ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Load Balancing in Cloud Computing: A Review

Akanksha Mathur, Virender Singh Shekhawat

Assistant Professor, Dept. of C.S.E, Govt. College of Engineering and Technology, Bikaner, India

Professor, Dept. of C.S.E, Birla Institute of Technology and Science, Pilani, India

ABSTRACT: Cloud computing in a nutshell provides on-demand access to visualized IT resources that can be shared by others on "pay-as-use" policy. It is an awesome platform in next stage of evolution of internet that leverages various opportunities to improve the way in which we think about and implement the practices and technology needed to secure the things that matters us the most. With the recent advent of technology, it has revolutionized the information technology industry by enabling elastic on-demand provisioning of computing resources. Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory, capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resource utilization and job response time by avoiding a situation where some of the nodes are heavily loaded while other under -loaded or nodes are idle. Load balancing ensures that all the processor in the system or every node in the network distributes equal amount of work at any instant of time. Technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types) for balancing load and can also be categorized static or dynamically. This paper is a brief discussion on different load balancing techniques and comparison between them.

KEYWORDS: Cloud Computing; Load Balancing

I INTRODUCTION

Cloud computing can be defined as 'a type of parallel and distributed system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned, and as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers'. There are several concepts for computational system, one of which is load balancing. Load balancing methodologies ensures that all processor in the system or every node for executing task to distribute workload equally in cloud environment. The goal is to improve the utilization of computing resources and reduce energy consumption under workload independent quality of service constraints. Also several techniques have been proposed to reduce the downtime of the VM transferred, at the expense of the total migration time, response time or data processing time. Energy consumption is reduced by dynamically deactivating and reactivating physical nodes to meet the current resource demand. For gaining this purpose load balancing implemented. This paper presents study and comparison between them.

II LOAD BALANCING

Cloud computing is causing reassess and gradual change in how IT is consumed in cloud computing where client request is fetched upon to provide services as per requirements is important tasks which is achieved by proper load balancing over the servers. Load parameters depend on CPU, network delay, memory, bandwidth, resources availability etc. Load balancing is a simple technique that ensures that no system left overloaded or under-loaded in different parameters and specifications. Load balancing focuses on maximum throughput, optimizing resource usage, avoid overloading and minimize response time, reduces network latency. Load balancing takes account of two things, one is the resource provisioning or resource allocation and other is task scheduling in distributed environment. An efficient provisioning of resources and scheduling of resources as well as tasks will ensure:

A.) Resources are easily available on demand.
B.)Resources are efficiently utilised under over-loaded and under-loaded conditions.
C.)Energy is saved in case of low load (i.e. when usage of cloud resources is below certain threshold).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

D.)Cost of using resources is reduced.
Load balancing can be categorised in 2 forms:

Depending on who initiate the process, load balancing algorithms can be of three categories as given in:
A.) Sender Initiated: If it is initiated by the sender.
B.) Receiver Initiated: If it is initiated by the receiver.
C.) Symmetric: It is the hybrid technique or combination of both sender and receiver initiated.

Depending on present state of the system, load balancing algorithms can be divided into 2 categories as given in:
Static and dynamic approaches:
In static approach requires prerequisite knowledge of nodes capacity, processing power memory, performance statistics of user requirements before getting executed and load is distributed equally. In dynamic approach resources are flexible in dynamic environment. In this scenario cloud cannot rely on prior knowledge whereas it accounts of run time statistics accordingly.

A. Metrics for load balancing

Nature: It determines the behaviour of algorithms either static or dynamic.

Overhead: It determines the implementation details of algorithm like inter-process communication, migration of tasks etc. Also this should be minimised so that algorithm can work efficiently.

Throughput: It is number of process that completes its execution per time unit which should be maximised for better performance.

Process migration: It determines the migration of tasks or resources from one node to another which should be minimised as it enhances the performance of the system.

Response time: It is time to compute or execute any task which should be minimised for better efficiency.

Resource utilisation: The proportion of the available time (expressed usually as a percentage) that a system or resources is operating which should be optimised for better performance.

Fault tolerant: The ability of a system to respond gracefully to an unexpected hardware or software failure.

Waiting time: It is the amount of time process takes while in ready queue. It should be minimised for system for better performance.

Scalability: It is the capability of a system network or a process to handle growing amount of work in a capable manner or its ability to be enlarged to accommodate that growth.

Performance: The completion of a given task measured against accuracy, completeness cost, speed etc. to check efficiency of any system.

III. RELATED WORK

In the recent time various load balancing algorithms developed that are efficient enough that resources are utilized equal load balancing in the system.

In 2011, T. kokilavani [13] proposed “Load balance Min-min” (LBMM) algorithm is a grid scheduling algorithm. It executes Min-Min in the first round. In the other round it chooses the resources with heavy load and reassigns them to the resources with light load. LBMM. It identifies the resources with high make span and then selects the task with

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

minimum execution time on that resource. Then the completion time is compared with make span produced by Min-Min, if it is less than task is rescheduled otherwise next maximum completion time of task is selected and loop continues and stop until all resources and tasks are fully utilised. But it is simple and produces a good make span compared to other algorithms. This increases load balancing but this also encounters the fact that when the number of the small tasks is more than the number of the large tasks in a meta-task, the Min-Min algorithm cannot schedule tasks, appropriately, and the make span of the system gets relatively large.

In 2012, O.M Elzeki [2] suggested improved "Max-Min" algorithm to increase Max- Min efficiency by concurrent execution of task as resources and focuses on selecting task with maximum completion time. The algorithm calculates the expected completion time of the submitted tasks on each resource. Then the expected execution time is assigned to a resource that has the minimum overall completion time. Finally, this scheduled task is removed from meta-tasks and all calculated times are updated and the processing is repeated until all submitted tasks are executed. The algorithm focuses on minimising the total make span which is the total complete time in large distributed environment. The proposed algorithm produces mapping schema similar to RASA in such concurrency executing tasks and minimisation of total completion time required to finish all tasks Although time complexity of developed algorithm is same as the previous one O(MN^ 2) and same execution time but produces better make span with more reliable scheduling allows concurrent execution of tasks.

In 2012, Ratan Mishra [4] introduced "Ant colony optimisation" (ACO) to avoid deadlock condition in cloud. The implementation carried out on two different job scheduling strategies i.e. time shared and space shared. According to obtained experimental result it consumed less memory during processing of tasks as previously implemented resources and provided high performance.

In 2014,Ekta Gupta[20] proposed another ACO algorithm, a head node in such a manner that it has the maximum number of neighboring nodes which helps ants to traverse in most possible directions of the network. The ants originate from head node continuously. There is limited number of ants to avoid the congestion of network and a suicide timer is set on on each ant, which when reaches zero the ant will halt itself. The selection of timer value would depend on the size and number of nodes in the network. These ants traverse the width and length of the network in such a way that they know about the location of under loaded or overloaded nodes in the network. These ants traversal will be updating a pheromone table, which will keep a tab on the resources utilisation by each node. This algorithm gives efficient use of resources.

In 2013, Manan D.Shah [9] proposed "Throttled algorithm" that handles request according to matching configuration of virtual machine. Also adds VM Id to user if configuration matches and if does not match then disclose some relevant virtual machines with configuration to provide better services.

Also in 2013, Shridhar G.Domanal and G.Ram Mohana Reddy [21] suggested a "Modified throttled algorithm" which maintains an index table of virtual machines and also the state of VMs similar to the throttled algorithm. This algorithm attempts to improve the response time and achieve efficient usage of available virtual machines. The algorithm employs a method for selecting a VM for processing client's request where, VM at first index is initially selected depending upon the state of the VM. If the VM found successfully, it is assigned with the request and id of VM is returned to data center, else -1 is returned. When the next request arrives, VM at next index to already assigned VM is chosen depending on the state of VM and follows the above step, unlike of the basic throttled algorithm, where the index table is parsed from the initial index every time the data center queries load balancer for allocation of VM.

In 2013, Kousik Dasgupta[3] proposed "Genetic algorithm"(GA) for efficient utilisation of resources and also guarantees QOS. In this randomly population of processing unit is initialised first and encoded them into binary strings. Then fitness value of each population is evaluated in crossover step followed by mutation where small value is picked as mutation probability and this GA process is repeated till either the fittest chromosome (optimal solution) is found or the termination condition(maximum number of iteration) is exceeded. This paper compares GA with three commonly used scheduling algorithms First come first serve (FCFS), Round robin (RR), Scholastic hill climbing (SHC). The merit of developed strategy has linear search capability to larger extend and is applicable to complex objective function and can avoid being trapping into local optimal solution. The complexity analysis of any algorithm includes computation time complexity analysis and space complexity analysis. Thus it is robust as compared with other three algorithms.

In 2010, Randles M [10] introduced load balancing concept describing interaction of nodes by cooperative or non-cooperative manner. It is degraded with a growth in system diversity. "Active clustering" embrace to provide grouping of similar nodes together by efficient use of resources, thereby increase throughput and performance of the system. In this match-making process takes place and group similar nodes when processes initiated and iterative

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

process continues in system till the process halts thus providing load balancing efficiently. The performance is thus increased with high availability of resources which further increases throughput.

In 2013, Elina Pacini [13] proposed a cloud VM scheduler based on "Particle swarm optimisation" (PSO). In this algorithm all hosts of cloud are regarded as swarm and each host in cloud is particle in the swarm. In this each iteration searching of host is performed and velocity difference is compared with neighbouring host. If any of the hosts nearby has a lower load than the original host, then the VM is moved to the neighbour host with a greater velocity. Additionally keeping information that the particles move through hosts of their neighbourhood in search of a host with the lower load and reaches up quickly. Therefore each particle makes a move to one of its neighbours, which has the minimum load among all. If all its neighbour are busier than the host itself, the VM is not moved from the current host. All particles move to the minimum load and eventually at the end particle delivered associated VM to the host with low load among neighbouring host and task end. Since each move that a particle performs, involves moving through the network, to minimise the number of moves: every time a particle moves to a neighbouring host with no allocated VM. The particle allocates its associated VM to it directly without performing further steps. The number of messages sent over the network by a particle to their neighbours hosts to obtain information regarding their availability load is accumulated in the network messages variable.

In 2014, Stuti-Dave et-al [1] presented a "Round Robin" (RR) for load balancing at virtualized environment. In this paper they have suggested improved Fair RR algorithm approach that provides dynamic time quantum strategy. When the request in enters ready queue, they are processed and calculated according to time quantum and burst time computation while VM's are allocated. Thus FRR provide fairness to larger and smaller incoming requests at executing load resulting in faster load balancing in cloud.

In 2013, Baris Yuce[22] introduce with "Honey bee inspired algorithm" which focuses on improving bench mark functions are compared with other optimised techniques ACO,PSO and EV for testing the bee behaviour and algorithm. In this aim was to improve bee's algorithm (BA) by utilising adaptive neighbourhood sizes and site abandonment (ANSSA) strategy. This algorithm tested accuracy, average evaluation and t-test between bee's algorithm and other for comparing and resulting in best behaviour study and analysis of the pattern of bees inspired algorithm.

Below table1 comparison of load balancing on the basis of parameters and table2 gives the comparison of load balancing based on survey of algorithms.

TABLE 1: Comparison on the basis of load balancing parameters

Metrics / Algorithms	**Overhead**	**Throughput**	**Process Migration**	**Response Time**	**Resource Utilization**	**Fault Tolerant**	**Waiting Time**
Round robin[1]	YES	YES	NO	YES	YES	NO	YES
Active clustering[13]	YES	NO	YES	NO	YES	NO	YES
Particle swarm optimization[22]	YES	YES	NO	YES	YES	NO	NO
Genetic algorithm[29]	NO	YES	NO	YES	YES	NO	NO

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

Throttled load balancing [27]	NO	YES	NO	YES	YES	NO	NO
Max-Min[19]	YES	YES	NO	YES	YES	NO	YES
Load balance Min-Min[16]	YES	YES	NO	YES	YES	NO	YES
Ant colony optimization[20][30]	NO	NO	NO	NO	YES	YES	NO
Bee algorithm [23]	NO	NO	NO	NO	YES	NO	NO

TABLE2: Comparison on load balancing survey (i.e. based on static/dynamic, simulator and key concepts)

Algorithms	**Static v/s dynamic**	**Simulator/ tools**	**Key concept**	**Metrics used**	**Merits**	**Demerits**	**Implementation complexity**
Round robin[1][6]	Static	cloud analyst	Follow FIFO manner and works on dynamic time quantum computation.	Completion time	Every process get equal weight age so no process will go under starvation	Most of the time processor remains idle	LOW
Active clustering[7] [10]	Dynamic	cloud sim	Optimizes job assignment by connecting similar services by local re-wiring	Throughput, Job completion time, Overhead	increase in throughput	More complex in networks	HIGH
Particle swarm optimization [13]	Dynamic	cloud sim	Iterative selection of particle delivering VM's to neighbouring host.	Throughput, Job completion time, Overhead	Particle will move through a multidimensional search space to find the best position in that space (the best position may possible to the maximum or minimum	Higher throughput: More sophisticated finite element formulations	LOW

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

values)

Genetic algorithm[3]	Dynamic	Cloud analyst	Randomly population of processing unit is initialized first and encoded them into binary strings	Genetic based parameters	Load balance, solving the problems of high migration	Genetic-based algorithm	HIGH
Throttled load balancing [9][21]	Dynamic	Cloud analyst	Allocate VM's according to indexing and matching configuration	Communication cost, Network Delay, Load Movement Factor	High Load Movement Factor	High communication	HIGH
Max-Min[2]	Static	Grid sim	Selecting task with minimum completion time.	Execution time	Better make span and selection of resources	Algorithm Complexity	LOW
Load balance Min-Min[2]	Static	Grid sim	It identifies resources with high make span and then selects the task with minimum execution time.	Execution time	Job with smallest execution time is executed	Number of the small tasks is more than the number of the large tasks in a meta-task, the Min-Min algorithm cannot schedule tasks appropriately	LOW
Ant colony optimization [8][11][4][20]	Dynamic	Grid sim, Cloud sim, net beans	Scheduling is performed to avoid deadlock with maximum resource utilization.	Completion time	Every process get equal weight age so no process will go under starvation	Most of the time processor remains idle	LOW
Bee algorithm [22]	Static	Normally distributed data sampling and T test simulator	An optimization algorithm inspired by the natural foraging behaviour of honey bees, called the Bees Algorithm.	Throughput, Job completion time, Overhead	increase in throughput	More complex in networks	HIGH

IV. CONCLUSION AND FUTURE WORK

Cloud world is evolving fast, furiously gaining greater momentum as we go into 2015 and leaving bequest on premise systems light years behind. Cloud has the ability to altering the way big/small organizations works according to demands that comes with the time and adapting the new scenarios that comes with its own set of challenges. Proper load balancing aids in avoiding fail-over, enabling flexibility, scalability, reducing over-provisioning VM allocation and provisioning, minimizing resource utilization and avoiding bottlenecks etc. This paper provides a illustration and

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 11, November 2015

complete survey on load balancing algorithms in cloud computing environment along with their corresponding advantages, disadvantages, implementations and performance metrics are discussed in a tabular manner.

In future research works will be done on implementation of cloud as IAAS for computing and further improvement in this field by hybrid techniques and more energy conservation techniques can be deployed. Various open access and open source infrastructures can be developed in cloud. Cloud robotics will be the next era of computing services which will surely optimize workload by providing accuracy, efficiency, flexibility, low cost times with powerful computation and processing resources.

V ACKNOWLEDGMENT

I want to express my heartiest gratitude towards my father Dr. A. K Mathur and mother Mrs. Hemlata Mathur and Fiancé Varun mathur for their support and encouragement throughout the period this work was carried out.

REFERENCES

1. Stuti Dave, Prashant Mehta "Utilizing Round Robin Concept for Load Balancing Algorithm at Virtual Machine Level in Cloud Computing" IJAC (0975-8887) Volume 94-No.4, May 2014.
2. O.M.Elzeki, M.Z.Reshad, M.A.Elsoud "Improved Max- Min Algorithm in Cloud Computing" IJCA (0975-8887) Volume 50- No. 12 July 2012.
3. Kousik Dasgupta, Brototi Mandal, Paramartha Dutta, Jyotsna Kumar Mondal, Santanu Dam, "A Genetic Algorithm (GA) based Load Balancing Strategy for Cloud Computing"in Proc. of Elsevier, Procedia Technology 2013.
4. Ratan Mishra and Anant Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud",in International Journal of Web & Semantic Technology (IJWesT), Vol.3, No.2, pp. 33-50, 2012.
5. Manan D. Shah ,MR.Amit A. Kariyani ,MR.Dipak L. Agrawal "Allocation Of Virtual Machines In Cloud Computing Using Load Balancing Algorithm"IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 3, No.1, February 2013.
6. Subasish Mohapatra, Subhadarshini Mohanty and K. Smruti Rekha "Analysis of Different Variants in Round Robin Algorithms for Load Balancing in Cloud Computing" in International Journal Of Computer Applications (0975-8887) Volume 69 – No. 22, May 2013.
7. Shanti Swaroop Moharana , Rajadeepan D. Ramesh & Digamber Powar, "Analysis of Load Balancers in Cloud Computing" in International Journal of Computer Science and Engineering (IJCSE), ISSN 2278-9960 Vol. 2, Issue 2, May 2013, 101-108 ©IASE.
8. M Dorigo, G.D. Caro, and L.M. Gambardella, "Ant algorithms for discrete optimization," Artif. Life, vol.5, no.2, pp.137-172, 1999
9. Rajwinder Kaur1 and Pawan Luthra , "Load Balancing in Cloud Computing" in ACEEE, Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC
10. Randles M., Lamb D. and Taleb-Bendiab A. (2010) 24th International Conference on Advanced Information Networking and Applications Workshops, 551-556, IEEE 2010.
11. Ratan Mishra and Anant Jaiswal, "Ant colony Optimization: A Solution of Load balancing in Cloud",International Journal of Web & Semantic Technology (IJWesT) Vol.3, No.2, April 2012.
12. Elina Pacini,Cristian Mateos and Carlos Garc´ıa Garino , , "Dynamic Scheduling based on Particle Swarm Optimization for Cloud-based Scientific Experiments"HPCLatAm 2013 VI Latin American Symposium on High Performance Computing.
13. T. Kokilavani and Dr. D.I. George Amalarethinam, "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.
14. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg and I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, Future generation system, and 2009.
15. Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, C´esar A. F. De Rose and Rajkumar Buyya,"CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", SOFTWARE – PRACTICE AND EXPERIENCE Softw. Pract. Exper. 2011; 41:23–50 published online 24 August 2010 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/spe.995
16. Bhathiya Wickremasinghe1, Rodrigo N. Calheiros, and Rajkumar Buyya "CloudAnalyst: A CloudSim-based Visual Modeller for Analyzing Cloud Computing Environments and Applications", 2009.
17. Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, "Above the Clouds: A Berkeley View of Cloud Computing" Electrical Engineering and Computer Sciences University of California at Berkeley, Technical Report No. UCB/EECS-2009-28.
18. Indresh Gangwar and Poonam Rana ," Juxtaposition of Load Balancing Algorithms in Cloud Computing using Cloud Analyst Simulator", International Journal of Computer Applications (0975 – 8887) Volume 97– No.2, July 2014.
19. Sridevi S, Chitra Devi D, and Dr. V. Rhymend Uthariaraj, "Efficient Load Balancing and Dynamic Resource Allocation in Cloud Environment",International Journal of Engineering Research & Technology (IJERT) ISSN: 2278-0181 IJERTV4IS020612 Vol. 4 Issue 02, February-2015.
20. Ekta Gupta, Vidya Deshpande, "A Load Balancing Technique for Servers of Datacenter of Cloud using Ant Colony Optimization ",ISSN (Online): 2347 - 2812, Volume-2, Issue -6,7, 2014.
21. Shridhar G.Domanal and G.Ram Mohana Reddy, "Load Balancing in Cloud Computing Using Modified Throttled Algorithm",IEEE 2013.
22. Baris Yuce, Michael S. Packianather, Ernesto Mastrocinque, Duc Truong Pham and Alfredo Lambiase, "Honey Bees Inspired Optimization Method: The Bees Algorithm",Insects 2013, 4, 646-662; doi:10.3390/insects4040646.
23. Mayanka Katyal, Atul Mishra, "A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment", International Journal of distributed and cloud computing volume 1 issue 2 December 2013.