Optimization of load distribution and balancing Over multiple server in cloud

M. Baby Jasmine [1], N. Subbulakshmi[2]
P.G Scholar, Department of IT, Dr. Sivanthi Aditanar College of Engineering, Tiruchendur, India[1]
Asst. Professor, Department of IT, Dr. Sivanthi Aditanar College of Engineering, Tiruchendur, India[2]

Abstract: Performance of the cloud of can be optimized by load distribution and balancing. Energy efficiency is one of the most important issues for large-scale server systems in current and future data centers. The multi service processor technology provides new levels of performance and energy efficiency. The present paper aims to develop power and performance constrained load distribution methods for cloud computing in current and future large-scale data centers. In particular, we address the problem of optimal power allocation and load distribution for multiple heterogeneous multi server processors across clouds and data centers. Our strategy 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 research problems in large-scale data centers are well-defined multivariable optimization problems, which explore the power-performance tradeoff by fixing one factor and minimizing the other, from the perspective of optimal load distribution. It is clear that such power and performance optimization is important for a cloud computing provider to efficiently utilize all the available resources.
.

Keywords: Load distribution, load balancing, multi server processor, partition, Cloud cluster

I. Introduction

A. Cloud Computing

Cloud computing has become very popular in recent years as it offers greater flexibility and availability of computing resources at very low cost. The major concern for agencies and organizations considering moving the applications to public cloud computing environments is the emergence of cloud computing facilities to have far-reaching effects on the systems and networks of the organizations. Many of the features that make cloud computing attractive, however, can also be at odds with traditional security models and controls. As with any emerging information technology area, cloud computing should be approached carefully with due consideration to the sensitivity of data. Planning helps to ensure that the computing environment is as secure as possible and is in compliance with all relevant organizational policies and that data privacy is maintained. It also helps to ensure that the agency derives full benefit from information technology spending. The security objectives of an organization are a key factor for decisions about outsourcing information technology services and, in particular, for decisions about transitioning organizational data, applications, and other resources to a public cloud computing environment. To maximize effectiveness and minimize costs, security and privacy must be considered from the initial planning stage at the start of the systems development life cycle. Attempting to address security after implementation and deployment is not only much more difficult and expensive, but also more risky. Cloud providers are generally not aware of a specific organization's security and privacy needs. Adjustments to the cloud computing environment may be warranted to meet an organization's requirements. Organizations should require that any selected public cloud computing solution is configured, deployed, and managed to meet their security and privacy requirements. Every industry standard states that there should be security and privacy policies. Security policies are the top-level set of documents of a company. They document company's decision on the protection, sharing and use of information. A complete and appropriate set of policies will help to avoid liability and compliance problems. Cloud computing has been defined by NIST as a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or cloud provider interaction[1]. Cloud computing technologies can be implemented in a wide variety of architectures, under different service and deployment models, and can coexist with other technologies and software design approaches. The security and privacy challenges cloud computing presents, however, are formidable, especially for public clouds whose infrastructure *and computational resources are owned by an outside public cloud.*

B. *Cloud Service Models*

Cloud service delivery is divided into three models. The three service models are :

a. *Cloud Software as a service (Saas)*

The capability provided to the consumer is to use the providers applications running on a cloud infrastructure. The applications are accessible from various client devices through a thin client interface such as a web browser. The consumer does not manage the underlying cloud infrastructure.

b. *Cloud Platform as a Service (Paas)*

The capability provided to the consumer is to deploy onto the cloud infrastructure consumer created or acquired applications created using programming languages and tools supported by the provider. The consumer does not manage or control the underlying cloud infrastructure, but has control over the deployed applications and possibly application hosting environment configurations.

c. *Cloud Infrastructure as a Service (Iaas)*

The capability provided to the consumer is to provision processing, storage, networks, and other fundamental computing resources where the consumer is able to deploy and run arbitrary software, which can include operating systems and applications. The consumer does not manage or control the underlying cloud infrastructure but has control over operating systems, storage, deployed applications, and possibly limited control of select networking components. Features and parts of IaaS.

II. Related Work

Cloud is made up of massive resources. Management of these resources requires efficient planning and proper layout. While designing an algorithm for resource provisioning on cloud the developer must take into consideration different cloud scenarios and must be aware of the issues that are to be resolved by the proposed algorithm. Therefore, resource provisioning algorithm can be categorized into different classes based upon the environment, purpose and technique of proposed solution load balancing algorithm, when a new request is received at the central server it asks for the real time load.

A. *Load Balancing on the basis of Cloud Environment*

Cloud computing can have either static or dynamic environment based upon how developer configures the cloud demanded by the cloud provider. Cloud computing technologies can be implemented in a wide variety of architectures, under different service and deployment models.

B. *Static Environment*

In static environment the cloud provider installs homogeneous resources. Also the resources in the cloud are not flexible when environment is made static. In this scenario, the cloud requires prior knowledge of nodes capacity, processing power, memory, performance and statistics of user requirements. These user requirements are not subjected to any change at run-time. Algorithms proposed to achieve load balancing in static environment cannot adapt to the run time changes in load. Although static environment is easier to simulate but is not well suited for heterogeneous cloud environment. Round Robin algorithm [1] provides load balancing in static environment. In this the resources are provision.

First-Come-First-Serve (FCFS- i.e. the task that entered first will be first allocated the resource) basis and scheduled in time sharing manner. The resource which is least loaded (the node with least number of connections) is allocated to the task. Eucalyptus uses greedy (first-fit) with round-robin for VM mapping. In this proposed system improved algorithm over round robin called CLBDM (Central Load Balancing Decision Model) [14]. It uses the basis of round robin but it also measures the duration of connection between client and server by calculating overall execution time of task on given cloud resource.

C. *Dynamic Environment*

In dynamic environment the cloud provider installs heterogeneous resources. The resources are flexible in dynamic environment. In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. The requirements of the users are granted flexibility (i.e. they may change at run-time). Algorithm proposed to achieve load balancing in dynamic environment can easily adapt to run time changes in load.

Dynamic environment is difficult to be simulated but is highly adaptable with cloud computing environment. Based on WLC [12] (weighted least connection) algorithm, Ren proposed a load balancing technique in dynamic environment called ESWLC. It allocates the resource with least weight to a task and takes into account node capabilities. Based on the weight and capabilities of the node, task is assigned to a node. LBMM (Load Balancing Min-Min) algorithm proposed in paper [6] uses three level frameworks for resource allocation in dynamic environment. It uses OLB (opportunistic load balancing) algorithm as its basis. Since cloud is massively scalable and autonomous, dynamic scheduling is better choice over static scheduling. . In this scenario cloud cannot rely on the prior knowledge whereas it takes into account run-time statistics. The requirements of the users are granted flexibility.

III. LOAD BALANCING IN CLOUD COMPUTING

Load Balancing is a method to distribute workload across one or more servers, network interfaces, hard drives, or other computing resources. Typical data center implementations rely on large, powerful (and expensive)computing [6] hardware and network infrastructure, which are subject to the usual risks associated with any physical device, including hardware failure, power and/or network interruptions, and resource limitations in times of high demand. Load balancing in the cloud differs from classical thinking on load-balancing architecture and implementation by using commodity servers to perform the load balancing. This provides for new opportunities and economies-of-scale, as well as presenting its own unique set of challenges .performance of the cloud can be optimized by the load balancing and load distribution.

Load balancing is used to make sure that none of your existing resources are idle while others are being utilized. To balance load distribution, you can migrate the load from the *source nodes* (which have surplus workload) to the comparatively lightly loaded *destination nodes*. When you apply load balancing during runtime, it is called *dynamic load balancing* — this can be realized both in a direct or iterative manner according to the execution node selection:

A. Goals of Load balancing

As given in [4], the goals of load balancing are:
1. To improve the performance substantially.
2. To have a backup plan in case the system fails even partially.
3. To maintain the system stability.
4. To accommodate future modification in the system.

IV. EXISTING SYSTEM

The important things to reflect on while developing any load balancing algorithm are: estimation and comparison of load, stability and performance of system, interaction between the nodes, nature of work, selection of nodes, etc. In this work we have implemented two load balancing algorithm using Window Azure Framework. To understand these load balancing algorithm let us consider an example of a cloud with five servers as shown in figure 2, where we assume that each request from the client is send to any of the servers using central node or central server. Performance of the cloud can be optimized by the load balancing and load distribution energy efficiency is one of the most important issues for the current and future data centers in the cloud.

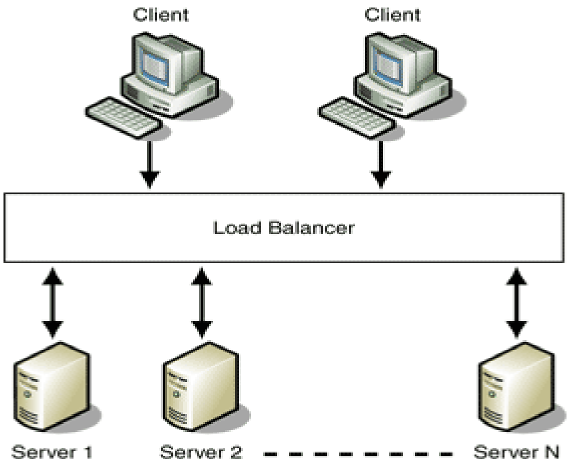

Figure 1: Load Balancing in a Cloud System

In this first proposed load balancing algorithm, consider that a new client request is received at the central server. Now the central server asks each of the servers in the cloud with their real time load. On receiving them the central server assigns this new request to the server with minimum load. In case of a tie it randomly assigns the request to any of the servers. This load balancing algorithm is a dynamic and extremely efficient, but requires for each new request the real time load to be calculated and estimated to the central server, which increases some overhead on the system.

In the proposed distributed load balancing algorithm, when a new request is received at the central server it asks for the real time load to each of the servers in the cloud. It then waits for the N requests to come hereafter. The value of this window size ‘N’ can be changed as per the requirement of the system. After waiting for the N new requests, the central node distributes these requests equally among all the servers in the cloud depending upon their load values. This load balancing algorithm is a more efficient one as it requires less computation at each server end.

A. Centralized Load Balancing

In centralized load balancing technique all the allocation and scheduling decision are made by a single node. This node is responsible for storing knowledge base of entire cloud network and can apply static or dynamic approach for load balancing. This technique reduces the time required to analyze different cloud resources but creates a great overhead on the centralized node. Also the network is no longer fault tolerant in this scenario as failure intensity of the overloaded centralized node is high and recovery might not be easy in case of node failure.

V. PROPOSED SYSTEM

The huge cloud environment consist of numerous node and it is partitioned into an n clusters based on our Cloud clustering technique. Our proposed model consists of main controller which controls all load balancer in each cloud cluster. The main controller maintains all the details include index table, its current status information of all load balancer in each cluster. The index table consists of both static parameters (number of CPU, Processing speed, memory size etc.) and dynamic parameters (network bandwidth, CPU and memory utilization ratio). All load balancer should refresh their status for every T. When the job arrives to main controller in cloud environment the initial step is to choose the correct cloud cluster and follows the algorithm as 1. Index table is initialized as 0 for all nodes which are connected to the central controller in cloud environment. 2. When the controller receives any new request from client, the controller queries the load balancer of each cluster for the job allocation. 3. The controller passes the index table to find the next available node which having less weight. If found, continues the processing. If not found the index table is reinitialized to 0 and in an increment manner, then controller passes the table again to find the next available node. 4. After finishing the process the load balancer update the status in the allocation table which is maintained by the controller.

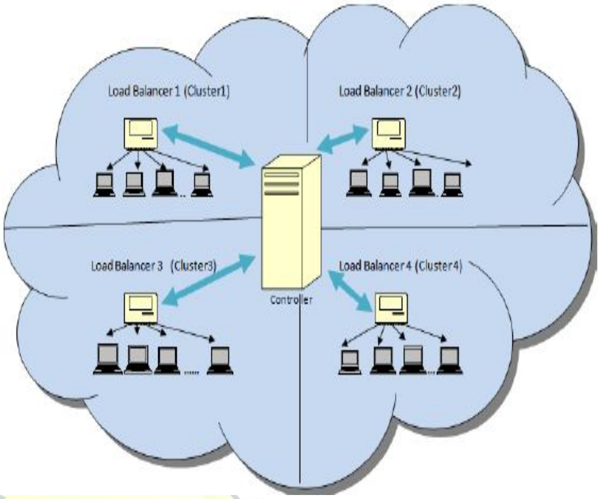
Load Balancer 1 (Cluster1)
Load Balancer 2 (Cluster2)
Load Balancer 3 (Cluster3)
Load Balancer 4 (Cluster4)
Controller

Figure 2: System Architecture

A. Distributed Load Balancing

Nodes in the cloud are highly distributed. Hence the node C that makes the provisioning decision also governs the category of algorithm to be used. There can be three types of algorithms that specify which node is responsible for balancing of load in cloud computing environment. In distributed load balancing technique, no single node is responsible for making resource provisioning or task scheduling decision. There is no single domain responsible for monitoring the cloud network instead multiple domains monitor the network to make accurate load balancing decision. Every node in the network maintains local knowledge base to ensure efficient distribution of tasks in static environment and re-distribution in dynamic environment. In distributed scenario, failure intensity of a node is not neglected. Hence, the system is fault tolerant and balanced as well as no single node is overloaded to make load balancing decision. Comparison of different static and dynamic load balancing algorithms is given in Table 3. It also compares them on the basis of spatial distribution of nodes. A nature inspired solution is presented in paper [7] called Honeybee Foraging for load balancing in distributed scenario. In Honeybee foraging the movement of ant in search of food forms the basis of distributed load balancing in cloud computing environment. This is a self organizing algorithm and uses queue data structure for its implementation. Biased random sampling [8] is another distributed load balancing technique which uses virtual graph as the knowledge base.

B. Cloud Clustering Technique

The cloud environment includes numerous node and the nodes are different geographical location. Portioning of large cloud into cluster into cluster helps to manage its performance effectively. Taking large cloud environment into consideration visit each node at a random order. This cloud portioning method includes 2 steps partitioning methods are: Step 1: a) Visit each node at random we matches it with the neighbour node. When it has same characteristics and shares similar candidate's data with minimal cost Once matched, two nodes are combined into new node, with that two node share same candidate details. b) Repeat the same until there is no one hop neighbour node having similar character tics. c) Subsequently cost between neighbour two nodes and current neighbour node has to be updated. Step2: Visit each node and join two neighbouring node into new node having same character tics. The visited node sends the information to new merged node instead of sending it twice to the previous node. So the cost between visited node and merged node are saved.

a. Partitioning-based

In such algorithms, all clusters are determined promptly. Initial groups are specified and reallocated towards a union. In other words, the partitioning algorithms divide data objects into a number of partitions, where each partition represents a cluster. These clusters should the following requirements: (1) each group must contain at least one object, and (2) each object must belong to exactly one group. In the K-means algorithm, for instance, a center is the average of all points and coordinates representing the arithmetic mean. In the K-medoids algorithm, objects which are near the center represent the clusters. There are many other partitioning algorithms such as K modes, PAM, CLARA, CLARANS and FCM. The main advantage of this approach is its fast processing time, because it goes through the dataset once to compute the statistical values for the partition. The accumulated grid-data make grid-based clustering techniques independent of the number of data objects that employ a uniform partition to collect regional statistical data. Such a method optimizes the fit between the given data and some (predefined) mathematical model. It is based on the assumption that the data is generated by a mixture of underlying probability distributions.

Algorithm*:* *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.
Input:*k*: the number of clusters,
D: a data set containing *n* objects.
Output: A set of *k* clusters.
Method:
(1) arbitrarily choose *k* objects from *D* as the initial cluster centers;
(2) repeat
(3) (re)assign each object to the cluster to which the object is the most similar based on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each cluster;
(5) until no change

VI. CONCLUSION

Load Balancing is an essential task in Cloud Computing environment to achieve maximum utilization of resources. In this paper, we discussed various load balancing schemes, distributed load balancing of algorithm provides better fault tolerance but requires higher degree of replication and divide the load at different levels of hierarchy with upper level nodes requesting for services of lower level nodes in balanced manner. Hence distributed environment provide better performance. However, performance of the cloud computing environment can be further maximized clustering technique divides the cloud environment into multiple partitions and simplifies the process load balancing effectively. The algorithm used in this paper is able to automatically supervise the load balancing work through load balancer assigned to each cluster. Multi service technique in which each server only handles a specific type of task, and each client requests a different type of service at a different time. The load balancer assigns clients requests for service tasks to server cluster and then each cluster assigned the task to the servers within its server cluster.

REFERENCES

[1]. Gaochao Xu, Junjie Pang, and Xiaodong Fu* "A Load Balancing Model Based on Cloud Partitioning for the Public Cloud ", ISSNl l1007-0214l l04/12l lpp34-39,Volume 18, Number 1, February 2013.

[2]. Jaspreet kaur ," Comparison of load balancing algorithms in a Cloud" (IJERA) ISSN: 2248-9622 , Vol. 2, Issue 3, May-Jun 2012, pp.1169-1173.

[3]. Z. Chaczko, V. Mahadevan, S. Aslanzadeh, and C.Mcdermind, Availability and load balancing in cloud computing, presented at 2011 International Conference on Computer and Software Modelling Singapore, 2011.

[4]. M. randles, D. Lamb, and A. Taleb-Bendiab, A comparative study into distributed load balancing algorithms for cloud computing, in Proc. IEEE 24th International Conference on

Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 551-556.

[5]. Microsoft Academic Research, Cloud Computing, computing?query= cloud%20computing, 2012.

[6]. Ajay Gulati, Ranjeev.K.Chopra "Dynamic Round Robin for Load Balancing in a Cloud Computing" IJCSMC, Vol. 2, Issue. 6, June 2013, pg.274 – 278.

[7]. N.Chandrakala Dr. P.Sivaprakasam " Analysis of Fault Tolerance Approaches in Dynamic Cloud Computing ", IJARCSSE, Volume 3, Issue 2, February 2013.

[8]. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: ―Cost Optimal Scheduling in Hybrid IaaS Clouds for Deadline Constrained Workloads. In: 3rd IEEE International Conference on Cloud Computing, Miami (July 2010).

[9]. Wang, S. C., Yan, K. Q., Liao, W. P., Wang, S. S.: Towards a Load Balancing in a three level cloud computing network. In: Computer Science and Information Technology, pp. 108—113, (2010).

[10]. R. M. Bryant and R. A. Finkel, "A Stable Distributed Scheduling Algorithm," in Proc. 2nd Int. Conf. Dist Comp., pp. 341-323, April 1981.

[11]. Grosu and A. T. Chronopoulos," Noncooperative Load Balancing in Distributed Systems," Journal of Parallel and Distributed Computing, vol. 65, no. 9, pp. 1022-1034, Sept. 2005.

[12]. Y. Chow and W. Kohler, "Models for Dynamic Load Balancing in Heterogeneous Multiple Processor System," IEEE Transactions on Computers, Vol. C-28, pp. 354-361, , May 1979.

[13]. L. Ni, and K. Hwang, K., "Optimal Load Balancing in a Multiple Processor System with Many Job Classes,"

About Author

M. Baby Jasmine received the B.E degree in Computer Science Engineering from Anna University, Chennai, in 2013. Currently, she is pursuing M.Tech degree in Information Technology from Anna University, Chennai. Her research areas of Interests include Data Mining, Cloud Computing, and Operating System.

N. Subbulakshmi has completed B.E in Computer Science Engineering from Jayamatha Engineering College in 2003 and M.Tech in Information Technology from Manonmaniam Sundaranar University in 2005. She is now working as an Assistant Professor for 9 years in Dr.Sivanthi Aditanar College of Engineering, Tiruchendur. Her research areas are Data Mining and Image Processing