy algorithm always makes the choice that looks best at that moment. That is, it makes a locally optimal choice in the hope that this choice will lead to a globally optimal solution" [15].

The algorithm used here incorporates a similar greedy approach. It takes every job as independent of each other and each of them is scheduled on a resource after making the greedy decision best suited to task requirements and service providers.

Deadline Constraint Based – To improve the completion time of tasks greedy algorithm is used with aim of minimizing the turnaround task of individual tasks, resulting in an overall improvement of completion time.

For each job, the greedy scheduler takes into consideration the following things

i) Arrival time of job
ii) Waiting time at resource
iii) Job size
iv) Processing power of resource
v) Expected waiting and turnaround time

Turnaround time is calculated as

Turnaround Time = Waiting Time of Resource + Task Length/ Proc. Power of Resource

After calculating the turnaround time for each resource, the resource with minimum turnaround time is selected and task is executed there. The scheduler locates the best suited resource that minimizes the turnaround time. The turnaround time is calculated on the basis of expected completion time of a job. Once the scheduler submits a task to a machine, the resource will remain for some time in processing of that job. The resource status is updated to find out when the resource will be available to process a new job.

Thus, minimizing the turnaround time of individual tasks is achieved with this approach which results in an overall improvement in completion time. But, if a large number of tasks have same early deadline, feature of hard and soft deadlines can be introduced to get appropriate results.

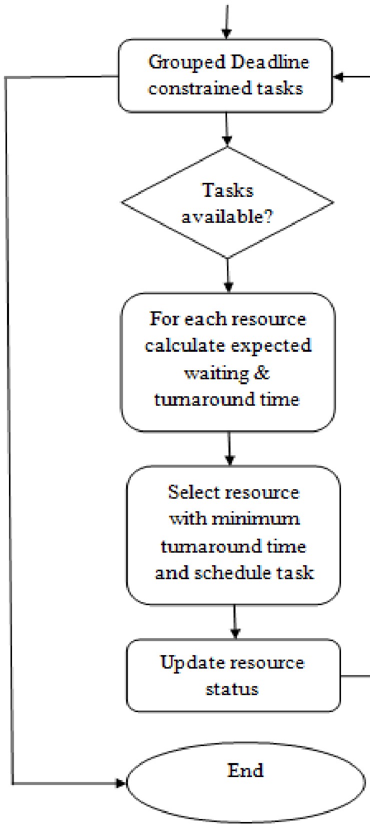

Figure I.1 Scheduling of Deadline Constrained Tasks

Minimum Cost based – To improve the cost of execution of tasks greedy algorithm is used with the aim of minimizing the cost of individual tasks. The overall cost of all the jobs is thus minimized.

For each job, the greedy scheduler takes into consideration the following things before scheduling the task -

1. Cost of resource
2. Job size
3. Processing power of resource

The cost of task at each resource is computed as -

Cost of Task = (Task length / Proc Power of Resource) * Resource Cost

The resource with minimum cost is selected and tasks are scheduled on it until its capacity is supported. After scheduling each task the resource status is updated accordingly. Thus the selection of task and target resource is sequential once they are prioritized according to user needs.

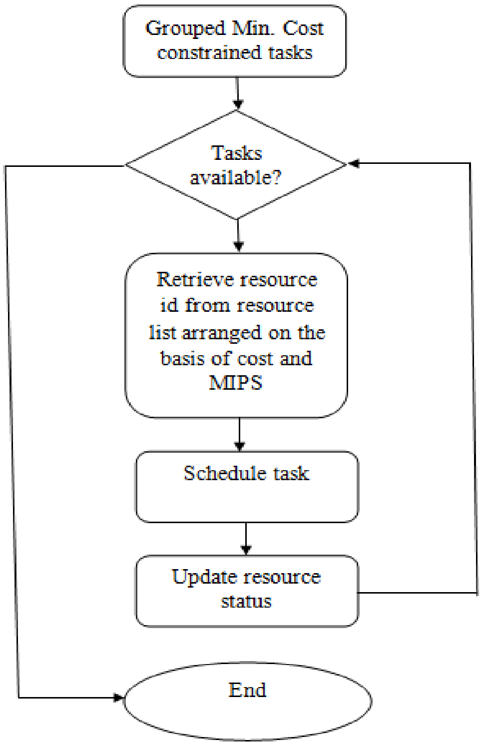

Figure I.2 Scheduling of Cost Based Tasks

Scheduler selects a particular resource and computes the total MI the resource can process (resource MIPS * granularity size). It then schedules next set of on the selected resource till total MI of tasks is less than or equal to resource MI. This process is repeated until all the jobs are grouped and assigned to the resource. Granularity size is the time within which a job is processed at resources.

Salient Features of Modules

The prioritization and greedy approach is beneficial to user while the other task grouping module support the resource provider. The advantages of the modules used are summarized in Table3. 1. The goal of proposed algorithm is to make the most of the advantages of the modules to result in an improved and optimized scheduling algorithm.

Table 3.1 Scheduling Modules with Advantages

Module	Advantages
Task Grouping	Reduced communication cost Easy scheduling and effective resource utilization
Prioritization	Task execution in order of priority Help to achieve better QoS
Greedy Approach	Less waiting time and cost

Proposed Algorithm

A dynamic scheduling algorithm is proposed and implemented in this section. The proposed algorithm works as follows

1. Incoming tasks to the broker are grouped on the basis of their type– deadline constrained or low cost requirement.
2. After initial grouping they are prioritized according to deadline or profit. This is required because the tasks with shorter deadline need to be scheduled first and similarly the tasks resulting in more profit should be scheduled on lost cost machines. Thus, the prioritizing parameter is different based on the nature or type of task.
3. a. For each prioritized task in deadline constrained group –

i) Turnaround time at each resource is calculated taking following parameters into account

- Waiting time
- Task length
- Processing Power of virtual machine

ii) The virtual machine with minimum turnaround time that is capable to execute the task is selected and task is scheduled for execution on that machine.
iii) Waiting time and resource capacity of selected machine are updated accordingly.
b. For cost based group
i) Virtual Machine are selected on the basis of processing power of machine and its cost.
ii) For each virtual machine cloudlets from the group are scheduled till the resource capacity is permitted.
iii) Resource capacity and waiting time are updated accordingly.
Tasks get executed according to their priority with proper resource utilization

REFERENCES

[1] J. Geelan, "Twenty-one experts define cloud computing," Cloud Computing Journal, vol. 2009, pp. 1-5, 2009.
[2] R. Mikkilineni and V. Sarathy. (2009). Cloud Computing and the Lessons from the Past in 18th IEEE International Workshop on Enabling Technologies: Infrastructures for Collaborative Enterprises, WETICE'09,2009.
[3] S. Ma, "A Review on Cloud Computing Development," Journal of Networks, vol. 7, no.2, pp. 305-310, 2012.
[4] P. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, vol. 53, no.6, 2009.
[5] I. Foster, Z. Yong, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," 2008, pp. 1-10.
[6] Cloud Computing[Online]: http://en.wikipedia.org/wiki/Cloud_computing
[8] A. Fox and R. Griffith, "Above the clouds: A Berkeley view of cloud computing," Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS, vol. 28, 2009.
[9] T. J. Velte, et al., Cloud Computing: A Practical Approach: McGraw-Hill Osborne Media, 2009.
[10] V. Hamscher, et al., "Evaluation of job-scheduling strategies for grid computing," Grid Computing—GRID 2000, pp. 191-202, 2000.
[11] Q. Cao, et al., "An optimized algorithm for task scheduling based on activity based costing in cloud computing," 2009, pp. 1-3.
[12] P. J. Wild, et al., "Understanding task grouping strategies," PEOPLE AND COMPUTERS, pp. 3-20, 2004.
[13] N. Muthuvelu, et al., "A dynamic job grouping-based scheduling for deploying applications with fine-grained tasks on global grids," 2005, pp. 41-48.
[14] S. Singh and K. Kant, "Greedy grid scheduling algorithm in dynamic job submission environment," in Emerging Trends in Electrical and Computer Technology (ICETECT), 2011 International Conference on, 2011, pp. 933-936.
[15] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein Introduction to algorithms: The MIT press, 2001.
[16] R. N. Calheiros, et al., "Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services," Arxiv preprint arXiv:0903.2525, 2009.
[17] R. N. Calheiros, et al., "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, pp. 23-50, 2011.