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 - A Survey

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 problems 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 days 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. The remaining part of the paper consist of the following. Section 2 contains the load balancing overview. Section 3 contains all the related work on Load balancing. Section 4 speaks about the static load balancing. Section 5 speaks about dynamic load balancing. Section 6 provides the comparison analysis.

2. Types of Load Balancing

Load Balancing is classified based on how the process is allocated to the nodes and information passed to the nodes.

a) Based on System Load

- **Centralized approach:** In this approach, one particular system manages the workload among all the other systems in the network.
- **Distributed approach:** In this approach, each and every node collects information from other nodes and builds its own load vector. This is called local load vector and these local load vectors are used for decision making[4]. This approach can be best suited for cloud computing environment since it uses a distributed approach.
- **Mixed approach:** This is the combination of the above two approaches that can make use of the advantages of those approaches.

b) Based on System Topology

- **Static approach:** When the system is designed and implemented this approach can be used. Where a single system is modelled to handle the other systems in the network.
- **Dynamic approach:** In this approach the systems are modelled to take decisions based on the current status of the systems in the network when the load is needed to be balanced. This method is more suitable for cloud computing environment since they are widely distributed.
- **Adaptive approach:** In this system the load is distributed based on the status changes. The parameters are changed dynamically and the algorithms can also be changed dynamically. This provides better performance during the status changes. This is also more suitable for cloud computing environment.

3. Related Work on Load Balancing Techniques

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.

A 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.

Load Balancing in Distributed System using Genetic Algorithm

The main goal of load balancing is to equalize the workload among the nodes by minimizing execution time, communication delays, maximizing resource utilization and throughput[7]. The scheduling in distributed system is NP-complete problem even in best conditions, and methods based on heuristic search have been proposed to obtain optimal and suboptimal solutions.

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 [8]. 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.

Load Balancing in Distributed Systems: An Approach Using Cooperative Games

The static load balancing problem in single class job distributed systems as a cooperative game among computers. It is shown that the Nash Bargaining Solution (NBS) provides a Pareto optimal allocation which is also fair to all jobs. For this game an algorithm for computing NBS is derived[9]. The framework is provided by cooperative game theory. The Nash Bargaining Solution provides a Pareto optimal operation point for the distributed system. The main goal was to derive a fair and optimal allocation scheme. This is used only for single job distributed system.

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. 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.

A Novel Load Balancing Model using RR Algorithm for Cloud Computing

This paper is also based on public cloud were the cloud is partitioned to make it more efficient. In order to make the cloud more flexible and efficiently use the resources the cloud is further sub divided into small sub partitions [12].So that the load balancing can be much more simplified for multiple nodes. The workloads are distributed in an even manner. In order to provide better performance and reduce the response time the Round Robin technique is used.

Towards Efficient Load Balancing and Green IT Mechanisms in Cloud Environment

Cloud Provides on-demand Services and resources, based on pay per use mechanism. Here optimizing the resource and to reduce the number of system used is a challenging task. In this paper they have provided with an algorithm that is

known has Adaptive Earliest Deadline First Algorithm (AEDF).In order to achieve Green Computing by efficiently utilizing the resources [13]. In Dynamic Resource allocation the decisions are made based on the systems current status. Here there is no need to have any prior knowledge about the previous status of the system. This is the better approach then the static approach. But still there are some issues in dynamic resource allocation.

Self Organizing Clouds using Multi-Attribute Resource Allocation

It is connected to a large number of computers by using P2P connectivity [14]. Each and every computer can act either has a resource provider or consumer or both. It has a bounded delay in locating the node that can satisfy the users task demand. Two algorithms are used here.

1. Dynamic Optimal Proportional Share
2. Multi Range Query Protocol

Dynamic Optimal Proportional Share (DOPS): the algorithm can redistribute the resources that are available to the tasks that are running dynamically, so that they can use the maximum capacity of the resource in the node.

Multi Range Query Protocol: In SOC environment the Multi Range query Protocol is used to find the qualified nodes.

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.

4. Static Load Balancing

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. 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.

Round Robin Algorithm: It distributes the job among the slave processors. It has a RR order based on which the jobs are assigned to the slave processors, such that the processors are choose in series. When the job is allocated to the last processor and then next will be the first processors turn to take another job. Inter process communication is not needed in RR. It works well only when the jobs have equal processing time. If they have uneven processing time some nodes suffer from overload and the others may be idle.

Threshold Algorithm: Here each and every node has a private copy of the system workload. The loads can be divided into three levels: Overloaded, Medium, Under Loaded. Initially all the processors are set to be under loaded. Here there are two threshold parameters t_upper and t_lower.Under loaded: load < t_lower, Medium: t_lower ≤ load ≤ t_upper,Over loaded: load > t_upper. Each time a processor exceeds the load limit. A message is sent to all the remote processors regarding the new load, so that the entire system is uploaded regularly. The disadvantage of this algorithm is that, when the remote processors are overload the local processors are allocated with jobs.

Randomized Algorithm: This algorithm randomly chooses a number and selects the slave processor. The random numbers can be generated using a static distribution. It can achieve a good performance when it is used for a special purpose application.

Central Manager Algorithm: Here there is a system that acts as a central manager. This system controls the central processor that will choose a slave processor to assign a job. The processor with the least load is selected has a slave processor. All the information regarding the processor load is collected by the central processor to provide an efficient use of their resources. But the problem is, when the system reaches bottle neck due to high inter process communication. This algorithm can be used when different hosts create different dynamic activities.

5. Dynamic Load Distribution

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 [15]. 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[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**.**

6. Comparison Analysis: 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.
Technique:
1. Virtualization technology
2. Skewness

Main Goals are:

a) **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.
b) **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.
c) **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.
d) **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.
e) **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. If the system is in hot spot then a new process arriving must be migrated to some other system in the network.
f) **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. When the server is in cold spot and if its resources are no longer needed, that server can be turned off.

7. Conclusion

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, skewness algorithm can be 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.

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.