International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Optimal Resource Allocation and Load Distribution for Server Processors using Hot Spot Migration

Leema D.A[1], Dr. K. N. Narasimaha Murthy[2]

[1]PG Scholar, Department of Computer Science and Engineering, Vemana IT, Visvesvaraya Technological University, Belagavi , Karnataka, India

[2]Professor and Head (PG), Department of Computer Science and Engineering, Vemana IT, Visvesvaraya Technological University, Belagavi , Karnataka, India

Abstract: ***Load distribution is one of the most important problem faced in the cloud computing environment when the servers are connected to the network. Load distribution is used to distribute the workload among multiple computers and their resources. Whenever a user needs to run an application on the server, the processing begins and the computer needs to allocate some resources for the application to run. Here the resources are allocated dynamically, depending on the application usage and load distribution the performance of the server is optimised. In this paper the concept of skewness is used to find the uneven resource utilization and distribution of load.***

Keywords: load distribution, virtualization, skewness, prediction, green computing, overhead

1. Introduction

Cloud Computing is one of the widely used methods to store large amount of data. It is one of the best ways of utilizing and managing the computer resources. In order to use the cloud efficiently a server consolidation approach is used. This will reduce the number of servers in the network [1]. Here the resource management is centralized. The cloud services are provided in the internet by using the cloud computing host services. The information in the database, hardware, software and all other resources are given for the user to use on-demand. Now a day a cloud of clouds approach is used. One can access multiple clouds and data centres [2]. It provides a flexible and more powerful way of utilizing the resources.

This can be used in different fields like scientific and business management for large scale computer system [3]. Here one can access a shared pool of resources like servers, networks, applications on the servers, on-demand services etc. Using all these resources may lead to inefficient usage of available resources, energy wastage. Hence load balancing is used to distribute the resources efficiently and more effectively among the computer systems [4].

The organization of the paper consists of the following. Section 2 contains all the related work on Load balancing. Section 3 deals about the static load balancing and dynamic load balancing. Section 4 is about dynamic resource allocation using virtual machines. Section 5 gives the system design. Section 6 explains the algorithms used. Section 7 provides the result analysis.

2. Related Work

The systems that are distributed in a network take lot of load. In order to provide a solution for this load, the load balancing techniques are used. There are many load balancing techniques and algorithms that are discussed below.

Decentralizes Dynamic Load Balancing For Computational Grid Environments .In this paper the load balancing is done for grid environment. Scheduling is done using the grids. A Decentralised dynamic load Balancing algorithm is used which combines the cluster and neighbour based load balancing techniques [5], here some of the system parameters are taken, like load on each system, resources, processing capacity, transfer delay, Load on each and every resources.

The main objective is to minimize the response time of the jobs that arrive to the grids for processing and while load information are collected the communication overhead has to be reduced[6]. The instantaneous job migration algorithm is used to compared with the load adjustment policy i,e is applied for the grid environment. The limitation is that the fault tolerance is less compared to the other applications.

Scalable Distributed Job Processing with Dynamic Load Balancing.The dynamic computing needs are provided by a distributed job processing system which provides an efficient load balancing and it is scalable for the heterogeneous systems. It is designed in such a way that each and every system is self contained and they do not depend on each other [7]. In order to provide a secure and reliable communication they are interconnected with an enterprise message bus. The data duplication can be avoided by using these transactional features.

Fault tolerance and data failover can be recovered by building the health check mechanism and the load balancing is done based on the queue. Various jobs and their progress can be tracked by having a central monitor. This is present in the centralized repository where it has the status and execution of the real time processors. The limitation is that the systems do not include the failover recovery for the frame work that provides the state of processing at various stages and maintaining their processes.

Optimal Load Distribution for Multiple Heterogeneous Blade Servers in a Cloud Computing Environment .The blade servers are striped server computers with a modular design that is used to optimize and minimize physical space and energy of the system. These heterogeneous blade servers have different sizes and speed. They can perform special tasks[8-9]. When these servers are used in cloud environment there is a problem in providing optimal distribution for generic jobs to the blade servers. So that the average response time of these task is minimized. In order to provide efficient utilization of all the available resources and to provide high quality service, the performance has to be optimized. Hence they have used a queuing model to group the heterogeneous blade servers. To formulate and solve the problem of optimal load distribution.

The average response time of these generic tasks are based on the server speed, task execution time, arrival rate of special tasks and other tasks, server size etc [10]. The heterogeneity of server size and speed do not have much impact on the average response time of generic tasks.

Balancing Server in Public Cloud using AJAS Algorithm.In this paper the load balancing is based on traffic and unwanted treat mechanisms. In most of the cloud environment the load balancing provides an impact on system performance. Here the improved efficiency and the user satisfaction are the main goals for good load balancing [11]. By using switch mechanism that chooses many strategies at different situations for providing an optimal solution.

The algorithm used here is the AJAS Algorithm (Adaptive Scoring Job Scheduling Algorithm). This uses the concept of game theory to balance the load and improve the public cloud environment efficiency.

Issues in Load Balancing and Scheduling

- In Load Balancing Methods the requirements like stability, Scalability, overhead of the system are all Interdependent.
- Processors are migrated from one node to another when they are running is a critical task.
- Balancing the Load together with the shortest possible time for executing the task is important.
- The Load sharing provides efficiency to a certain extent, but still some nodes will be idle while the others are overloaded.
- Since the systems are distributed over the network, balancing and scheduling is a critical task. Because the overall systems in the network are non-uniform and non-pre-emptive, since there processors have different configurations and capacities.

3. Static and Dynamic Load Distribution

Static Load Balancing is done depending on the average behaviour of the system. It is independent of the system's current Status. It makes use of the statistical information of the system. When the process is executed the performance of that processor is determined. Once the performance of the particular processor is determined the workload is assigned to that system by master processor. The slave processor's processes the job and provides the result to the master[12]. The main goal of static Load balancing is to reduce the communication delay and execution time of each and every task. The disadvantage of static approach is that once the process execution starts the system load cannot change. There are four algorithms that make use of static load balancing they are: Round Robin, Threshold Algorithm, Randomized Algorithm and Central Manager Algorithm.

In dynamic load balancing the load is distributed to the processors during the run time. Once the master collects all the new information from the slaves regarding the current status of the system. It assigns the job to the slaves. Dynamic load distribution is done in two ways: For distributed system and non distributed system.

In distributed system, all the nodes execute the dynamic load balancing algorithm and shares the information among them. The nodes interactions are either cooperative or non-cooperative. In cooperative, all the nodes start working side-by-side to achieve a common objective. In non-cooperative, the nodes work independently to achieve their goals [13]. The main advantage of distributed system is that, when any node fails. It doesn't affect the entire system and the processes are not halt. It will affect the performance of the entire system to a certain extent.

In non-distributed system, one or group of nodes performs the load balancing[14-16]. Here there are two forms of non-distributed dynamic Load Balancing: Centralized and Semi-distributed.

In centralized, the load balancing algorithm is executed on a single node. This is called the central node. This node is responsible for balancing the load of the entire system. All the other nodes interact with the central node only.

In semi-distributed, clusters are formed by dividing or partitioning the nodes in the system. They are provided with a central node for each cluster[17]. These central nodes of each cluster take care of load balancing within the cluster and it sends the status update to the central node that manages all the clusters. Some of the qualitative parameters that as to be considered for load balancing are given below.

- **Nature** of load balancing algorithm, whether it is static or dynamic.
- **Reliability** is one of the important factor that as to be considered in case of system failure. The static load balancing is less reliable, since data cannot be transferred to another host for execution during the system failure. Dynamic is more reliable because data can be transferred. When the current system that executes the program fails.
- **Adapting** to changes is more important. This is done in dynamic load balancing.
- **Prediction:** The deterministic and non-deterministic factors can be predicted. prediction in static load balancing algorithm can be accurate, since the average execution time of each process and there workloads are fixed.

In dynamic, prediction can be done by seeing the internal and external behaviour of the system**.**

4. Dynamic Resource Allocation Using Virtual Machines

In this paper the Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment Cloud computing allows business customers to scale up and down their resource usage based on needs. In this paper, using virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. Techniques used are : (1) , (2). Virtualization technology, Skewness

This paper addresses:

Overload Avoidance: In order to satisfy the resource needs of the Virtual Machines (VM) the Primary Machines (PM) capacity must be sufficient to run the applications that are requested to run on the server by the clients.

Green Computing: The number of PM's used must to be reduced and still provide with the sufficient resources together with satisfying the VM's needs. To save the energy of the idle PM's they can be turned off.

Virtualization Technology: Based on the demands of the applications the virtualization can be used in data centres. This can provide an efficient utilization of resources.

Skewness: There are many PM's used in the network. Each and every primary machine has its own number of VM's running on the PM's. In order to measure the uneven utilization of their resources skewness is used.

Hot spot: Hot Spot is a situation where the servers are overloaded. Here there is a hot threshold that measures whether the system is in hot spot. **Cold spot**: when the servers are underutilized, it's called a cold spot server. Here there is a cold threshold indicates that the sever is in cold spot.

5. System Design

System Architecture:
When the active server resource utilization is too low, the system can be turned off in order to save energy. The green computing algorithm is used to handle this problem. The main challenge is that, without sacrificing the performance of the system, the number of active servers used must be reduced.

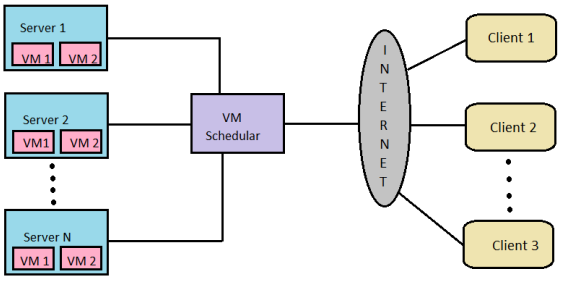

Figure 5.1: Client-server Architecture Diagram

The client server architecture diagram has a VM Scheduler has shown in Figure.5.1 which consist of the mechanisms called hot spot and cold spot. A server is a hot spot if the utilization of any of its resources is above a hot threshold. This indicates that the server is overloaded and hence some VMs running on it should be migrated away. A server is a cold spot if the utilizations of all its resources are below a cold threshold. This indicates that the server is mostly idle and a potential candidate to turn off to save energy.

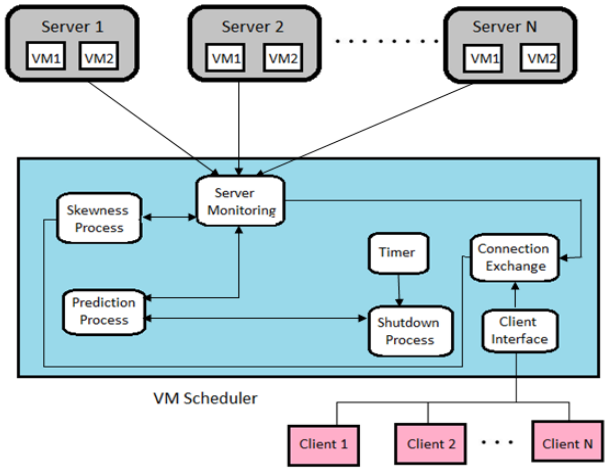

Figure 5.2: Modular Design Diagram

The list of hot spots in the system in descending temperature. Our goal is to eliminate all hot spots if possible. Otherwise, keep their temperature as low as possible. For each server p, we first decide which of its VMs should be migrated away. The list of VMs based on the resulting temperature of the server if that VM is migrated away. We aim to migrate away the VM that can reduce the server's temperature the most. In case of ties, we select the VM whose removal can reduce the skewness of the server the most.

The Figure 5.2 shows the modules used inside the VM Scheduler. It consist of four important modules Skewness process which contains the skewness algorithm to calculate the hot and cold spot. The next module is prediction Process which consist of the prediction algorithm to predict the future resource needs. Server Monitoring is used to check the servers during processing of the applications. Shutdown Process is used to turn off if there is any idle server.

Algorithm for Load Prediction:

The prediction algorithm plays an important role in improving the stability and performance of resource allocation decisions. Based on the PMs usage, the server can be selected using the prediction algorithm.

Step 1: The EWMA is calculated (Exponentially Weighted Moving Average).

$$E(t) = \alpha * E(t - 1) + (1 - \alpha) * O(t);$$

Step 2: The CPU load of each server is calculated using this EWMA formula.

Step 3: When the observed resource utilization is reducing and going down, the one need to reduce the estimated load

value. But most of the time the predicted values will be higher then the observed values.

Step 4: The predicted values are between historic and observed values when α is between 0 to 1.
The above prediction is done, on the bases of the past external behaviour of the VMs.
The above prediction is done, on the bases of the past external behaviour of the VMs.

Skewness Algorithm
The Server consist of many resources, the uneven utilization of these rsources can be measured using the skewness algorithm. Let n be the number of resources on a server and ri is the resource utilization of the servers ith resource.

6. Result Analysis

Here hot spot is migration is done by taking the number of jobs run by a server. Each time the algorithm runs, a VM is migrated from the server that is overloaded to another server. Sometimes the overload elimination may be a difficult task. Even after migration of some VM the server may be in hot spot. If it's still in hot spot, the process will be repeated. Algorithm can be designed to migrate multiple VMs also. This may create problems, by adding more load to the server. The VM are said to be in hot spot and they are migrated to warm spot and will not become hot by accommodating the VMs of the server .

Figure 7.1 shows the Different Spot Changes of the Server. This is done for the tested system. It may vary from one system to another.

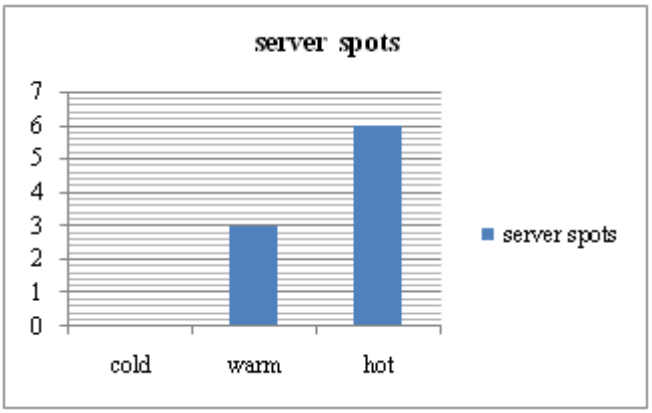

Figure 7.1: Server Spots

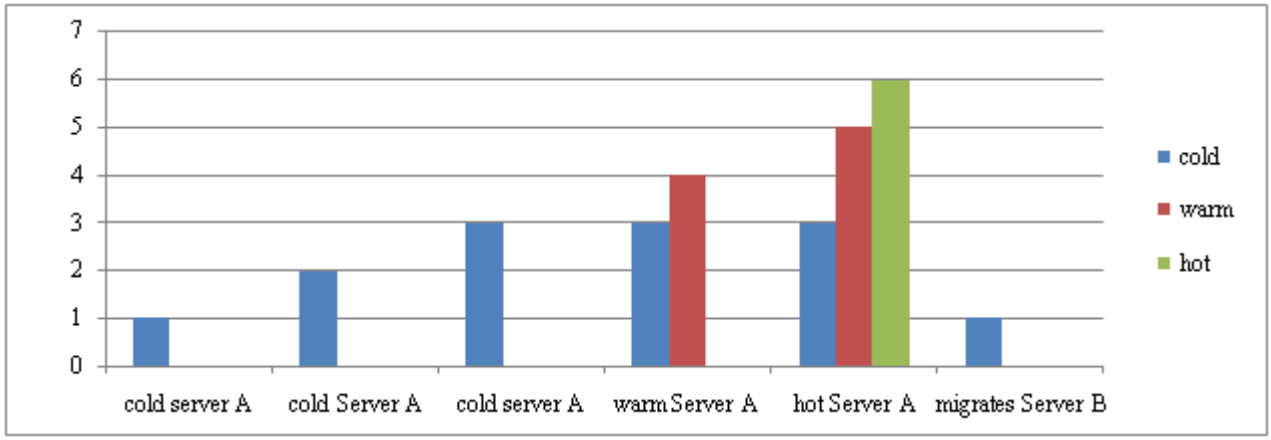

Figure 7.2: Load Migration from Hot Spot to Warm spot

Figure 7.2. This shows the migration from server A to Server B. When the number of job increases from 5 to 6 , then the 6th job is migrated to Server B, this happens based on the skewness algorithm.

Table 7.1: Performance Comparison with and without Skewness

No of jobs	Without Skewness Performance	With Skewness Performance
1	0.015	0.932
2	0.026	0.982
3	0.039	1.027
4	0.046	1.054
5	0.059	1.091
6	0.070	1.178
7	0.086	1.186

The table 7.1 indicates the increase in performance of the proposed system, where skewness is implemented. Here the comparison between the systems, where skewness is implemented and system without skewness is done. In this one can see that the skewness has increased the performance of the system. Figure 7.3 indicates the graph of the above table.

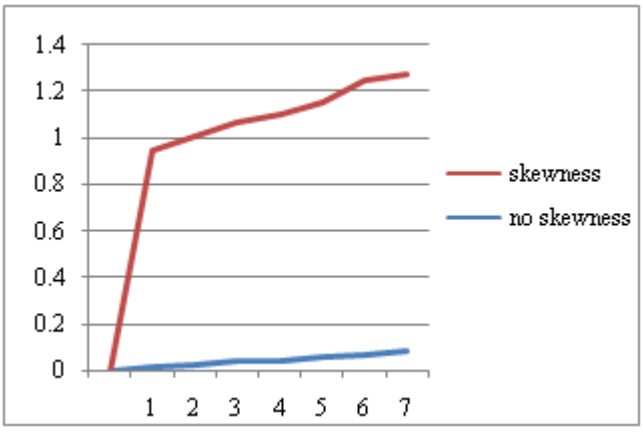

Figure 7.3: Comparison with and without Skewness

7. Conclusion & Future Work

The Optimal Resource Allocation and Load Distribution for server Processors using Hot Spot Migration. Here the significance and importance of balancing the load across multiple servers. The performance is optimized and the power consumption is reduced. In order to achieve these goals, Prediction and skewness algorithms are used to find the Hot spot and Cold spot. If the system is in hot spot the process can be migrated and if the system is in cold spot the prediction algorithm can be used to predict the future resource needs and turn off the system. Turning the system on is a complicated process. This can be done has future work using external controllers.

References

[1] http://en.wikipedia.org/wiki/CMOS, 2013.
[2] http://searchdatacenter.techtarget.com/definition/serverconsolidation,2013.
[3] A. Berl, E. Gelenbe, M.D. Girolamo, G. Giuliani, H.D. Meer, M.Q.Dang, and K. Pentikousis, "Energy-Efficient Cloud Computing,"The Computer J., vol. 53, pp. 1045-1051, 2009.
[4] F. Bonomi and A. Kumar, "Adaptive Optimal Load Balancing in a Nonhomogeneous Multiserver System with a Central Job Scheduler," IEEE Trans. Computers, vol. 39, no. 10, pp. 1232-1250, Oct.1990.
[5] A. Gandhi, V. Gupta, M. Harchol-Balter, and M.A. Kozuch, "Optimality Analysis of Energy-Performance Trade-off for Server Farm Management," Performance Evaluation, vol. 67, no. 11, pp. 1155-1171, 2010.
[6] L. He, S.A. Jarvis, D.P. Spooner, H. Jiang, D.N. Dillenberger, and G.R. Nudd, "Allocating Non-Real-Time and Soft Real-Time Jobs in Multiclusters," IEEE Trans. Parallel and Distributed Systems, vol. 17, no. 2, pp. 99-112, Feb. 2006.
[7] IBM, "The Benefits of Cloud Computing - A New Era of Responsiveness, Effectiveness and Efficiency in IT Service Delivery," Dynamic Infrastructure, July 2009.
[8] H. Kameda, J. Li, C. Kim, and Y. Zhang, Optimal Load Balancing in Distributed Computer Systems. Springer-Verlag, 1997.
[9] K. Li, "Optimizing Average Job Response Time via Decentralized Probabilistic Job Dispatching in Heterogeneous Multiple Computer Systems," The Computer J., vol. 41, no. 4, pp. 223-230, 1998.
[10] K. Li, "Minimizing the Probability of Load Imbalance in Heterogeneous Distributed Computer Systems," Math. and Computer Modelling, vol. 36, nos. 9/10, pp. 1075-1084, 2002.
[11] K. Li, "Optimal Load Distribution in Nondedicated Heterogeneous Cluster and Grid Computing Environments," J. Systems Architecture, vol. 54, nos. 1/2, pp. 111-123, 2008.
[12] K. Li, "Optimal Power Allocation among Multiple Heterogeneous Servers in a Data Center," Sustainable Computing: Informatics and Systems, vol. 2, pp. 13-22, 2012.
[13] K. Li, "Optimal Load Distribution for Multiple Heterogeneous Blade Servers in a Cloud Computing Environment," J. Grid Computing, vol. 11, no. 1, pp. 27-46, 2013.
[14] K.W. Ross and D.D. Yao, "Optimal Load Balancing and Scheduling in a Distributed Computer System," J. ACM, vol. 38, no. 3, pp. 676-690, 1991.
[15] Scheduling and Load Balancing in Parallel and Distributed Systems, B.A. Shirazi, A.R. Hurson, and K.M. Kavi, eds. IEEE CS Press, 1995.
[16] X. Tang and S.T. Chanson, "Optimizing Static Job Scheduling in a Network of Heterogeneous Computers," Proc. Int'l Conf. Parallel Processing, pp. 373-382, Aug. 2000.
[17] A.N. Tantawi and D. Towsley, "Optimal Static Load Balancing in Distributed Computer Systems," J. ACM, vol