ISSN: 2321-7782 (Online)

Volume 2, Issue 11, November 2014

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

Survey of Load Balancing Algorithms in a Distributed Data Centers

N. Deebaa[1]
Post Graduate / CSE
SNS College of Technology
Coimbatore, India

A. Jayanthi[1]
M.E
Assistant Professor / CSE
SNS College of Technology
Coimbatore, India

Dr. S. Karthik[3]
M.E, Ph.D
Professor & Dean / CSE
SNS College of Technology
Coimbatore, India

Abstract— Cloud computing is a developing technology to store and retrieve data or information. One of the issues in cloud computing technology is to store the information in a balanced order over data centers. Load balancing is a process to distribute the workload across many computers or ins data centers to maximize throughput and minimize work load on resources. A survey is made on some papers in load balancing and the techniques used in it. Algorithms like scheduling algorithm, static and dynamic load balancing, etc are used for load balancing in data centers.

Keywords— Cloud computing, Load balancing, scheduling algorithm, static and dynamic algorithm.

I. INTRODUCTION

Cloud computing is a pool of data which can be accessed from anywhere in the world through internet. The computer device will be an intermediate to access the data from the pool to the users. Cloud computing will have hardware and software which is installed in its own server. Cloud computing is said to be a new computing paradigm, which involves the data and other computation outsourcing with – more and expandable resource scalability, on-demand “just-in-time” provisioning, and it is a “pay-as-you-go” method. The name cloud computing was inspired by the cloud symbol that's often used to represent the internet in flowcharts and diagrams.

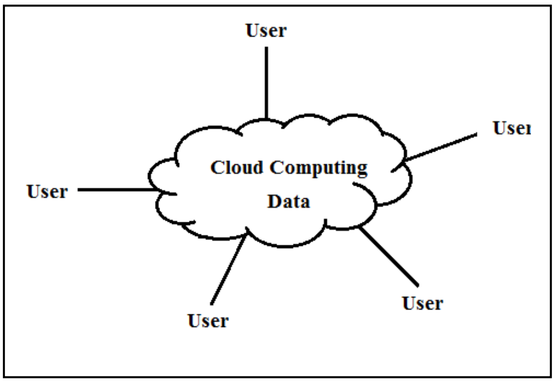

Fig 1- Cloud Computing

Some of the characteristics in cloud computing are on-demand self-service, broad network access, resource pooling, rapid elasticity, measured service, etc,.

Cloud computing can be provided with three services:

Infrastructure-as-a-Service (IaaS): It is also said to be hardware-as-a-service. It is the base layer of the cloud stack and to execute other two layers it serves as a foundation. IaaS can be rented for limited period of time as per user needs.

Platform-as-a-Service (PaaS): It is the platform for developers to write and create their own application. The customer has the freedom to built their own application, which run on the provider's infrastructure.

Software-as-a-Service (SaaS): SaaS is an on-demand service. It can be accessed from anywhere in the world if we have the computer and internet connection.

Load balancing is a process of reassigning the total load for each separate servers which is present in the data centers. Load balancing is done to make resource utilized in a proper way. Through load balancing we can manage a large amount of data traffic in the data centers. Load balancing goals are to improve the performance substantially, to have a backup plan if the system fails, to maintain the stability of the system, and to accommodate potential modification in the system. In load balancing there are two types, static load balancing and dynamic load balancing. In static algorithm the receiving work can be divided equally to all servers present in the system. Here in the static algorithm, knowledge about the resource in the system should be known to the algorithm earlier itself. So that is easy to shift the load among systems and it does not depend on the present state of the system. Static algorithm will be proper in the data centre when the system load is low. A load balancing algorithm which is dynamic in nature does not consider the previous state or behavior of the system i.e. it depends on the present state of the system.

II. LITERATURE SURVEY

a) Cloud Computing Online Scheduling

Arabi E. Keshk is proposed an ant colony algorithm for online cloud task scheduling based on virtual machine adaptive fault tolerance and load balancing. In cloud computing, task scheduling is the basic issue. To make use of the cloud efficiently, a good task scheduling algorithm is required to assign tasks to the resources in cloud. The two categories of cloud task can be on-line mode service and the batch mode service. By using an ant colony algorithm we can achieve a good result than the other algorithms like Joint-shortest-queue (JSQ) and Modified Ant colony Optimization (MACO) algorithm by simulating in CloudSim toolkit package.

In Online mode, whenever a request arrives it is allotted to the first free resource immediately. In this method, the arrival order of the request is very essential. For matching and scheduling the request in resources it is considered only once at a time. In Batch mode service, the requests arrived are collected first. Then the scheduler fixes the approximate execution time for each task and use the heuristic approach to make better choice [2]. The main task of the paper is to find the good resource allocation for each task in the dynamic cloud by balancing the load of the system and adapting the dependability of the system.

Cloud scheduling can be categorized as user level and the system level. User level scheduling will deal with problems occur by the service provider and the customers. The system level scheduling will handles resource management. The heuristic-based request scheduling is done at each server in each of the geographically distributed data centers to minimize the penalty charge to the cloud computing systems. The proposed online cloud task scheduling algorithm is used to allocate the incoming jobs to virtual machines for load balancing and fault tolerance to utilize the available resources and to minimize the resource consumption. The advantage of the ant colony algorithm are to use the positive comment mechanism, internal parallelism and extensible. The disadvantages of this algorithm are overhead and the stagnation phenomenon or searching for to a certain extent i.e. all the individuals will exactly found the same solution and further it can't search for the solution space and make the algorithm converge to local optimal solution.

b) Load Rebalancing for Distributed File Systems in Clouds

Hung-Chang Hsiao, et al., proposes a fully distributed rebalancing algorithm. It is used to solve the problem of load unbalancing in resources. Distributed file system plays a main role in cloud applications based on the MapReduce technique.

Here in MapReduce, the files are divided and allotted to the required nodes. So that MapReduce can perform the task in parallel over all the nodes. But in the cloud system failure can occur due to upgraded, replace and adding of new data in the system. So load imbalance can occur in a distributed file system. Fully distributed rebalancing algorithm is compared with the centralized approach and a competing distributed solution. The simulation result can give better performance than the prior techniques in load unbalance factor, cost and algorithm overhead.

MapReduce technology is used to reduce the redundant data in the system. So load balancing can be done while the data are processed in the system. For example if we take a list of names and same name can be repeated in the list. Here MapReduce can eliminate the repeating name by scanning and parsing the file and it will keep its id or some other data for the reference of that name. So here the data load can be reduced and it will process in a balanced order. The State-of-the-Art distributed file system will relay on central node of the system. The central load can also be over loaded as in the middle of the process a bottle neck can occur due to accommodation of more data in a same place.

Load rebalancing in distributed file system, number of chunk servers are used. In each server more number of data is loaded. Then each file is separated in to number of disjoint, with fixed size chunk. After that, the load of a chunkserver is equal to the number of chunks hosted by the server. Here the chunks may be removed, edited or replaced. So there is an unbalanced chunk in the server.

Load rebalancing algorithm is used to reallocate the chunks equally in the distributed file system. The load rebalancing algorithm helps to balance the work load in the system when compare to the centralized algorithm. Here it improves the movement cost, load imbalance and algorithm overhead.

c) Temporal Load Balancing with Service Delay Guarantees for Data Center Energy Cost Optimization

Jianying Luo, et al., proposes a novel two-stage design and eco-IDC (Energy Cost Optimization-Internet Data Centers) to sequential diversity of electricity cost and dynamically scheduled workload to perform on IDC servers through an input queue. This operations are performed in the data center practically and the result shows that the electricity cost got reduced and provide a service delay bound, and when the service delay bound is large it will alleviates workload drop.

Internet Data Centers is to support the cloud computing infrastructure. An IDC will be designed with hundreds and thousands of servers and it will consume more power for running and cooling the equipments. As the usage of cloud computing is becoming more and more, the usage of power is also getting high. So both the industries and academia were interested to reduce the energy consumption in data centers as if the energy get reduced, the cost of energy will automatically reduced. IDC operators are come across two major problems: the hesitation in workloads of user requests, and the need for service interruption guarantee. To reduce the electricity price, the energy consumption in IDC can be reduced and the other is to manage the heavy workload.

An energy cost minimization problem for IDC and eco-IDC algorithm is considered. Three goals are to be formulated here to achieve the target. First, exploiting the sequential variety of electricity price to reduce energy cost; second, guaranteeing a service delay bound to each user request; and third, balancing the work load when it exceeds the maximum level of IDC. A novel two-stage design is proposed at the end. In the second stage, an energy cost minimization scheduler is proposed on eco-IDC algorithm to full fill the first two goals. In the first stage, a workload shaping controller completes our third goal. A practical work can done in the data centers and it can be used to improve the energy cost reduction, queuing delay, work load drop proof.

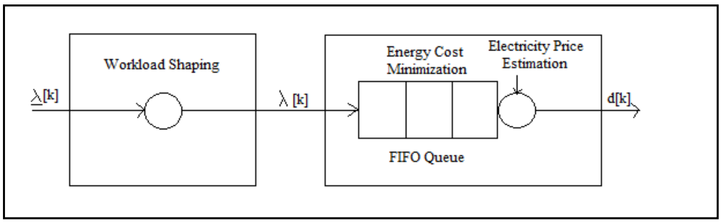

Fig 2- Two-stage design

A novel two-stage design and eco-IDC algorithm is used to reduce the power consumption and the work load scheduling in data centers. Extensive performance evaluation demonstrates that the proposed approach significantly reduces energy cost for IDC operations, guarantees a service delay bound for user requests, and alleviates workload drop if the service delay bound is sufficiently large.

d) Optimal Power Allocation and Load Distribution for Multiple Heterogeneous Multicore Server Processors across Clouds and Data Centers

Junwei Cao, et al., approach is to formulate optimal power allocation and load distribution for multiple servers in a cloud of clouds as optimization problems, i.e., power constrained performance optimization and performance constrained power optimization. Our optimization problems are solved for two different models of core speed, where one model assumes that a core runs at zero speed when it is idle, and the other model assumes that a core runs at a constant speed. In a data center with multiple servers, the aggregated performance of the data center can be optimized by load distribution and balancing. Cloud-based applications depend even more heavily on load balancing and optimization than traditional enterprise applications. For end users, load balancing capabilities will be seriously considered when they select a cloud computing provider. For cloud providers, load balancing capabilities will be a source of revenue, which is directly related to service quality (e.g., task response time). Hence, an efficient load balancing strategy is a key component to building out any cloud computing architecture.

The two important research problems that explore the power performance tradeoff in large-scale data centers from the perspective of optimal power allocation and load distribution. Our strategy is to formulate optimal power allocation and load distribution for multiple servers in a cloud of clouds as optimization problems. Our problems are defined for multiple multicore server processors with different sizes, and certain workload:

Power constrained performance optimization - Given a power constraint, our problem is to find an optimal power allocation to the servers (i.e., to determine the server speeds) and an optimal workload distribution among the servers, such that the average task response time is minimized and that the average power consumption of the servers does not exceed the given power limit.

Performance constrained power optimization - Given a performance constraint, our problem is to find an optimal power allocation to the servers (i.e., to determine the server speeds) and an optimal workload distribution among the servers, such that the average power consumption of the servers is minimized and that the average task response time does not exceed the given performance limit.

Performance Optimization and performance constrained power optimization are considered here. By proper load balancing algorithm this problem can be avoided. Multicore server processor is used as a queuing system with multiple servers. Optimization problem can be solved in core speed and constant speed models. A new theoretical approach can be provide for power management and performance optimization.

e) Analysis of Load Balancers in Cloud Computing

Shanti Swaroop Moharana., et al., is proposed a load balancing algorithm which is used for assigning data to different systems so that none of the nodes gets loaded heavily or lightly. The load balancing should to be done accurately, because failure in any one of the node can lead to loss of data. In distributed system environment, both resource utilization and job response can be improved by distributing nodes to various other distributed systems.

Load balancers can work in two ways: cooperative and non-cooperative. In cooperative, the nodes work concurrently in order to reach the common goal of optimizing the overall response time. In non-cooperative mode, in order to improve the response time of local task, the main task will runs independently. Load balancing algorithms can be divided into two categories: static and dynamic load balancing algorithm [3].

Static Load balancers:

Round-Robin Load Balancer - It is a static load balancing algorithm, it does not consider the order of receiving state of the node while the job is allocating to servers. It chooses the first node randomly and allocate the jobs to all nodes in a round robin method. In cloud computing, some nodes will be heavily loaded and some may be lightly loaded. So this algorithm is not suitable.

Min-Min - It is a static load balancing algorithm. Here, the information about the jobs will appear in advance. Min-Min algorithm starts with a set of all unassigned jobs. Here, the job with minimum completion time will be selected first and the node which has minimum completion time for all jobs is selected then. At last the selected job and node will be mapped. Updating of ready time will be finished. This process will be continued until the unassigned job are assigned. The advantage is that the job which is having minimum time will be executed quickly and the disadvantage is that the job with maximum time can face a starvation problem.

Load Balance Min-Min – LBMM [4,5] is a static load balancing algorithm. LBMM algorithm is done for load balancing problem among nodes and the scheduling problem is considered. The main need of this algorithm is Make-span, which means that the maximum time should be calculated for the completion time for all the jobs scheduled in their respective resources.

Load Balance Max-Min-Max **-** S.-C. Wang et al. combine two algorithms and propose a two-phase scheduling algorithm. Here the algorithms combined are OLB (Opportunistic Load Balancing) and LBMM (Load Balance Min-Min) scheduling algorithms. In OLB scheduling algorithm all the nodes will be in working state to achieve the goal of load balancing. In LBMM scheduling algorithm is used to reduce to execution time of each task on the node and the overall completion time will be reduced. It helps to utilize the resource efficiently.

Dynamic Load Balancers:

Equally Spread Current Execution (ESCE) **–** ESCE [6] is a dynamic load balancing algorithm and it works in priority basis. Priority is done according to the size of the project. It first check the size of the project and distribute the load to the virtual machine and distribute to the lightly loaded. The load balancer spread the load to different nodes, so it I said to be spread spectrum technique.

Throttled Load Balancer (TLB) - Throttled load balancer [6] is a dynamic load balancing algorithm. Here, the client will request for the suitable virtual machine to perform the required operation.

Honeybee Foraging Algorithm (HFA) - The key thought at the back of Honeybee Foraging algorithm [7] is derived from the activities of honeybees. In honeybees we can find two types: finders and reapers. The finder honeybees first go outside from the honey comb and discover the honey sources. After discovering the source, they come back to the honey comb and do a waggle dance indicating the quality and quantity of honey available. Then, reapers go out and reap the honey from those sources. After collecting, they come back to beehive and do a waggle dance. This dance shows how much food is there.

M. Randles planned a decentralized honeybee based algorithm for self-organization. In this case, the servers are combined as virtual server and for each server a seperate process queue is allotted. Each server, after giving out a request from its queue, calculates the income which is similar to the quality that the bees show in their waggle dance. If profit is sky-scraping, the server will remain for further work otherwise it will be back to the forage. This algorithm requires that each node to maintain an individual queue. The computation of profit on each node will make additional overhead. The drawback of this algorithm is that it won't show any considerable improvement in throughput because of additional throughput and computational overhead.

Biased Random Sampling (BRS) - Biased Random Sampling is a dynamic load balancing algorithm. To achieve the self-organization a random sampling of system domain is used. In BRS, a virtual graph is constructed with the connectivity of each node by representing the load on server. Each node is depict as a vertex in a directed graph and each in-degree represents free resources of that node.

Active Clustering – Active clustering [8, 9] is a cluster based algorithm. It introduces the concept of cluster in cloud computing. The procedure of creating a cluster revolves in the region of the concept of match maker node. In this process, first node selects a nearby node called the matchmaker node which is of a various type. This matchmaker node makes connection with its nearby node which is of similar type as the starting node. Finally the matchmaker node gets separate. This process is followed iteratively. The performance of the system is enhanced with high availability of resources, thereby increasing the throughput. This increase in throughput is due to the efficient utilization of resources.

In cloud computing to solve the load balancing problem, static and dynamic algorithms are used. Both static and dynamic having more than one algorithm for each condition in load balancing. In future, the algorithms with numerical analysis and simulation can be analyzed.

f) Scheduling Jobs With Unknown Duration in Clouds

Siva Theja Maguluri, et al., proposes a MaxWeight Scheduling algorithm. It is a nonpreemptive algorithm. Nonoreemptive algorithm is when the jobs are arrived it will routed in one of the server and it will be queued to servers. From the queue each server chooses a set of jobs so that work can be divided equally. This problem can be known that the data are known and upper-bounded, and the algorithm can be proposed for proper load balancing.

MaxWeight scheduling algorithm is known to have good delay performance and through simulation it had been studied. Heavy traffic optimality and large deviations optimality have been established in [12 nd [13]. As the MaxWeight scheduling algorithm is a nonpreemptive it is very complex to study as the state of the system in different time-slot were combine with other. For example, the MaxWeight schedule cannot be chosen in each time-slot nonpreemptively. If there are any unfinished jobs it will be server at the beginning of time-slot. This job cannot be paused in the new time-slot. So, the new schedule should be chosen to include these jobs. But the MaxWeight won't allow these jobs to include in the process. Nonpreemptive algorithms were considered in literature in the situation of input queued switches with inconsistent packet sizes. The nonpreemptive algorithm will use a special structure of a switch, so it is not clear as to how it can be generalized for th case of a cloud system.

The MaxWeight scheduling algorithm uses the wireless nodes to transfer the data. The main drawback of this algorithm is that, if the number of wireless nodes increases its complexity also increases.

III. CONCLUSION

A survey is made on load balancing algorithms and there are two categories in it, static and dynamic load balancers. Here algorithms like ant colony algorithm, load balancing algorithms, fully distributed rebalancing algorithm, MaxWeight scheduling algorithm, and Energy Cost Optimization-Internet Data Centers are used. Some techniques like performance optimization, performance constrained and novel two stage design for electricity consumption are used

References

1 Arabi E. Keshk "Cloud Computing Online Scheduling" ISSN (e): 2250-3021, ISSN (p): 2278-8719 Vol. 04, Issue 03 (March. 2014), ||V6|| PP 07-17

2 Banerjee S, et al., "Cloud computing initiative using modified ant colony framework" In: World academy of science, engineering and technology, p. 221–224, 2009

3 Shanti Swaroop Moharana, et al., "Analysis of Load Balancers in Cloud Computing" International Journal of Computer Science and Engineering (IJCSE) ISSN 2278-9960 Vol. 2, Issue 2, May 2013, 101-108

4 Graham Ritchie, John Levine, ""A fast effective local search for scheduling independent jobs in heterogeneous computing environments" Center for Intelligent Systems and their applications School of Informatics University of Edinburg

5 Shu Ching Wang, Kuo-Qin Yan Wen-pin Liao, and Shun Sheng Wang Chaoyang University of Technology, Taiwan R.O.C

6 Nitika, Shaveta, Gaurav Raj, International Journal of advanced research in computer engineering and technology Vol-1 issue-3 May-2012

7 Yatendra sahu, M. K. Pateriya "Cloud Computing Overview and load balancing algorithms", Internal Journal of Computer Application Vol-65 No.24, 2013

8 Nayandeep Sran, Navdeep kaur "Comparative Analysis of Existing Load balancing techniques in cloud computing", International Journal of Engineering Science Invention, Vol-2 Issue-1 2013

9 Nidhi Jain Kansal, Inderveer Chana "Existing Load balancing Techniques in cloud computing: A systematic review" Journal of Information system and communication Vol-3 Issue-1 2012

10 Hung-Chang Hsiao, et al., "Load Rebalancing for Distributed File Systems in Clouds", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 24, NO. 5, MAY 2013

11 Junwei Cao, et al., "Optimal Power Allocation and Load Distribution for Multiple Heterogeneous Multicore Server Processors across Clouds and Data Centers", IEEE TRANSACTIONS ON COMPUTERS, VOL. 63, NO. 1, JANUARY 2014

12 A. Stolyar, "Maxweight scheduling in a generalized switch: State space collapse and workload minimization in heavy traffic," Ann. Appl. Probab., vol. 14, no. 1, pp. 1–53, 2004

13 V. Venkataramanan and X. Lin, "On the queue-overflow probability of wireless systems: A new approach combining large deviations with Lyapunov functions," IEEE Trans. Inf. Theory, vol. 59, no. 10, pp. 6367–6392, Oct. 2013

14 Jianying Luo, et al., "Temporal Load Balancing with Service Delay Guarantees for Data Center Energy Cost Optimization", IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, VOL. 25, NO. 3, MARCH 2014