International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 5, Issue 3, May-June 2016 **ISSN 2278-6856**

Load Balancing Algorithm for Cloud Computing Using Distributed Data-Centers

Rajkumar Sharma

Computer Centre, Vikram University, Ujjain (MP) India

Abstract

Virtualization techniques, high speed internet access and Service Oriented Architecture (SOA) allow users to access computing resources on pay-as-you-go basis under Cloud environment. The organizations neither need to purchase expensive hardware such as servers, storage, networking equipments etc. nor require skilled manpower for in-house software development. Cloud computing provides scalable and cost-effective solution at a nominal initial investment. Distributed data-centers in Cloud environment run user's applications with minimum latency by exploiting computing resources geographically near their locations. Load balancing algorithms play important role in equalizing load among data-centers and in efficient use of computing resources. In this paper, performance of a dynamic load balancing algorithm has been evaluated by dividing data-centers in different zones. It has been shown that the proposed algorithm improves the computing efficiency of data-centers and minimizes the response time of user's applications.

Keywords- Cloud computing; Latency; Load balancing

1. introduction

Effective and efficient use of underlying hardware resources has always been the target of computing researchers in their contemporary time. Starting from mainframes to grid computing and to recent virtual machines on 'Clouds', computational history experienced several combinations of architecture for the efficient utilization of computing resources. Some architectural designs use resources in centralized manner while others use it in decentralized manner [3]. Grid computing gained much attention from scientific computing community whereas Cloud computing is being penetrated gradually to commercial application users.

1.1 Cloud Computing

Cloud computing has become a leading edge technology as it provides dynamically scalable and virtualized resources as a service over the network at a nominal initial investment. Economically, the main attraction from Cloud computing is that customers use only what they need and pay for what they actually use [1]. Resources as a service are available over Cloud at any time and from any location via the Internet [10]. Three main types of service levels as delivery models are:

Software as a Service (SaaS): The clients may opt for ready customized application, but do not have control over background environment such as operating system, hardware or network parameters.

Platform as a Service (PaaS): In this types of services, clients have control over change in application and hosting environment such as system software. But PaaS does not provide control over operating system, hardware and network parameters.

Infrastructure as a Service (IaaS): The clients can create a virtual processing environment by specifying choice of processing power, storage, network parameter etc. and have control over operating system and application environment.

One can choose services from pool of available services and negotiate price through Service Level Agreements (SLAs). Among the popular Cloud service providers are: Amazon [7], Google [8], Microsoft etc. Data-center works as backbone in Cloud computing where a large number of servers are networked to host computing & storage needs of the users. The area which needs more attention is Latency Optimization for cloud architecture to work more efficiently [3]. Many data intensive applications produce enormous amounts of data which travel on cloud network. As the cloud users grow, cloud architecture should accommodate movement of voluminous data to avoid data congestion in the network.

1.2 Virtualization Technology

With virtualization techniques, multiple operating systems can concurrently run on a single physical system. A user can opt for his choice of operating system and other hardware configuration called virtual machine (VM) and run his application by sharing underlying hardware resources.

It is the 'Virtual Infrastructure Management Software' (VIMS) that centrally manages many VMs on a single physical system. In user's perception, each VM is a single, logical bunch of resources as shown in Figure 1. Virtualization techniques provide cost-effective & efficient utilisation of IT infrastructure. Presently, Xen (http://www.xen.org) [11] and VMWare (http://www.vmware.com) [12] are two leading virtualization technology providers.

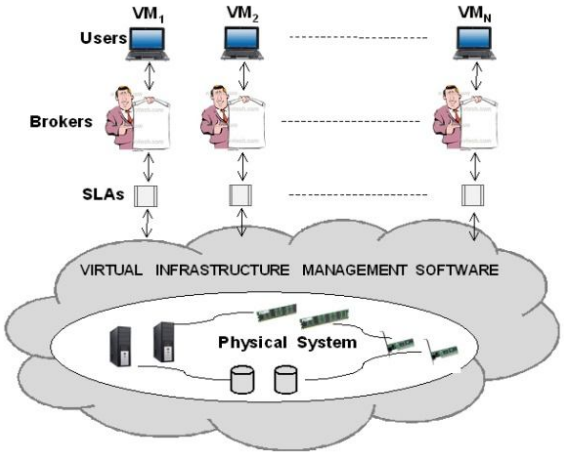

Fig. 1 Virtual Machines in Cloud Architecture

1.3 Load Balancing in Cloud

On user's request, instance is created as virtual machine on underlying hardware present in one of the data-centers. It is the situation sometimes that there is long queue for creating instances on one data-center whereas resources in other data-centers are idle. Load balancing plays in important role to overcome this situation by equalizing loads among data-centers leading to efficient utilization of available resources. The two categories for load balancing algorithms are static and dynamic [13]. In static load balancing algorithms, the formation of instance at data-center is done at compile time. No recent load status of data-center is considered while forming instances. The advantage in this sort of algorithm is the simplicity, both in implementation and overhead, as no current performance statistics is required. In dynamic load balancing algorithms, the formation of instance at data-center is done at run time. The current load information of data-centers is used while forming instances. However, in this case the overhead of collecting and maintaining of load information is additional. In this paper, performance of a dynamic load balancing algorithm has been evaluated by dividing data-centers in different zones. It has been shown that the proposed algorithm improves the computing efficiency of data-centers and minimizes the response time of user's applications.

2. Related Work

The issue of Cloud computing architecture is being addressed by many researchers. A common trend found in almost all architectures is the use of centralized resources at the Cloud provider's location, leading to increase in latencies. A local stream server and a cloud stream server was used by Wilhelm et al. to reduce execution time [2]. Very little improvement was found although overhead of maintaining two parallel server was involved. A strategy to manage complex and unpredictable workload entering cloud is proposed by Paton et al.[5]. As all computing & storage resources are managed centrally, the system is susceptible to network congestion, despite workload balance. Dabas & Gupta propose a Cloud architecture for radio frequency identification [6]. The data captured by radio frequency reader is sent to data processing system present in the cloud. A substantial time delay is observed if radio frequency reader and cloud resources are physically located at long distance. Padhy and Rao[4] distributed all application tasks across nodes in Cloud and observed a substantial latency in aggregating tasks from all geographically remote locations.

3. Proposed Algorithm

Cloud service provider may be located far away from the clients, compelling data to travel from several mediums and network equipments, thereby imposing a time delay in getting Cloud services. Existing Cloud providers use centralized data-center to host computing & storage needs of the clients. Due to world-wide hype and rapid growth in associated technologies, Cloud computing clients continue to multiply. The large number of service requests to fulfill the demands of millions of users will broaden the latency problem. In previous study[3], we proposed an intelligent energy efficient Cloud computing architecture based on distributed data-centers which form a client's instance in nearest neighborhood and fulfill client's request in optimized latency. In this study, we propose a dynamic load balancing algorithm by dividing data-centers in different zones as shown in Figure 2.

Data-centers work in master-slave paradigm. Nearest data-centers form a computing zone and users may opt for creating their instances in multiple zones. The main entities involved in proposed architecture are :

Matser/Slave Data-Center: Master data center is located at Cloud provider's administrative premises. User's accounting on pay-as-you-go basis is completed here. Slave data-center are geographically scattered to serve user's requests in minimum physical distance.

Users/Brokers: Users communicate directly or via brokers to submit requests which automatically reaches at master data-center. Master data-center creates user instance at appropriate slave data-center minimum latency.

Service Level Agreements (SLAs): Quality of Service (QoS) and pricing negotiations are settled through SLAs. Master data-center scans SLA each time to host needs of the users.

3.1 Informal Description of Algorithm

To maximize the performance of cloud applications in a computing environment consisting of distributed data-centers, it is necessary to minimize the idle time of individual data-center and also suffering of data-centers from overload of work. In the proposed algorithm, we monitor load at regular time interval for all data-centers within a zone and store it in a load vector. We decide lower and upper threshold values for lightly loaded and heavily loaded data-center respectively. Whenever the load on a data-center is above threshold value, arriving new request for creating instance is to be migrated to a lightly loaded data-center which is located within the same zone.

As all the data-centers of this zone are near to user's physically location, his cloud application gets executed with minimum latency. If the load status of all data-centers is above threshold value, i.e., remaining data-centers of this zone are heavily loaded then algorithm tries to form instance to a data-center belongs to another zone, if fails, keeps on searching all zones until it finds desired lightly loaded data-center. With this mechanism we try to maximize the efficiency of data-centers as well as minimize the latency, which results in reducing overall cloud application response time.

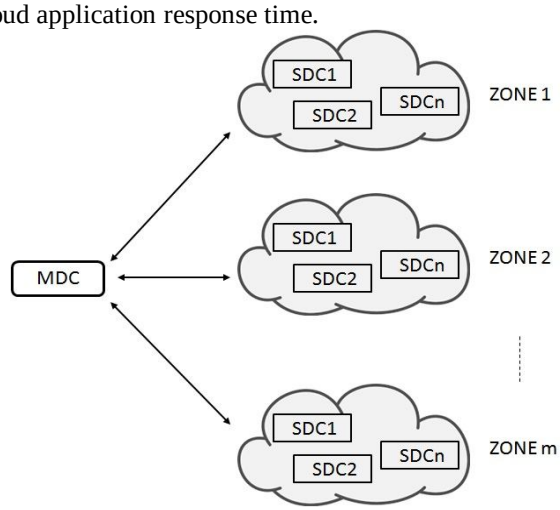

Fig. 2 Proposed Cloud Architecture

3.2 Formal Description of Algorithm

We consider M zones each having N data-centers, as shown in Figure 2. If we denote:

SDC_{ij} - Slave Data-Center in i^{th} zone and at j^{th} location (i=1...M, j=1...N),

MDC - Master Data-Center,

U_{kq} - An user in physical neighborhood of q^{th} slave data-center present within k^{th} zone,

LD_{ij} - Queue length at j^{th} slave data-center of i^{th} zone which is maintained periodically.

The proposed dynamic load balancing algorithm is described below:

Algorithm : Create_Instance

Request for instance from user $U_{kq} \rightarrow$ MDC

MDC $\rightarrow$ if (LD_{ij} < Upper-Threshold Value) (i = k, j = q)

create instance (U_{ij}) at SDC_{ij} ;

else (LD_{ij} < Upper-Threshold Value) (i = k, j = 1...N, j ≠ q)

create instance (U_{ij}) at SDC_{ij} ;

else if multiple zones = 'yes' and (LD_{ij} < Upper-Threshold Value) (i = 1..M, i ≠ κ, j = 1...N)

create instance (U_{ij}) at SDC_{ij} ;

else make a fresh request $U_{kq} \rightarrow$ MDC ;

End of algorithm.

4. Experimental Results and Comparison

We simulated our experiments [9] with several zones containing data-centers having different communication latencies, instance user requests from different geographic locations, inter-zone and within zone instance migration etc. We experimented and compared a cloud application by applying four different strategies described below :

With no load balancing & random data-center (WLBR): After receiving request from user to create instance, MDC forwards this request to a random data-center. No load balancing is applied in this strategy.

With no load balancing & with data-center in local zone (WLBZ): In this strategy, MDC forwards request to user's geographically nearest data-center. No load balancing is applied.

Dynamic Load Balancing within single zone (DLBSZ): Load balancing is applied in this strategy. Users' request to create instance is migrated within local zone only.

Dynamic Load Balancing with multiple zones (DLBMZ): Load balancing is applied here. To create user's instance, first a data-center within local zone is searched having queue length less than upper threshold value. If resources are not free, all remaining zones are searched for instance creation.

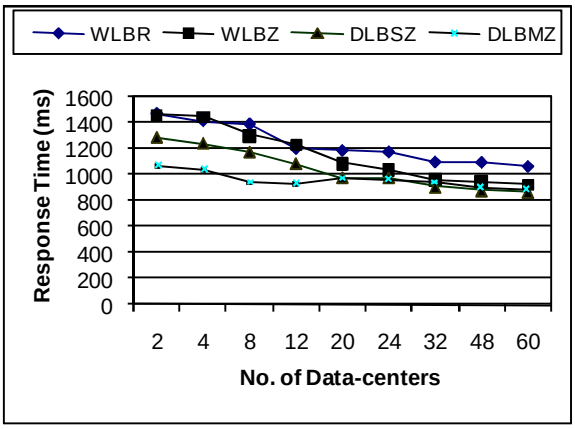

Fig. 3 Comparative Performance of Four Strategies

We observe that enforcement of dynamic load balancing provides better results than strategies with no load balancing. However, WLBZ outperforms WLBR due to less communication latencies within local zone as shown in Figure 3. DLBMZ yields minimum response time out of all strategies. DLBSZ provides similar results when number of data-centers are below 20. We also analyzed number of migrations across data-centers. We observed that migrations decrease as number of data-centers increases as shown in Figure 4. By running applications from different geographic locations, we compared the average CPU and memory utilization of data-center with dynamic load balancing and with no load balancing. We observed about 8% to 10% improvement in resource utilization by using dynamic load balancing algorithm as shown in Figure 5 and Figure 6.

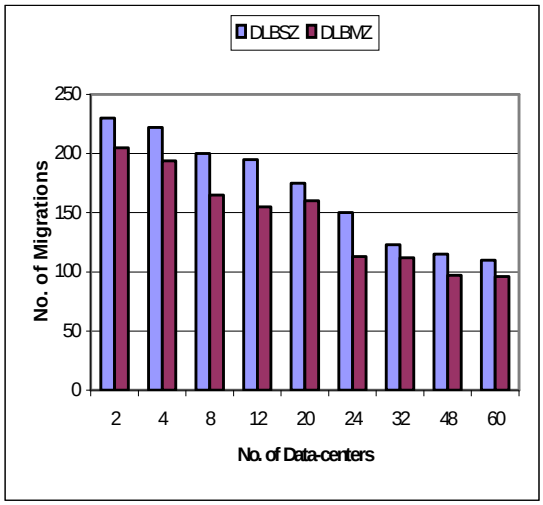

Fig. 4 No. of Migrations with single versus multiple zones

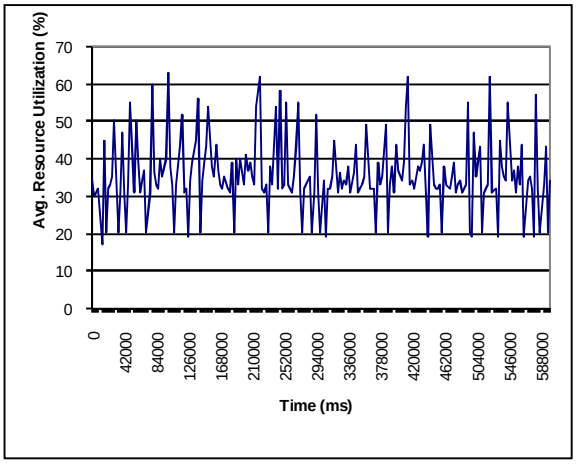

Fig. 5 Resource Utilization in Data-Centers With No Load Balancing

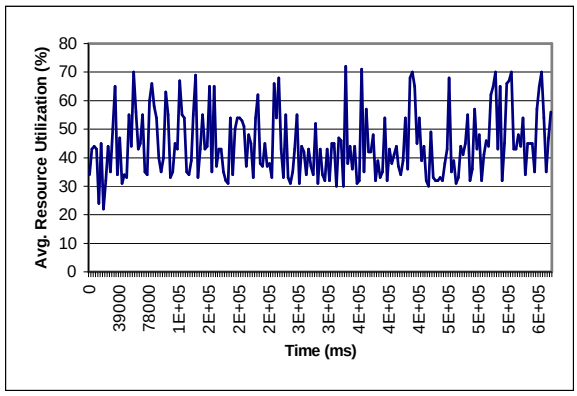

Fig. 6 Resource Utilization in Data-Centers With Dynamic Load Balancing

5. CONCLUSION

In recent time, Cloud computing has attracted significant attention from business community. At present the architectural design of Cloud computing is in its rudimentary phase and requires efficient resource utilization of data-centers. To accommodate growing number of Cloud users, its architectural design should avoid network data congestion as well as response to users in minimum latency. In this paper, a dynamic load balancing algorithm for distributed data-centers in Cloud is proposed which results in substantial gain in response time. The proposed algorithm reduces number of migrations for instance requests and improves resource utilization of data-centers. Many data intensive applications produce huge amount of data which travel on cloud network. As the proposed algorithm create users instance in geographically nearest data-center, it reduces network congestion across all data-centers of all the zones.

Reference

[1] Rajkumar Buyya, Chee Shin Yeo, and Srikumar Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality," Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, Dalian, China, Sept. 25-27, 2008.

[2] Wilhelm Kleiminger, Evangelia Kalyvianaki and Peter Pietzuch, "Balancing Load in Stream Processing with the Cloud," IEEE International Conference on Data Engineering," Germany, April 2011, pp 16-21.

[3] Rajkumar Sharma and Priyesh Kanungo, "An Intelligent Cloud Computing Architecture Supporting e-Governance," IEEE International Conference on Automation and Computing, University of Huddersfield, Huddersfield, UK, Sept. 2011, pp 1-4.

[4] Ram Prasad Padhy and Goutam Prasad Rao, "Load Balancing in Cloud Computing Systems," Thesis retrieved from http://ethesis.notrkl.ac.in

[5] Norman W. Paton, Marcelo de Aragao, Kevin Lee, Alvaro Fernandes and Rizos Sakellariou, "Optimizing Utility in Cloud Computing through Autonomic Workload

Execution," retrieved from http://research.microsoft.com/pub/debull/A09mar.

[6] Chetna Dabas and J.P Gupta, "A Cloud Computing Architecture Framework for Scalable RFID," International Multi Conference of Engineers and Computer Scientsts (IMECS 2010,) Hong Kong, Vol 1, March 2010, pp 217-220.

[7] Amazon Elastic Compute Cloud (EC2), http://www.amazon.com/ec2/

[8] Google App Engine, http://appengine.google.com/

[9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, Cesar A. F. De Rose, and Rajkumar Buyya, "CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing," Volume 41, Number 1, New York, USA, January, 2011, pp 23-50

[10] Suraj Pandey, "Cloud Computing Technology & GIS Applications," Asian Symposium on Geographic Information Systems From Computer & Engineering View (ASGIS 2010), Chine, April 2010.

[11] Virtualization Technology, http ://www.xen.com/

[12] Vitual Machines through VMware, http://www.vmware.com/

[13] R. Sharma, M. Chandwani and P. Kanungo, "A New Dynamic Load Balancing Algorithm based on Workstation Priority in NOW using MPI Environment ," Journal of the Institutions of Engineers, Volume 92, May 2011, pp 1-5.