International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 04 Issue: 05 | May -2017 www.irjet.net p-ISSN: 2395-0072

LOAD BALANCING FOR CLOUD ECOSYSTEM USING ENERGY AWARE APPLICATION SCALING METHODOLOGIES

Asma Anjum[1], Dr. Rekha Patil[2]

[1]*PG Student, Department of CSE, PDA College of Engineering, Kalaburagi, Karnataka, India.*
asmacs13@gmail.com
[2]*Professor, Department of CSE, PDA College of Engineering, Kalaburagi, Karnataka, India.*
rekha.patilcse@gmail.com

Abstract - *Cloud computing endeavors the utility oriented IT services to the users worldwide. Almost Nearly all of the companies are locating their data onto the cloud. Managing the data onto the servers and making it available to users as per their requirement with respect to the SLAs in an energy efficient manner is difficult. We explain an energy aware application scaling and load balancing operation model for cloud eco system. The main concept of our approach is taking into concept an energy-optimal operation regime and attempting to maximize the number of servers that are operating in this regime. Lightly-loaded and Idle servers are switched to one of the sleep states in order to save energy. The servers are being added in order to balance the load and avoid the deadlock or overload condition by deploying the scaling methodologies. Henceforth, we were capable to show how the load is balanced by adding the number of servers and maximizing the count of servers in order to serve the request of client. Allotting evenly the workload to a set of servers minimizes the response time , maximizes the throughput and increases the system resilience to faults preventing overloading the systems.*

Key Words: **Energy Aware Load Balance, Server, Creating Load, System Model, Cloud Computing.**

1. INTRODUCTION

The concept of "load balancing" means exactly what the name denotes, to evenly distribute the workload to a set of servers to minimize the response time, maximize the throughput, and increase the system flexibility to errors by avoiding overloading the systems. An important method for reduction in energy is concentrating the load on a small subset of servers and switching the remaining of them to a state possessing low energy consumption whenever it is possible. This observation signifies that the traditional approach of load balancing in a large-scale system could be reevaluated as follows: distribute uniformly the workload to the smallest set of servers that are performing at optimal or near-optimal energy levels, while observing the Service Level Agreement (SLA) between the cloud user and CSP [1]. Idle and underutilized servers contribute to significantly more wastage of energy [2]. The energy efficiency is carried out by the ratio "Performance per watt of power". Since the previous two decades the performance of computing systems has increased more faster than their energy efficiency [3]. An ideal energy level is one when the performance per Watt of power is maximized in response to a request consistent with the SLA [2]. Scaling is the technique of allocating additional resources to a cloud application.

We distinguish two scaling modes, vertical and horizontal scaling. Horizontal scaling is the most common method of scaling on a cloud; it is done by increasing the number of Virtual Machines (VMs) when the load of applications increases and minimizing this number when the load decreases. Vertical scaling keeps the number of VMs of an application constant, but maximizes the amount of resources that is allocated to each one of them.

Scaling is the ability of the system, process or network to handle the situation to handle the growing network or its capability to be enlarged to accommodate that growth. A scalable system is one whose performance improves after adding hardware.

The two ways of performing the scaling are horizontal scaling and vertical scaling.

Horizontal scalability is the ability of increasing capacity by connecting multiple hardware or software entities so that they coordinate as a single logical unit. When the servers are clustered, the main original server is being scaled out horizontally. If a cluster requires more number of resources to enhance the performance and provide high availability (HA), an administrator can scale out by adding more number of servers to the cluster.

Vertical scalability besides, increases the capacity by adding more number of resources, such as more additional CPU or memory, to a machine. Scaling vertically that is scaling up, usually needs downtime beside new resources are being added and has limitation which are defined by hardware.

Organization of paper: 1.Introduction, 2.Related work, 3.Proposed Work, 4. Implementation and Results, 5.Conclusion and Future Work.

2. RELATED WORK

The vast expansion of the cloud computing has a remarkable impact on the energy consumption on US and world[3], which motivated us to work on the energy optimization methods for cloud computing. An ideal energy-proportional system is always operating at 100% efficiency [4]. With various degree of energy efficiency we are introducing a new model of cloud servers which are based on different operating regimes. Load balancing is carried out in order to provide the efficient and optimized utilization of resources and overall cost minimization [5]. So that the servers need not wait for the server to available, we introduce a novel algorithm that performs the load balancing and application scaling. The scaling is done in two ways, horizontal scaling and vertical scaling . Amongst the requirements of Cloud Computing one of the important requirements for a Cloud computing environment is providing reliable QoS. It can be defined in terms of Service Level Agreements (SLA) that describe such characteristics as minimal throughput, maximal response time or latency delivered by the deployed system[6].Therefore the load balancing must deploy the scaling mechanism with respect to the SLA. In [7] the authors investigated the regime of sleep states that would be advantageous in data centers. This consider the benefits of sleep states across three orthogonal dimensions: (i) the variability in the workload trace,(ii)the type of dynamic power management policy employed, and (iii) the size of the data center. In [8] the authors combined the reactive controller with a workload placement controller that observes current behavior to i) migrate workloads off of overloaded servers and ii) free and shut down lightly-loaded servers[11]. The main purpose for data centers in cloud computing is to enhance the profit and lower down the power consumption and maintain SLAs. In [9], author describes a framework for resource management that combines a dynamic virtual machine placement manager and dynamic VM provisioning manager. It can take many more experiments that explain how system can be controlled to make trade-offs between application performance and energy consumption. The energy consumption of the system have workload scalability problem. The lightly loaded server also needs the more amount of energy so in[10]author proposed the concept load balancing to optimize the energy consumption for large scale system that can allot the workload among different set of servers that can observe the response time and operate on optimal energy level.

3. PROPOSED WORK

There are three main contributions of this paper: (1) with various degrees of "energy efficiency", a new model of cloud servers which is based on different operating regimes; (2) a novel algorithm that performs application scaling and load balancing to increase the number of servers which are operating in the energy-optimal regime; and (3) comparison and analysis of techniques for application scaling load balancing and using two different average load profiles and three differently-sized clusters.

The proposed work includes two modules. The admin module and the user module where in the admin will be able to add the servers, check the server's configuration and monitor the servers. The admin module involves server consolidation and load balance. The user will be able to request server and make the use of cloud creating the load for the servers .The user module involves the system model and creating load.

The proposed model includes the following entities:
System Model, Server Consolidation, Creating Load, and Load Balance.

3.1. System Model

In this module, we design the system, such that client makes request to server. Usually, it is designed with adequate resources in order to satisfy the traffic volume generated by end-users. Generally, a judicious furnishing of resources can make sure that the input rate is always lower than the service rate. In such a case, the system will be capable to efficiently serve all users' requests. Applications for one instance family have similar profiles, e.g., are CPU, memory, or I/O-intensive and run on clusters optimized for that profile; thus, the application interference with one another is minimized. The normalized power consumption and the normalized system performance varies from server to server; yet, warehouse scale computers supporting an instance family use the same processor or family of processors [13] and this reduces the effort to determine the parameters required by our model. In our model the migration decisions are based solely on the CPU units demanded by an application and the available capacity of the host and of the other servers in the cluster. The model could be extended to take into account not only the processing power, but also the dominant resource for a particular instance family, e.g., memory for R3, storage for I2, GPU for G2 when deciding to migrate a VM. This extension would complicate the model and add additional overhead for monitoring the application behavior.

3.2. Server Consolidation

The term server consolidation is used to explain: (1) switching lightly loaded and idle systems to a sleep state; (2) workload migration to prevent overloading of systems [11]; or (3) any optimization of cloud performance and energy efficiency by redistributing the workload [12]. In this module we design the Server System, where the server processes the client request. Cloud is a large distributed system of servers deployed in multiple data centers across the Internet. The goal of a cloud is to serve content to end-users with high availability and high performance. Cloud provides a large amount of the Internet data today, involving web objects

(graphics, text and scripts), applications (e-commerce, portals), live streaming media, downloadable objects (media files, software, and documents), on-demand streaming media, and social networks. Apart from better availability and performance, cloud also offload the traffic served directly from the content provider's origin infrastructure, resulting in cost savings for the content provider.

3.3. Creating Load

In this module, we create the load to the server. Although, in this research we exclusively focus on critical conditions where the global resources of the network are close to saturation. This is a realistic assumption since an unusual traffic condition characterized by a high volume of requests, i.e., a flash crowd, can always overfills the available system capacity. In such kind of a condition, servers are not all overloaded. Rather, we typically have local instability conditions where the input rate is greater than the service rate. In this example, by redistributing the excess load to less loaded servers , the balancing algorithm helps us to avoid a local instability condition.

3.4. Load Balance

The aim of the algorithms is to make sure that the largest possible numbers of active servers operate within the boundaries of their respective optimal operating regime. The actions that are implementing this policy are as follows: (a) migrate the VMs from an overloaded server, a server that is operating in the undesirable-high regime with applications predicted to increase their demands for computing in the next reallocation cycles; (b) switch an idle server to a sleep state and when the cluster load increases then reactivate servers in a sleep state; (c) migrate VMs from a server operating in the undesirable-low regime and then switch the server to a sleep state. We give a new structure for redirecting incoming client requests to the most appropriate server, and therefore balancing the overall system requests load. Our mechanism advantages local balancing in order to achieve global balancing. This is carried out through a periodic interaction amongst the system nodes. Depending upon mechanisms and the network layers involved in the process, generally request routing techniques can be classified in transport-layer request routing, cloud request routing, and application-layer request routing.

4. IMPLEMENTATION AND RESULTS

The simulation experiments were implemented using Visual Studio 2012 connecting it to the SSMS R2(SQL Server Management Studio) for the database management. Using C# .NET as back end and HTML, CSS and bootstrap as front end technologies.
The simulation experiments reported in this paper were carried out by taking an online shopping website maintaining the database by adding the servers and hence we were able to demonstrate the application scaling mechanism i.e., horizontal scaling and vertical scaling for balancing the load onto the cloud. Thus we were able to demonstrate how the load balancing is done by adding the servers and maximizing the number of servers in order to serve the client's request. Distributing evenly the workload to a set of servers maximizing the throughput, minimizing the response time and increasing the system resilience to faults by avoiding overloading the systems.

5. CONCLUSION AND FUTURE WORK

Through simulation we were able to explain load balancing for the cloud eco system by deploying the scaling methodologies. Therefore we were able to serve the client's request by maximizing the number of servers operating in this regime.

Our future work will involve the implementation of server application manager which can be integrated in self-management policies [14].

REFERENCES

[1] M. Blackburn and A. Hawkins. "Unused server survey results analysis." www.thegreengrid.org/media/White Papers/Unused%20Server%20StudyWP101910v1. ashx?lang=en (Accessed on December 6, 2013).

[2] D. Ardagna, B. Panicucci, M. Trubian, and L. Zhang. "Energy-aware autonomic resource allocation in multitier virtualized environments." IEEE Trans. on Services Computing, 5(1):2–19, 2012.

[3] S. V. Vrbsky, M. Lei, K. Smith, and J. Byrd. "Data replication and power consumption in data grids." Proc IEEE 2nd Int. Conf. on Cloud Computing Technology and Science, pp. 288–295, 2010.

[4] L. A. Barroso and U. H¨ozle. "The case for energy proportional computing." IEEE Computer, 40(12):33– 37, 2007.

[5] A.S. Thorat , Prof. S.K.Sonkar." A review on energy efficient load balancing techniques for secure and reliable cloud eco system" IJARIIE-ISSN(O)-2395-4396 Vol-2 Issue-1 2016]

[6] A. Beloglazov, R. Buyya "Energy efficient resource management in virtualized cloud data centers." Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Comp., 2010.

[7] A. Gandhi, M. Harchol-Balter, R. Raghunathan, and M.Kozuch. "Are sleep states effective in data centers?" Proc. Int. Conf. on Green Comp., pp. 1–10, 2012.

[8] D. Gmach, J. Rolia, L. Cherkasova, G. Belrose, T. Tucricchi, and A. Kemper. "An integrated approach to resource pool

management: policies, efficiency, and quality metrics." Proc. Int. Conf. on Dependable Systems and Networks, pp. 326–335, 2008.

[9] H. N. Van, F. D. Tran, and J.-M. Menaud. "Performance and power management for cloud infrastructures." Proc. IEEE 3rd Int. Conf. on Cloud Comp., pp. 329{336, 2010.

[10] A. Paya and D. C. Marinescu. "Energy-aware load balancing policies for the cloud ecosystem." http: //arxiv.org/pdf/1307.3306v1.pdf, December 2013.

[11]A. Beloglazov and R. Buyya. "Managing overloaded hosts for dynamic consolidation on virtual machines in cloud centers under quality of service constraints." IEEE Trans. on Parallel and Distributed Systems, 24(7):13661379, 2013.

[12] C. Mastroianni, M. Meo, G. Papuzzo. " Probabilistic consolidation of virtual machines in self-organizing cloud data centers." IEEE Trans. on Cloud Computing, 1(2):215–228, 2013.

[13] L. A. Barossso, J. Clidaras, and U.H¨ozle. The Datacenter as a Computer; an Introduction to the Design of Warehouse-Scale Machines. (Second Edition). Morgan & Claypool, 2013.

[14]D. C. Marinescu, A Paya, and J.P. Morrison. "Coalition formation and combinatorial auctions; applications to self-organization and self-management in utility computing." http: //arxiv.org/pdf/1406.7487v1.pdf, 2014.