Volume 6, Issue 5, May 2016 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Efficient Scheduling in Federated Cloud Computing Environment Using Dynamic Load Balancing

Priyesh Kanungo
SCSIT, Devi Ahilya University, Indore,
Madhya Pradesh, India

Abstract— Cloud computing services have now become important for both enterprises and personal users for hassle-free, on demand computing resources on anywhere, anytime basis at a reasonable cost with no or little upfront investment. Cloud service provider claim to have virtually infinite computing power and storage with help of virtualization concept in a efficient manner. However, processing large number of jobs in this environment is a complex task and the use of scheduling and resource allocation techniques is necessary to maintain stability and utilize resources efficiently. Dynamic Load balancing is necessary to improve the performance and resource utilization while maintaining stability in the cloud computing environment. This paper highlights the importance of dynamic load balancing in federated cloud architectures and describes various algorithms used for cloud load balancing.

Keywords— Load Balancing, Scheduling Techniques, Cloud Computing, Federated Cloud Architectures, Cloud Operating Systems, Hypervisor.

I. INTRODUCTION

Cloud computing is a distributed computing environment that uses the high speed network and processing and analysis of the data is moved from private PC or servers to the remotely located big data centers owned by the cloud service providers. The services provided by cloud are primary source of computing power for both enterprises and personal computing applications. A cloud environment comprises of networked servers each of which may host multiple virtual machines (VMs). Each VM requires resources like Central Processing Unit, Memory, and storage space apart from high speed networked communication. Cloud has large number of nodes with distributed computing resources placed at many different geographic locations [12].

Although Cloud computing is efficient and scalable but maintaining the stability while processing large number of jobs in the cloud computing environment is a very complex problem and the techniques like dynamic load balancing are receiving attention for researchers to solve this complexity. Dynamic load balancing improves system performance and maintains stability necessary as arrival pattern of jobs is not predictable and the capacities of each node in the cloud differ considerably. Dynamic load balancing plays an important role in cloud scheduling specially in federated cloud architectures, In clouds, dynamic load balancing is used across different data centres to ensure the network availability by minimizing use of computer hardware, software failures and mitigating resource limitations. Load balancing and virtualization technologies, enables cloud resources into an infinite computing resource pool. Load balancing algorithms play a vital role in resource allocation and virtual machine (VM) scheduling by distributing the workload evenly to the whole cloud federation [9].

II. CLOUD SCHEDULING

Cloud computing has attracted large number users to run their applications in the remote data centers. These complex applications also need parallel processing capabilities [10]. Various objectives of cloud computing involves:

- Improved server utilization
- Priority to processes as per users Service Level Agreement
- Improved utilization of resources
- Minimize the completion time of the processes
- Minimizing the context switching and migration time

To achieve the above objectives, following policies should be decided for efficient scheduling:

- Admission control mechanism: The system should accept workload in accordance system's scheduling policies.
- Capacity allocation: Allocation of system's resources for individual services.
- Dynamic load balancing: To evenly distributes the workload among the available servers. This insures not only efficient utilization of processing power but also improves response time of service request.
- Energy optimization minimization of energy consumption: This ensures environmental friendly green computing.
- Quality of service QoS: To satisfy timing or other conditions specified by a Service Level Agreement with the users.

Cloud scheduling may be at host level and at the cloud level. At host level, scheduling is managed by the hypervisor scheduler. The hypervisor can be assumed to be an operating system on physical hardware that manages the virtualized environment. The hypervisor decides about when the virtual machines will get physical resources e.g. Central Processing Unit (CPU), or memory and which processor is assigned to the VM. The Scheduler also decides the placement of each VM based on specific criteria and the servers where each VM is deployed. A VM can support several applications and an application may consist of multiple threads. A scheduling algorithm should be fair and efficient. It should not lead to starvation of processes. In a batch processing systems, it should maximize the throughput whereas in a real time system it must be able to meet the scheduling deadline. Common scheduling algorithms include first come first served, round robin, and shortest process next and priority based algorithms. Real time applications have hard real time constraints, strict timings, and precise amount of resources. These applications use scheduling algorithms like Earliest Deadline First and Rate Monotonic Algorithms. On the other hand, multimedia applications have soft real time constraints and require guaranteed timing constraints [2,11].

In a federated cloud environment, VM can also be deployed on remote server if sufficient resources are not available in the local datacenter. The scheduler can provide dynamic relocation of the VM. A number of different scheduling policies can be used for different clusters for initial placement or dynamic relocation. The users can also specify scheduling constraints. These constraints may be for hardware requirement e.g. CPU and memory, platform, hypervisor location or Service Level Agreement (SLA) which specifies Quality of Service (QoS) requirements of the users etc.

Various scheduling criteria used for cloud scheduling may be:

(i) Load Balancing: Dynamic load balancing improves the utilization of computing resource and provides better response to the processes. Cloud scheduler dynamically distributes processing workload among available servers. It may interact with the virtual machine manager to allocate new VM to lightly loaded server also. The scheduler also interacts with cloud federation manager in order to deploy Virtual Machines to remotely located federation partners' resources in case the local cloud is in overloaded.

(ii) Thermal Balancing: One of the important objectives of cloud computing is promoting energy efficient and environmental friendly green computing. This can be achieved by reducing the cooling requirement of servers. To balance the temperature of the servers, VMs allocated and relocated to the serves based on temperatures.

(iii) Server Consolidation: VMs are allocated to minimum number of servers and also relocated at runtime to reduce the number of servers being used in order to optimize the electricity consumption.

III. DYNAMIC LOAD BALANCING IN CLOUD COMPUTING

Load balancing is the process of distributing the load among various nodes to improve resource utilization and process response time [9]. It ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. The processing workload is transferred from heavily loaded nodes to lightly loaded nodes. The process of dynamic load balancing can be sender initiated, receiver initiated. The load can be processing load in CPU, memory capacity, or network load. Dynamic load balancing algorithms are more effective as compared to static algorithms as they consider present behaviour of the system. Dynamic load balancing is also required to support elasticity in heterogeneous cloud environment so that runtime changes may be adapted easily [9]. The important factors in a load balancing algorithm include load estimation, load information exchange, node selection, systems stability and improving performances [3]. Objectives of Cloud Load Balancing are:

- Efficient resource utilization of resources even when the load distribution is uneven.
- Server consolidation is possible when the load the system is low. This ensures energy efficiency.
- Processes are allocated resources on demand in a way that their response time is improved.

Types of Load Balancing in Cloud: In the following paragraphs, we briefly discuss various load balancing algorithms that can be used in a cloud environment [6,2]:

A. Centralized Load Balancing:

In centralized load balancing, the scheduling decisions are made by a single node. This node stores information about entire cloud network and can use either a static or dynamic approach for load balancing. This technique reduces the time required to analyse different cloud resources but creates substantial overhead on the centralized node. Moreover, the system is not fault tolerant as failure of centralized node may disrupt the whole system [9].

B. Distributed Load Balancing:

In this technique, multiple nodes monitor the network to make appropriate load balancing decisions. Every node in the network maintains information to ensure efficient distribution of tasks. In distributed scenario, the system is fault tolerant as disruption of a single node doesn't affect the system performance. It also compares them on the basis of spatial distribution of nodes.

C. Hierarchical or Partition Load Balancing:

Different cloud levels are involved in load balancing decisions. A tree is used where every node is balanced under the control of its parent. Scheduling decisions are based on load information provided by the parent node. Three-phase hierarchical scheduling has multiple phases of scheduling. Request monitor acts as a head of the network and is responsible for monitoring service manager which in turn monitor service nodes. First phase uses BTO (Best Task Order)

scheduling, second phase uses EOLB (Enhanced Opportunistic Load Balancing) scheduling and third phase uses EMM (Enhanced Min-Min) scheduling [9, 5, 7].

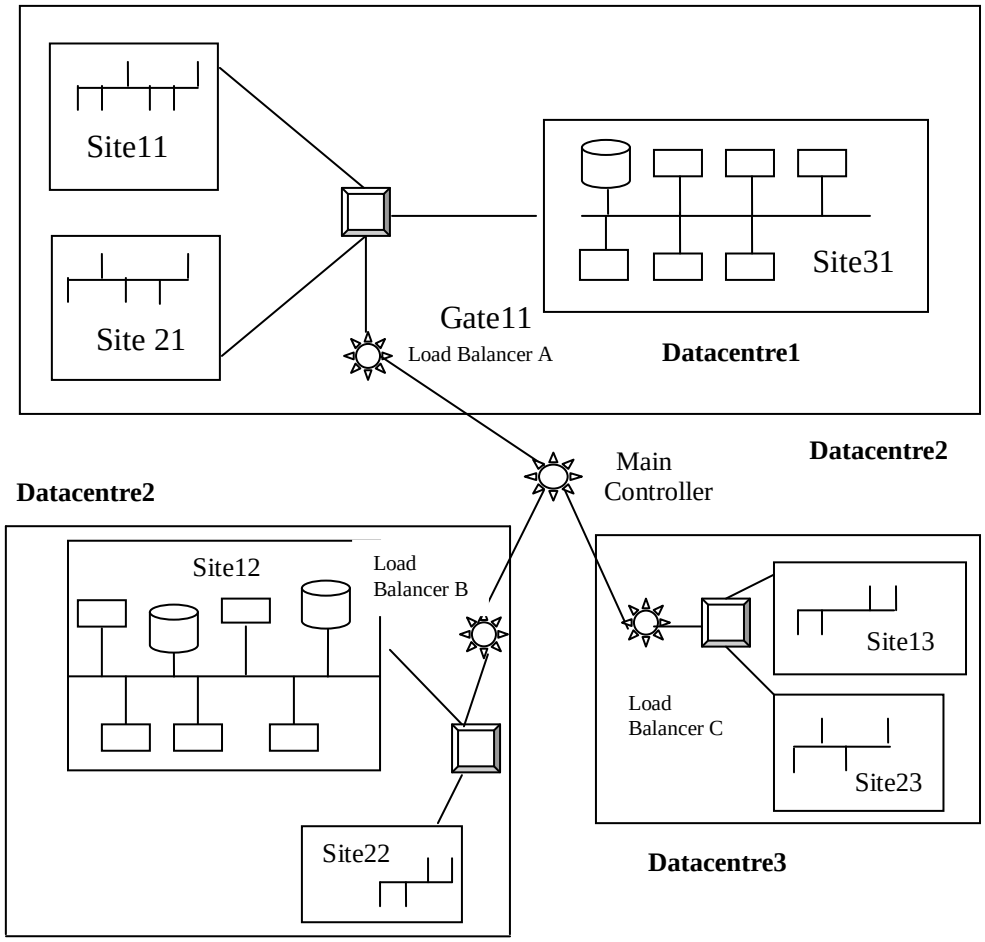

Fig 1 Cloud Partition Load Balancing

Cloud Partition Load Balancing Strategy: Cloud partitioning is hierarchical partition and used to manage a large public cloud that includes a number of nodes in different geographical locations. Partition is a subarea of the public cloud having divisions on the basis the geographic locations. Steps in Cloud Partition Load Balancing include [4]:

(i) Start
(ii) Job Arrives at the main controller
(iii) Choose cloud partition
(iv) If partition is lightly loaded then
{Job arrives at the partition
Assign the job to a node according to a DLB policy
}
Else
(v) goto Step 3 (Choose another partition)

Important considerations in partitioned load balancing include:

- Cloud division rules: Cloud division is not a simple problem and a detailed cloud division methodology is needed to be worked out e.g. on the basis of geographic location of data centres.
- Load balance strategy within a cluster: Statistical tests are needed to compare various load balancing strategies to be used within a cluster.
- Load status information: An improved and comprehensive technique is needed to calculate the load level of a cloud partition i.e. whether the partition is overloaded or under loaded.
- Deciding the refresh period: If the period of data statistics by main controller and the cloud partition Balancers is too short, the high overhead time in collecting the information will degrade the system performance. If the refreshing period is too long, the information will not be useful in making efficient scheduling decisions.

D. Honeybee Foraging Algorithm:

Honeybee foraging is used for load balancing in a complex distributed environment. In this algorithm, movement of honey bees in search of food forms the basis of dynamic load balancing in cloud computing environment. This algorithm is inspired from the behaviour of honey bees for finding and reaping food. Upon finding food sources, forager bees, come back to the beehive to signal this using a waggle dance to gives the idea of the quantity of food and its distance from the

beehive. Scout bees then follow the foragers to this location and then began to reap the food. Then they return to the beehive waggle dance to give idea of how much food is left for further perusal of the food source [1].

In a cloud, the demand of web servers' increase or decrease with the changing demands of the user. Using dynamic load balancing, the services may be assigned dynamically [9]. The servers are grouped as a virtual server with every virtual server having its own queue, calculates a profit or reward, which is analogous to the quality that the bees show in their waggle dance when they find food. The measure of this reward can be the time spent by the CPUs on the processing of a request. The dance floor in case of honey bees is analogous to an advert board here. This board is also used to advertise the profit of the entire colony. Each of the servers takes the role of either a forager or a scout. The server after processing a request can post their profit on the advert boards with a probability.

A server serving a request, calculates its profit and compares it with the colony profit. If his profit is high, then the server stays at the current virtual server; posting an advertisement for it by. If the profit is low, then the server returns to the forage or scout behaviour [8].

E. Biased Random Sampling:

Biased random sampling is distributed load balancing technique which uses virtual graph for required information. That is a virtual graph, with the connectivity of each graph node (a server) representing the load on the server. Each node in the graph has in-degree directed to the free resources of the server. When a node executes a process, it deletes an incoming edge, which indicates reduction in the availability of free resource. After process completion, the node creates an incoming edge, which indicates an increase in the availability of free resource. Addition and deletion of processes is done using random sampling.

The walk starts at any of the nodes and at every step a neighbor is chosen randomly. The last node is selected for allocation of load. The selection of a node for load allocation may also be based on certain criteria like efficiency, load status viz. underloaded i.e. having highest in degree etc.. If walk length increases, the efficiency of load allocation also increases. Threshold value of walk-length b is equal to log n. A node on receiving a request, will execute it only if its current walk length is equal to or greater than the threshold value. Otherwise, the walk length of the job is incremented and another neighboring node is selected at random. On the completion of execution of a process, an incoming edge of that node is deleted. Finally what we get is a directed graph. The load balancing scheme used here is decentralized, therefore, this algorithm is suitable for federated cloud architecture.

F. Load Balancing Based on Task Dependencies:

When the execution tasks are dependent on one or more sub-tasks, they can be executed only after completion of the sub-tasks. Therefore, scheduling of such task before sub-tasks is in-efficient. Task dependency is modelled using workflow based algorithms. Workflow basically uses Directed Acyclic Graph (DAG) to represent dependency. Different workflow based algorithms are designed keeping in mind whether single or multiple workflows are to be modelled or single or multiple QoS parameters are to be maintained. Different workflows with or without completely different structure are termed as multiple workflows. Workflows can also be classified as Transaction Incentive and Data Incentive. Transaction Incentive workflow are multiple instances of one workflow that have same structure. In data incentive workflows (size and quantity of data is large.

IV. CONCLUSIONS

Maintaining the stability while processing large number of jobs in the cloud computing environment is a very complex problem and the techniques like load balancing are receiving attention for researchers to solve this complexity. As Scheduling is an important issue in cloud computing to improve the server performance and resource utilization. Load balancing techniques help to take efficient scheduling decisions in a cloud computing environment. In this paper, we have highlighted the importance of dynamic load balancing in cloud computing environment and discussed various algorithms which can be used in different situations for effective scheduling.

REFERENCES

[1] Zehua Zhang and Xuejie Zhang, A Load Balancing Mechanism Based on Ant Colony and Complex Network Theory in Open Cloud Computing Federation retrieved from www.researchgate.net /profile/Zehua_Zhang2/publication

[2] Siva Theja, Maguluri and R. Srikant, Stochastic Models of Load Balancing and Scheduling in Cloud Computing Clusters Retrived from www.ideals.illinois.edu.

[3] Y. 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 2010 IEEE 3rd International Conference on Cloud Computing , 2010, pp. 91–98

[4] Gaochao Xu, Junjie Pang, and Xiaodong Fu A Load Balancing Model Based on Cloud Partitioning for the Public Cloud TSINGHU A SCIENCE AND TECHNOLOGY pp34-39 Volume 18, Number 1, February 2013.

[5] Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C. Mcdermid, Availability and load balancing in cloud computing, presented at the 2011 International Conference on Computer and Software Modeling, Singapore, 2011.

[6] M. Randles, D. Lamb, and A. Taleb-Bendiab, A comparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on Advanced Information Networking and Applications , Perth, Australia, 2010, pp. 551-556.

[7] Hu-Ching, W., Yan, K. Q., Sheng, S. & Wei, W. C. (2011). A three-phases scheduling in a hierarchical cloud computing network. 2011 Third International Conference on Communications and Mobile Computing 978-0-7695-4357-4/11 $26.00 © 2011 IEEE DOI 10.1109/CMC.2011.2.

[8] Mayank Katyal, Atul Mishra. A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment A. International Journal of Distributed and Cloud Computing Volume 1 Issue 2 December 2013pg 5-11.

[9] Sharma, Rajkumar, and Priyesh Kanungo. "Dynamic Load Balancing Algorithm for Heterogeneous Multi-core Processors Cluster." Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies. IEEE Computer Society, 2014.

[10] Mehta H., Kanungo P. and Chandwani M., "EcoGrid: A Dynamically Configurable Simulation Environment for Economy-Based Grid Scheduling Algorithms," 3rd ACM Annual Conference Compute-2011, Bangalore, March 25-26, 2011.

[11] Mehta, Hemant Kumar, Manohar Chandwani, and Priyesh Kanungo. "Performance evaluation of Grid simulators using profilers." 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE).

[12] Mehta, H., Kanungo, P. and Chandwani, M., "Towards Development of a Distributed e-Learning EcoSystem," 2nd International Conference on Technology for Education (TforE-2010), IIT Mumbai, July 1-3, 2010.

