INTERNATIONAL JOURNAL OF RESEARCH IN COMPUTER APPLICATIONS AND ROBOTICS
ISSN 2320-7345

A SURVEY OF VARIOUS LOAD BALANCING AND TASK SCHEDULING TECHNIQUES IN CLOUD COMPUTING

S Chithra[1], Dr J Anitha[2]

[1]PG Scholar, Computer Science and Engineering Department, Sri Ramakrishna Engineering College, Coimbatore.
[2]Associate Professor, Department of Information Technology, Sri Ramakrishna Engineering College, Coimbatore.

Abstract: - Cloud computing is emerging as a new paradigm for manipulating, configuring, and accessing large scale distributed computing applications over the network. Load balancing is one of the main Challenges in cloud computing which is required to distribute the workload evenly across all the nodes. Properly dispatching tasks among CPU cores is crucial to reduce response time of jobs, which provides benefit for both systems. Load balancing is a main challenge in cloud environment. Load balancing is helped to distribute the dynamic workload across multiple nodes to ensure that no single node is overloaded. It helps in proper utilization of resources. It also improves the performance of the system. Many existing algorithms provide load balancing and better resource utilization. There are various types load are possible in and survey was made in cloud computing.

Index Terms: Cloud Computing, Task scheduling, Load Balancer, Load Balancing, Load Balancing algorithm.

1. Introduction

The term "cloud computing" is everywhere. In the simplest terms, cloud computing means storing and accessing data and programs over the Internet instead of your computer's hard drive[1].Cloud provides the services by deploying virtual machines (VMs) in their data centre. Cloud computing and storage solutions provide users and enterprises with various capabilities to store and process their data in third-party data centres. Cloud providers are facing rapidly increasing traffic loads; they must have proper expansion strategies for their ultra-scale data centres [3][2]. It relies on sharing of resources to achieve coherence and economies of scale, similar to a utility like the grid over a network.

1.1 Types of cloud

There are four types of cloud deployment model

- Public cloud
- Private cloud
- Hybrid cloud
- Community cloud

1.1.1 Public cloud The Public Cloud allows systems and services to be easily accessible to the general public. Public cloud may be less secure because of its openness, e.g., e-mail.

1.1.2 Private cloud The Private Cloud allows systems and services to be accessible within an organization. It offers increased security because of its private nature

1.1.3 Hybrid cloud The Hybrid Cloud is mixture of public and private cloud. However, the critical activities are performed using private cloud while the non-critical activities are performed using public cloud**.**

1.1.4 Community cloud The Community Cloud allows systems and services to be accessible by group of organizations

1.2 Types of service models

Service Models are the reference models on which the Cloud Computing is based. These can be categorized into three basic service models as listed below:

1. Infrastructure as a Service (IaaS)

2. Platform as a Service (PaaS)

3. Software as a Service (SaaS)

1.2.1 Infrastructure-as-a-Service (IaaS) Infrastructure-as-a-Service is the first layer and foundation of cloud computing. Using this service model, you manage your applications, data, operating system, middleware and runtime. The service provider manages your virtualization, servers, networking and storage.

1.2.2 Platform-as-a-Service (PaaS) This cloud service model could be considered the second layer. It manages our applications and data and the cloud vendor manages everything, Benefits for using Platform-as-a-Service include streamlined version deployment and the ability to change or upgrade and minimize expenses.

1.2.3 Software-as-a-Service (SaaS) this is the final layer of the cloud services model. This allows our business to run programs in the cloud where all portions are managed by the cloud vendor. Examples of this are online banking and email such as Gmail and Hotmail.

This survey was formed as following sections, Section 1 includes the introduction, and section 2 and 3 includes load balancing and task scheduling concepts, and section 4 compares various techniques, and section 5 includes challenges, and section 6 draws the conclusion.

2. Load balancing in cloud computing

Load balancing is the process of distributing the load among various resources in any system. Thus load need to be distributed over the resources in cloud-based architecture, so that each resources does approximately the equal amount of task at any point of time. Basic need is to provide some techniques to balance requests to provide the solution of the application faster.

Cloud vendors are based on automatic load balancing services, which allow clients to increase the number of CPUs or memories for their resources to scale with increased demands. This service is optional and depends on the clients business needs. So load balancing serves two important needs, primarily to promote availability of Cloud resources and secondarily to promote performance. In order to balance the requests of the resources it is important to recognize a few major goals of load balancing algorithms:

a) Cost effectiveness*:* primary aim is to achieve an overall improvement in system performance at a reasonable cost.

b) Scalability and flexibility: the distributed system in which the algorithm is implemented may change in size or topology. So the algorithm must be scalable and flexible enough to allow such changes to be handled easily.

c) Priority: prioritization of the resources or jobs need to be done on beforehand through the algorithm itself for better service to the important or high prioritized jobs in spite of equal service provision for all the jobs regardless of their origin.

2.1 Load balancing classification: Represents different load balancing algorithms. This is mainly divided into two categories: static load balancing algorithm and dynamic load balancing algorithm:

1) Static approach: This approach is mainly defined in the design or implementation of the system. Static load balancing algorithms divide the traffic equivalently between all servers.

2) Dynamic approach: This approach considered only the current state of the system during load balancing decisions. Dynamic approach is more suitable for widely distributed systems such as cloud computing. Brief reviews of few existing load balancing algorithms are presented in the following:

a) Token Routing*:* The main objective of the algorithm is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents cannot have the enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbour's working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. so in this approach no communication is overhead.

b) Round Robin*:* In this algorithm [7], the processes are divided between all processors. Each process is assigned to the processor in a round robin order. The process allocation order is maintained locally independent of the allocations from remote processors. Though the work load distributions between processors are equal but the job processing time for different processes are not same. So at any point of time some nodes may be heavily loaded and others remain idle. This algorithm is mostly used in web servers where Http requests are of similar nature and distributed equally.

c) Randomized*:* Randomized algorithm [9] is of type static in nature. In this algorithm a process can be handled by a particular node n with a probability p. The process allocation order is maintained for each processor independent of allocation from remote processor. This algorithm works well in case of processes are of equal loaded. However, problem arises when loads are of different computational complexities. Randomized algorithm does not maintain deterministic approach. It works well when Round Robin algorithm generates overhead for process queue.

d) Central queuing*:* This algorithm [4] works on the principal of dynamic distribution. Each new activity arriving at the queue manager is inserted into the queue. When request for an activity is received by the queue manager it removes the first activity from the queue and sends it to the requester. If no ready activity is present in the queue the request is buffered, until a new activity is available. But in case new activity comes to the queue while there are unanswered requests in the queue the first such request is removed from the queue and new activity is assigned to it. When a processor load falls under the threshold then the local load manager sends a request for the new activity to the central load manager. The central manager then answers the request if ready activity is found otherwise queues the request until new activity arrives.

e) Connection mechanism*:* Load balancing algorithm [12] can also be based on least connection mechanism which is a part of dynamic scheduling algorithm. It needs to count the number of connections for each server dynamically to estimate the load. The load balancer records the connection number of each server. The number of connection increases when a new connection is dispatched to it, and decreases the number when connection finishes or timeout happens.

f) Equally Spread Current Execution Algorithm*:* Equally spread current execution algorithm [4] process handle with priorities. it distribute the load randomly by checking the size and transfer the load to that virtual machine which is lightly loaded or handle that task easy and take less time, and give maximize throughput. It is spread spectrum technique in which the load balancer spread the load of the job in hand into multiple virtual machines.

g) Throttled Load Balancing Algorithm: Throttled algorithm is completely based on virtual machine [11]. In this client first requesting the load balancer to check the right virtual machine which access that load easily and perform the operations which is given by the client or user. In this algorithm the client first requests the load balancer to find a suitable Virtual Machine to perform the required operation.

3. Task Scheduling

Task scheduling is done after the resources are allocated to all cloud entities. Scheduling defines the manner in which different entities are provisioned. Resource provisioning defines which resource will be available to meet user requirements whereas task scheduling defines the manner in which the allocated resource is available to the end user (i.e. whether the resource is fully available until task completion or is available on sharing basis). Task scheduling provides “Multiprogramming Capabilities” in cloud computing environment. Task scheduling can be done in two modes:

a. Space shared

b. Time shared

Both hosts and VM can be provisioned to users either in space shared mode or time shared mode. In space sharing mode resources are allocated until task does not undergo complete execution (i.e. resources are not pre-empted); whereas in time sharing mode resources are continuously pre-empted till task undergoes completion.

3.1 A Task Scheduling Algorithm Based on Load Balancing:

Two-level task scheduling mechanism based on load balancing to meet dynamic requirements of users and obtain high resource utilization. It achieves load balancing by first mapping tasks to virtual machines and then virtual machines to host resources thereby improving the task response time, resource utilization and overall performance of the cloud computing environment.

a) Biased Random Sampling: A distributed and scalable load balancing approach that uses random sampling of the system domain to achieve self-organization thus balancing the load across all nodes of the system. Here a virtual graph is constructed, with the connectivity of each node (a server is treated as a node) representing the load on the server. Each server is symbolized as a node in the graph, with each in degree directed to the free resources of the server. The load balancing scheme used here is fully decentralized, thus making it apt for large network systems like that in a cloud. The performance is degraded with an increase in population diversity.

b) Min-Min Algorithm: It begins with a set of all unassigned tasks [9]. First of all, minimum completion time for all tasks is found. Then among these minimum times the minimum value is selected which is the minimum time among all the tasks on any resources. Then according to that minimum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines. Then again the same procedure is followed until all the tasks are assigned on the resources. But this approach has a major drawback that it can lead to starvation.

c) Max-Min Algorithm: Max-Min is almost same as the min-min algorithm [9] except the following: after finding out minimum execution times, the maximum value is selected which is the maximum time among all the tasks on any resources. Then according to that maximum time, the task is scheduled on the corresponding machine. Then the execution time for all other tasks is updated on that machine by adding the execution time of the assigned task to the execution times of other tasks on that machine and assigned task is removed from the list of the tasks that are to be assigned to the machines.

d) Token Routing: The main objective of the algorithm [13] is to minimize the system cost by moving the tokens around the system. But in a scalable cloud system agents cannot have the enough information of distributing the work load due to communication bottleneck. So the workload distribution among the agents is not fixed. The drawback of the token routing algorithm can be removed with the help of heuristic approach of token based load balancing. This algorithm provides the fast and efficient routing decision. In this algorithm agent does not need to have an idea of the complete knowledge of their global state and neighbour’s working load. To make their decision where to pass the token they actually build their own knowledge base. This knowledge base is actually derived from the previously received tokens. So in this approach no overhead is generated.

4. Comparison table for various load balancing and task scheduling techniques:

Title	*Content*	*Advantage*	*Disadvantage*
The Power of Two Choices in Randomized Load Balancing	Supermarket model is introduced and task to be scheduled, probing two slaves randomly and dispatching the task to the one with the least load.	Easier to analyze because its behaviour is completely deterministic	The supermarket model proves difficult to analyze dependencies task. Reduce the system performance.
Gang scheduling in multi-core clusters implementing migrations	The performance of gang scheduling algorithms for homogeneous and heterogeneous clusters which consist of multi-core processors is evaluated. The gang scheduling considers the inter-communication of jobs	Improve the system performance. Less time for resources allocation.	Increase the communication overhead between masters and slaves.
Delay Scheduling: Achieving Locality and Fairness in Cluster Scheduling	Two approaches to reassigning resources are introduced: killing tasks from existing jobs to make room for new jobs, and waiting for tasks to finish to assign slots to new jobs. The delay scheduling is designed for resources related task	Improve response time Double throughput in an IO-heavy workload	High latency, low applicability.
Randomized Load Balancing with General Service Time Distributions	Randomized load balancing, where a job is assigned to a server from a small subset of randomly chosen servers, is very simple to implement	Good performance, Reducing collisions and waiting time.	load balancing is a problem to distribute tasks among multiple resources, Increase overhead communication.

5. Challenges in load balancing

1**. **Throughput This metric is used to estimate the total number of tasks, whose execution has been completed successfully. High throughput is necessary for overall system performance.

2**. **Overhead Overhead associated with any load balancing algorithm indicates the extra cost involved in implementing the algorithm. It should be as low as possible.

3. Fault Tolerance It measures the capability of an algorithm to perform uniform load balancing in case of any failure. A good load balancing algorithm must be highly fault tolerable.

4. Migration Time is defined as, the total time required in migrating the jobs or resources from one node to another. It should be minimized.

5. Response Time It can be measured as, the time interval between sending a request and receiving its response. It should be minimized to boost the overall performance.

6. Resource Utilization It is used to ensure the proper utilization of all those resources, which comprised the whole system. This factor must be optimized to have an efficient load balancing algorithm.

7. Scalability It is the ability of an algorithm to perform uniform load balancing in a system with the increase in the number of nodes, according to the requirements. Algorithm with higher scalability is preferred.

8. Performance It is used to check, how efficient the system is. This has to be improved at a reasonable cost, e.g., reducing the response time though keeping the acceptable delays.

6. CONCLUSION:

Cloud computing provides everything to the user as a service over network. The major issues of cloud computing is Load Balancing. Overloading of a system may lead to poor performance which can make the technology unsuccessful, for the efficient utilization of resources; the efficient load balancing algorithm is required. In this paper, we have surveyed various load balancing algorithms in the Cloud environment. We have discussed the already proposed algorithms by various researchers. The various load balancing algorithms are also being compared here on the basis of different types of parameter.

REFERENCES

[1] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing", EECS Department, University of California, Berkeley, Technical Report No., UCB/EECS-2009-28, pages 1-23, February 2009.

[2] R. W. Lucky, "Cloud computing", IEEE Journal of Spectrum, Vol. 46, No. 5, May 2009, pages 27-45.

[3] M. D. Dikaiakos, G. Pallis, D. Katsa, P. Mehra, and A. Vakali, "Cloud Computing: Distributed Internet Computing for IT and Scientific Research", IEEE Journal of Internet Computing, Vol. 13, No. 5, September/October 2009, pages 10-13.

[4] G. Pallis, "Cloud Computing: The New Frontier of Internet Computing", IEEE Journal of Internet Computing, Vol. 14, No. 5, September/October 2010, pages 70-73.

[5] B. P. Rima, E. Choi, and I. Lumb, "A Taxonomy and Survey of Cloud Computing Systems", Proceedings of 5th IEEE International Joint Conference on INC, IMS and IDC, Seoul, Korea, August 2009, pages 44-51.

[6] A. M. Alakeel, "A Guide to dynamic Load balancing in Distributed Computer Systems", International Journal of Computer Science and Network Security (IJCSNS), Vol. 10, No. 6, June 2010, pages 153-160.

[7] ZhongXu,RongHuang,(2009)"Performance Study of Load Balancing Algorithms in Distributed Web Server Systems", CS213 Parallel and Distributed Processing Project Report.

[8] P.Warstein, H.Situ and Z.Huang(2010), "Load balancing in a cluster computer" In proceeding of the seventh International Conference on Parallel and Distributed Computing, Applications and Technologies, IEEE.

[9] Ms.NITIKA, Ms.SHAVETA, Mr. GAURAV RAJ; "Comparative Analysis of Load Balancing Algorithms in Cloud Computing", International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 3, May 2012.

[10] Y. Fang, F. Wang, and J. Ge, "A Task Scheduling Algorithm Based on Load Balancing in Cloud Computing", Web Information Systems and Mining, Lecture Notes in Computer Science, Vol. 6318, 2010, pages 271-277.

[11] T.R.V. Anandharajan, Dr. M.A. Bhagyaveni" Co-operative Scheduled Energy Aware Load-Balancing technique for an Efficient Computational Cloud" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 2, March 2011.

[12] T. Kokilavani J.J. College of Engineering & Technology and Research Scholar, Bharathiar University, Tamil Nadu, India" Load Balanced Min-Min Algorithm for Static Meta-Task Scheduling in Grid Computing" International Journal of Computer Applications (0975 – 8887) Volume 20– No.2, April 2011.

[13] Peter Mell, Timothy Grance, "The NIST Definition of Cloud Computing", NIST Special Publication 800-145, September 2011.

[14] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh, Christopher Mcdermid (2011)"Availabity and Load Balancing in Cloud Computing" International Conference on Computer and Software Modelling IPCSIT vol.14 IACSIT Press, Singapore 2011.

[15] [15] A. Khiyaita, M. Zbakh, H. El Bakkali and Dafir El Kettani, "Load Balancing Cloud Computing: State of Art" , 9778-1- 4673-1053-6/12/$31.00, 2012 IEEE. [16] Wayne Jansen Timothy Grance" Guidelines on Security and Privacy in Public Cloud Computing" NIST Draft Special Publication 800-144.