Advances in Computer Science and Information Technology (ACSIT)
Print ISSN : 2393-9907; Online ISSN : 2393-9915; Volume 2, Number 7; April – June, 2015 pp 81 – 86
Krishi Sanskriti Publications
http://www.krishisanskriti.org/ACSIT.html

An Energy Aware Cloud Load Balancing Technique using Dynamic Placement of Virtualized Resources

Sumita Bose[1], Jitender Kumar[2]

[1]Department of computer science, Dcrustm, murthal
[1]sumita18singh@gmail.com

[2]Department of computer science, Dcrustm, murthal
[2]jitenderkbhardwaj@gmail.com

Abstract : *With increasing demand and popularity of large scale cloud datacenter, it has become very important to allocate the resources in such a way that not only leads to efficient utilization of resources as well as reduces the energy consumption. Cloud applications consume a lot of energy and thereby contribute to high operational costs and also affect the environment by carbon emission. To tackle this problem, we propose an energy efficient resource allocation framework that uses virtualization to allocate the data center resources based on the application demands. Our algorithm minimizes the total number of running servers and improves the resource utilization. Also we introduce a method for dealing with the problem of overloading while reducing the energy used. Through experimental evaluations it is demonstrated that the proposed algorithm is able to achieve the substantial reduction in energy consumption and improves the overall resource utilization.*

Keywords*: Cloud Computing, energy aware, resource utilization, virtual machine request, Green Computing*

1. INTRODUCTION

Cloud computing is emerging as a transformative computing trend in IT industry as it is changing the way in which business and other type of IT services are delivered. It has moved information data and computing away from desktop and portable PCs to large datacenters. Cloud computing provides virtualized and scalable resources as a service over the web on pay per use basis such as software as a service, infrastructure as a service etc. Different types of cloud computing models and projects have been tremendously beneficial to network applications, such as Amazon EC2 [1], Google Compute Engine [3], IBM Blue Cloud [2] etc.

In today's scenario, the demand for high performance computing infrastructure is growing day-by-day that has led to development of large scale datacenters consuming a very large amount of energy and thereby contributing to increased total operational costs [4]. Despite of all the improvements in the infrastructure and data centers energy consumption is still increasing. In a report[5] from NRDC, it is pointed that all the data centers in the United States are estimated to consume more than 75 billion kilowatt-hours of electricity annually, which is approximately 2% of total consumption of electricity consumed in US, increasing steadily. Much of this energy is wasted by idle or low utilized servers which consumes about 60% of the power when machine is fully utilized. This increasing consumption of energy not only increases the operational costs but also results in an enormous amount of carbon emission and therefore affecting the environment.

In data centers, there is an overprovision of computing, power distribution and cooling infrastructures to ensure high levels of reliability [6]. Therefore, the energy consumption is not balanced with the workload processing. In a parallel and distributing computing environment most of the load balancing approaches [7], the main focus is on how to balance the workload among the servers so as to maximize the throughput but these lack in case of energy considerations. When we deal with cloud computing such approaches can cause a large amount of energy consumption. Hence, there is a need to develop a resource allocation technique that not only improves the resource utilization but also reduces the amount of energy consumed.

1.1. Research Scope

In this paper, the focus is on energy aware load balancing. Our approach takes leverage of virtualization technology to allocate the resources on the basis of application demands. We present the design and implementation of a resource allocation scheme that not only improves the overall utilization but also is energy aware. The major contributions are summarized as follows:

- An energy-aware algorithm is proposed to allocate the resources in such way that optimizes the utilization while saving the energy.

- Also a mechanism is introduced for optimizing the thermal state of a server node by avoiding the overload.
- Through experimental results it is shown that the proposed algorithm is able to reduce the energy consumption by reducing the number of servers used and optimizing the utilization.

The rest of the paper is organized as follows. In Section2. Related work discussed and Section3. describes the assumptions and problem statement. Section4. details the energy aware resource allocation algorithm. Section5. contains experimental setup and compares results of proposed algorithm with some existing techniques. In Section6. paper is concluded and future work is discussed.

2. RELATED WORK

In this section, we discuss some relevant techniques proposed to deal with energy efficiency and resource utilization.

Initially, the main focus of algorithms has been on optimal allocation of jobs that has led research to algorithms like first fit, best fit [8]. In the first fit algorithm, job is allocated to the first server where the job's requirement can be fulfilled. The best fit algorithm places the request at the server where the job fits the best. First fit algorithm is also used as for comparisons in [9,10].Max-min, min-min, round robin and dynamic round robin algorithms are other widely adopted job allocation algorithms in the literature [12,14,13,]and assigns the resources guaranteeing a target processing time. But one of the major drawbacks of these techniques is that they are only concerned about resource allocation and managing the utility of resources so as to maintain the throughput. Another disadvantage is their inability to work in dynamic environment.

In [15, 16] power-aware algorithms are proposed. One is PALB which keeps track of state of all computing nodes and decide the number of computing nodes need to be in operating state on the basis of their respective utilization percentage. Also PALB works in three sections and each section has its own unique function. This approach considers the heterogeneous nature of local organization's cloud and could be applied to power aware cluster controller of a local cloud. Another one is a power aware virtual machine placement scheme for real time environment that manages the placement of the virtual machines using DVFS.

In [10, 17] energy aware application schemes are presented named as Ena-Cloud and EVISBP. In both techniques application placement problem is considered as a bin packing problem and also uses live migration. In Ena-Cloud main goal is to reduce the number of servers and also minimizing the number migrations. EVISBP i.e. Enhanced Variable Item Size Bin Packing also aims to reduce the number of servers. But the one main demerit of these algorithms is the migration overhead.

A number of researches [4, 18] have been done that not only claims to improve energy efficiency but also supports quality of services like SLAs etc. Zhen Xiao et al. [19] have discussed an energy aware approach that exploits the benefits of virtualization technology for allocating the datacenter resources dynamically based on the application requirements. In the paper, a load predictive algorithm is proposed to predict the pattern of future demands. Also it defines thresholds to detect hot spots and cold spots. Also it introduces the concept of skewness to balance the resource utilization of multi resources on server. In [20] Dzmitry Kliazovich et al. have presented a scheduling solution named as e-STAB which focuses on traffic requirements of cloud applications along with the role of communication fabric providing optimized energy efficient traffic load balancing in data center networks.

Nguyen Truing Hieu et al. [21] have proposed a virtual machine placement algorithm for large datacenters and named that as MAX-BRU algorithm. This algorithm aims to maximize the minimum load and for doing so it utilizes two matrices: resource utilization ratio and resource balance ratio. Some of the disadvantages of this algorithm are: it has not specified any method to deal with overloading. Also nothing has explained about how the algorithm will react in case when most of the servers get idle due to less amount of workload on them. In contrast to the above discussed studies, we designed an algorithm that provides a mechanism for maximizing the resource utilization along with improving the energy efficiency by reducing the number of servers actively in use. Also it introduces a method to optimize the thermal state of the servers by avoiding overload condition.

3. PROBLEM FORMULATION

In this section, we discuss the main characteristics goals and parameters of our energy aware resource allocation problem. Also it includes

3.1. Problem definition

The cloud computing environment consists of hundreds or even thousands of computing nodes. These computing nodes serve the application users by fulfilling their resource requirements and the requests won't be able to meet, they get rejected. For our system we assume that there is no rejection, all the coming requests for the resources will be mapped to suitable server.

In this paper, we consider the problem of energy efficient resource allocation problem as to allocate the resources on the basis of type of vm requests while taking care of the fact that no server will get overloaded. The main goals of our algorithms are to:

- Minimize the number of servers used
- maximize the resource utilization until upper threshold of utilization is reached
- Optimize the thermal state of each server

For achieving the above goals two thresholds: upper threshold and minimum threshold have been defined.
When utilization of all the running reaches upper threshold then we start the new server for mapping the requests.

To formulate the problem, we use properties in the form of t =< t_i, r >, where r denotes the remaining resource capacity of the server t_i, and t_i used to uniquely identify the server with resource capacity r. We have used two flag startedstatus to define the status of a server, the flag defines whether the server is started or not.

Utilization of a server can be defined as the amount of resources divided by the total resource capacity of the server i.e.

$$\text{Utilization percentage} = \frac{\text{amount of resources}}{\text{Total resource capacity of a server}} \times 100$$

In the proposed algorithm initially we try to place the new incoming request on already running servers providing that utilization of any server must not exceed the upper utilization threshold. The new server is started only if the already running servers are unable to meet the requirements of vm request.

3.2. System model

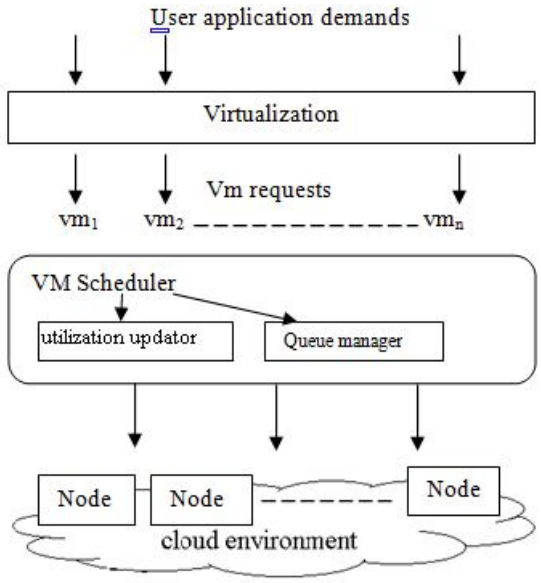


Figure 1. System Model

The proposed system model for our algorithm presented in figure 1. can be explained as:

Whenever user sends application demands requesting for datacenter resources than each and every request is encapsulated in vm requests with the help of virtualization. For each vm request in domain V, server from the resource pool of server nodes is allocated with the help of vm scheduler. The vm scheduler calculates the utilization of each running server and find the most suitable server for placing the vm request according the algorithm proposed and updates the utilization after each placement of vm request. It also ensures that upper threshold must not be crossed. The queue manager at vm scheduler maintains a queue of allocated vm requests for each server. No other request can placed on the server until these allocated queues of requests keep executing.

.

4. Algorithm and its details

The proposed algorithm is presented below in figure 2.

Energy efficient resource allocation algorithm

Input: v, type of vm request on the basis of size

Output: a resource allocation scheme

1. While V ≠ ∅ do
2. Satisfied Flag ←false;
3. Other Flag ←false
4. For each started server
5. If RemainingCap[t]> req & startedStatus[t] ←true)
6. if(util[t]<minThUtil||newutil< upThUtil)
7. Place the request on the server
8. Satisfied flag ←true
9. Update the utilization and remaining capacity of server t
10. If (Satisfied flag ←false)
11. Other flag← true
12. Start a new server t_{new}
13. Started status[t_{new}] ←true
14. Place the vm request on t_{new}
15. Satisfied flag ←true
16. Update the utilization and remaining capacity of t_{new}
17. Add the t_{new} to the list of started servers

Figure 2. Algorithm

Our basic idea is to place the vms in such a way that results in maximizing the resource utilization of each started server without exceeding the upper utilization of threshold. Also the new server is started if and only if when already set of started servers are unable to provide resources to the incoming vm request.

Here we have used several flags: Satisfied flag tells about whether the vm request is placed or not, Other flag is set true to start the new server, started status and used status defines the status of a server. The resource allocation procedure will continue until all incoming vm requests are met. Within the while loop, the suitable server is selected to place the vm request.

When any vm request arrives, satisfied flag is set to false (line 2) until it is placed. Initially other flag status is also set to false (line 3). For placing the incoming vm request at first, remaining capacity and utilization of each started server is

checked. if the remaining capacity of the server is less than the requirement and it can accommodate it then place the vm request, also we check if remaining capacity is less than the minimum threshold and after placing the request new utilization will be remain less than the upper threshold, then vm request is placed on the suitable server and satisfied flag is set to true and utilization and remaining capacity of the server where the request has been placed is updated (lines 5to 9). Otherwise, the other flag is set true and new server is started. The unsatisfied vm request is forwarded to newly started server and satisfied flag is set to true and utilization and remaining capacity is updated (10 to 16). In the algorithm, vm request is placed at first suitable server.

When we start a new server then it takes some time to get started. To tackle this problem, a queue is maintained at each server which contains the set of allocated request to that server. When already started servers are unable to satisfy the vm request, the new server is started and in the meanwhile we keep allocating the vm requests without waiting for the time that server takes to get started. These allocated requests to the starting server are kept in a queue and these types of queues are maintained at vm scheduler. When the server gets started these requests start executing.

4.1. Thermal state Optimization and Green Computing

In cloud computing environment, overloading can cause overheating of a server node. This overheating is also not only results in more consumption of energy than required but also causes a large amount of carbon emission. Consecutively, it increases the total operational costs and also have hazardous affect on the environment. Therefore it is necessary to manage the thermal state of servers. The upper threshold is applied in the algorithm to prevent the occurrence of overloading and thereby avoids the overheating. So, we can say that upper threshold helps in optimizing the thermal state of servers as well as supports the green computing by avoiding the energy consumed and carbon emission caused by overheating. This is also very helpful in reducing the operational costs as the energy consumption is reduced.

5. EXPERIMENTAL SETUP AND RESULTS

In this section, we will explain the experimental setup used for the simulation. Also performance of the algorithm is evaluated with help of simulation results.

5.1. Experimental setup

Let us discuss the experimental setup done for the simulations. We have used two datasets for the simulations. For the first dataset, we consider each vm request's requirement to be equal to the vm instances provided by Amazon EC2 [1]; this will help in knowing about the exact resource usage at different servers. We take specifically, five types of vm instances corresponding to the vm deployment requests; including four instances for general purpose applications and one instance for high-cpu deployment requests.

For the second dataset, vm requests resource requirements follow the normal distribution for comparison purposes.

Request type	Vm Instances	No. of CPUs required
1	small	1
2	medium	2
3	large	4
4	x-large	8
5	High-cpu	20

Table 1. CPU resource metrics from amazon EC2

We evaluate the proposed algorithm using custom simulator written in java. We evaluated following metrics in our experiments:

- Number of servers used
- Remaining capacity of the server
- Utilization percentage of the server

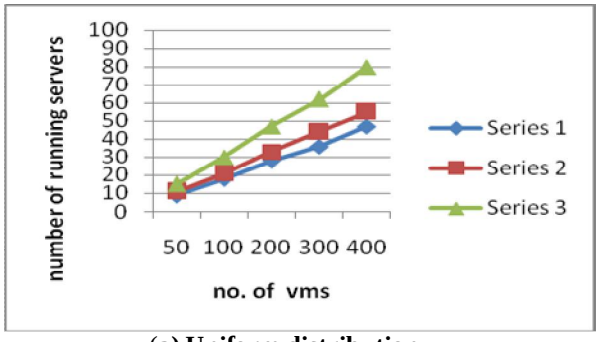
We compared our results with following algorithms:

- Greedy First-fit, it assigns a VM request to the first scanned physical server that satisfies the demands of all resources for that specific request. It is used for comparison purposes in [9,10]
- Min-min, the VM request with the lowest CPU capacity or lowest minimum completion time requirement is assigned first [12].

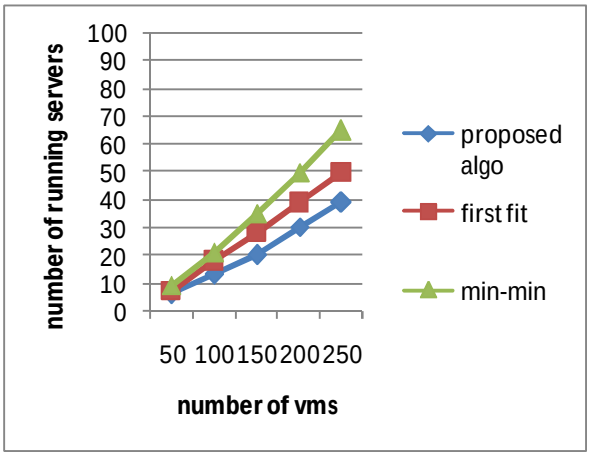
5.2. Simulation results

Here we compare the energy aware resource allocation algorithm with above discussed strategies.

At first we compare the algorithms considering the number of started servers corresponding to the number of vm requests given as input. This comparison is done using two different datasets as shown in figure 3.



(a) Uniform distribution



(b) Random workload generated corresponding to vm instances from Amazon EC2

Figure 3. (a) Uniform work load,(b) Random workload generated corresponding to vm instances from Amazon EC2

Through simulation results shown in the figure3. given below it is clear that the proposed algorithm is able to produce better results when compared to first fit and min-min algorithm in both cases i.e. when uniform and random workload generated corresponding to vm instances from Amazon EC2.

From the results we can say that the number of running servers are being reduced with help of our algorithm while avoiding the overload and overheating. Therefore, energy consumption is also reduced as less number of servers are started by energy aware resource allocation algorithm and thereby total operational cost is also reduced.The proposed algorithm also supports green computing as it avoids overheating of server nodes and consumes less energy and it also decreases the carbon emissions.

6. CONCLUSION AND FUTURE WORK

Cloud computing is an emerging computing paradigm that provides computing power as utility. Also it delivers software, hardware and infrastructure as a service to the users on pay-per-use basis. But cloud computing results in an enormous amount of energy consumption. In today's scenario, how to allocate the resources in an energy efficient way is a major issue. In this paper we have presented an energy efficient resource allocation algorithm to minimize the energy consumption by minimizing the number of servers used. Also the algorithm optimizes the thermal state of the servers by avoiding the overloading.

Our future work focuses on improving the proposed algorithm by designing a technique that can be used when downscaling is required.

REFERENCES

[1] The Amazon Elastic Compute Cloud (Amazon EC2), http://aws.amazon.com/ec2/

[2] IBM Blue Cloud. http://www.ibm.com/ibm/cloud/

[3] Google Compute Engine, https://cloud.google.com/products/compute-engine, 2013.

[4] Beloglazov, Anton, and Rajkumar Buyya. "Energy efficient resource management in virtualized cloud data centers." In Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, pp. 826-831. IEEE Computer Society, 2010

[5] Natural Resources Defense Council "Is Cloud Computing Always Greener? Finding the Most Energy and Carbon Efficient Information Technology Solutions for Small- and Medium-Sized Organizations",http://www.energystar.gov/ia/partners/prod_development/revisions/downloads/computer/RecommendationsTierICompSpecs.pdf.

[6] G. L. Valentini, W. Lassonde, S. U. Khan, N. Min-Allah, S. A. Madani, J. Li, L. Zhang, L. Wang, N. Ghani, J.Kolodziej, H. Li, A. Y. Zomaya, C.-Z. Xu, P. Balaji, A. Vishnu, F. Pinel, J. E. Pecero, D. Kliazovich, and P. Bouvry, "An Overview of Energy Efficiency Techniques in Cluster Computing Systems," Cluster Computing, 2011

[7] Fu, S. and Xu, C. 2004. Migration Decision for Hybrid Mobility in Reconfigurable Distributed Virtual Machines. In Proceedings of the 2004 international Conference on Parallel Processing (August 15 - 18, 2004). ICPP. IEEE Computer Society, Washington, DC, 335-342.

[8] Bays, Carter. "A comparison of next-fit, first-fit, and best-fit." *Communications of the ACM* 20, no. 3 (1977): 191-192.

[9] . Jin, D. Pan, J. Xu, and N. Pissinou, "Efficient VM placement with multiple deterninistic and stochastic resources in data centers," in IEEE Global Communications Conference (GLOBECOM), 2012, pp. 2505–2510.

[10] Li, Bo, Jianxin Li, Jinpeng Huai, Tianyu Wo, Qin Li, and Liang Zhong. "Enacloud: An energy-saving application live placement approach for cloud computing environments." In Cloud Computing, 2009. CLOUD'09. IEEE International Conference on, pp. 17-24. IEEE, 2009.

[11] A Virtual Machine Placement Algorithm for Balanced Resource Utilization in Cloud Data Centers

[12] Kokilavani, T., and Dr DI George Amalarethinam. "Load balanced min-min algorithm for static meta-task scheduling in grid computing." International Journal of Computer Applications 20, no. 2 (2011): 43-49

[13] Xu, Zhong, and Rong Huang. "Performance study of load balancing algorithms in distributed web server systems." CS213 Parallel and Distributed Processing Project Report 1 (2009).

[14] Lin, Ching-Chi, Pangfeng Liu, and Jan-Jan Wu. "Energy-efficient virtual machine provision algorithms for cloud systems." In Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on, pp. 81-88. IEEE, 2011.

[15] Galloway, Jeffrey M., Karl L. Smith, and Susan S. Vrbsky. "Power aware load balancing for cloud computing." In Proceedings of the World Congress on Engineering and Computer Science, vol. 1, pp. 19-21. 2011.

[16] Kim, Kyong Hoon, Anton Beloglazov, and Rajkumar Buyya. "Power-aware provisioning of cloud resources for real-time services." In Proceedings of the 7th International Workshop on Middleware for Grids, Clouds and e-Science, p. 1. ACM, 2009

[17] Usmin, S., M. Arockia Irudayaraja, and U. Muthaiah. "Dynamic placement of virtualized resources for data centers in cloud." In Information Communication and Embedded Systems (ICICES), 2014 International Conference on, pp. 1-7. IEEE, 2014.
[18] He, Li. "A method of virtual machine placement based on gray correlation degree." In Software Engineering and Service Science (ICSESS), 2014 5th IEEE International Conference on, pp. 419-424. IEEE, 2014.
[19] Xiao, Zhen, Weijia Song, and Qi Chen. "Dynamic resource allocation using virtual machines for cloud computing environment." Parallel and Distributed Systems, IEEE Transactions on 24, no. 6 (2013): 1107-1117
[20] Kliazovich, Dzmitry, Sisay T. Arzo, Fabrizio Granelli, Pascal Bouvry, and Samee Ullah Khan. "e-STAB: energy-efficient scheduling for cloud computing applications with traffic load balancing." In Green Computing and Communications (GreenCom), 2013 IEEE and Internet of Things (iThings/CPSCom), IEEE International Conference on and IEEE Cyber, Physical and Social Computing, pp. 7-13. IEEE, 2013
[21] Hieu, Nguyen Trung, Mario Di Francesco, and Antti Yla Jaaski. "A Virtual Machine Placement Algorithm for Balanced Resource Utilization in Cloud Data Centers." In Cloud Computing (CLOUD), 2014 IEEE 7th International Conference on, pp. 474-481. IEEE, 2014.