Perseverance Force Load Stabilize and Purpose Escalate for the Cloud Ecosystem

Priyadharshini P[1], Kannabiran G[2]

[1]Priyadharshini P, M.E (CSE)/ Global Institute of Engineering and Technology, Vellore India.

[1]priyadharshini631@gmail.com

[2]Kannabiran G, Asst Prof (CSE)/ Global Institute of Engineering and Technology, Vellore India.

[2]vishnukannan10@gmail.com

ABSTRACT

Energy aware operation model used for load balancing and application scaling on a cloud. The basic philosophy of our approach is defining an energy optimal operation regime and attempting to maximize the number of servers operating in this regime. Idle and lightly –loaded servers are switched to one of the sleep states to save energy. The load balancing and scaling algorithms also exploit some of the most desirable features of server consolidation mechanisms .In the last few years packaging computing cycles and storage and offering them as a metered service became a reality. Large farms of computing and storage platforms have been assembled and a fair number of cloud service providers(CSPs) offer computing services based on three cloud delivery models SAAS (software as a service)PAAS(platform as a service)an IASS(infrastructure as a service). The number of CSPs, the spectrum of services offered by the CSPs, and the number of cloud users have increased drastically during the last few years. For example, in 2007 the EC2 (Elastic Computing Cloud) was the first service provided by AWS (Amazon Web Services); five year later, in 2012, AWS was used by businesses in 200 countries. Amazon's S3 (Simple Storage Services) has surpassed two trillion objects and routinely runs more than 1.1 million peak requests per second.

Keywords—load balancing, cloud service providers.

1. INTRODUCTION

Warehouse-Scale Computers are the building blocks of a cloud infrastructure. A hierarchy of networks connect 50; 000 to 100; 000 servers in a WSC. The servers are housed in racks; typically, the 48 servers in a rack are connected by a 48-port Gigabit Ethernet switch. The switch has two to eight up-links which go to the higher level switches in the network hierarchy. This observation implies that the traditional concept of load balancing a large-scale system could be reformulated as follows: distribute evenly the workload to the smallest set of servers operating at optimal or near-optimal energy levels, while observing the Service Level Agreement (SLA) between the CSP and a cloud user. An optimal energy level is one when the performance per Watt of power is maximized. The ability to use as many resources as needed at any given time, and low cost, a user is charged only for the resources it consumes, represents solid incentives for many organizations to migrate their computational activities to a public cloud. The number of CSPs, the spectrum of services by the CSPs, and the number of cloud users have increased dramatically during the last few years .The concept of "load balancing" dates back to the time when the first distributed computing systems were implemented. It means exactly what the name implies, to evenly distribute the workload to a set of servers to maximize the throughput, minimize the response time, and increase the system resilience to faults by avoiding overloading the system. An important strategy for energy reduction is concentrating the load on a subset of servers and, whenever possible, switching the rest of them to a state with a low energy consumption

The number of VMs of an application constant, but increases the amount of resources allocated to each one of them. This can be done either by migrating the VMs to more powerful servers or by keeping the VMs on the same servers, but increasing their share of the server capacity. The first alternative involves additional overhead; the VM is stopped a snapshot is taken, the file is migrated to a more powerful server, and the VM is restarted at the new site. Scaling is the process of allocating resources to a cloud application in response to a request consistent with the SLA. We distinguish two scaling modes, horizontal and vertical scaling. Horizontal scaling is the most common mode of scaling on a cloud; It is done by increasing the number of virtual machines(VMs) when the load of application increases and reducing this number when the load decreases.

2. OVERVIEW OF EXISTING SYSTEM

We also assume a clustered organization, typical for existing cloud infrastructure .When the existing applications scale up above of the capacity with all servers running then the cluster leader interacts with the leaders of other clusters to satisfy the requests. This case is not addressed in this project. In order to integrate business requirements and application level needs, in terms of Quality of Service (QoS), cloud service provisioning is regulated by Service Level Agreements (SLAs): contracts between clients and providers that express the price for a service, the QoS levels required during the service provisioning, and the penalties associated with the SLA violations. In such a context, performance

evaluation plays a key role allowing system managers to evaluate the effects of different resource management strategies on the data center functioning and to predict the corresponding costs/benefits.

1) On-the-field experiments are mainly focused on the offered QoS, they are based on a black box approach that makes difficult to correlate obtained data to the internal resource management strategies implemented by the system provider.

2) Simulation does not allow to conduct comprehensive analyses of the system performance due to the great number of parameters that have to be investigated.

3. PROPOSED APPROACH

We introduce an energy-aware operation model used for load balancing and application scaling on a cloud. The basic philosophy of our approach is defining an energy-optimal operation regime and attempting to maximize the number of servers operating in this regime. Idle and lightly-loaded servers are switched to one of the sleep states to save energy.

The objective of the algorithms is to ensure that the possible number of active servers operate within the boundaries of their respective optimal operating regime. The actions implementing this policy are: (a) migrate VMs from a server operating in the undesirable-low regime and then switch the server to a sleep state; (b) switch an idle server to a sleep state and reactivate servers in a sleep state when the cluster load increases; (c) migrate the VMs from an overloaded server, a server operating in the undesirable-high regime with applications predicted to increase their demands for computing in the next reallocation cycles.

1)After load balancing, the number of servers in the optimal regime increases from 0 to about 60% and a fair number of servers are switched to the sleep state.

2) There is a balance between computational efficiency and SLA violations; the algorithm can be tuned to maximize computational efficiency or to minimize SLA violations according to the type of workload and the system management policies.

4. SYSTEM DESIGN

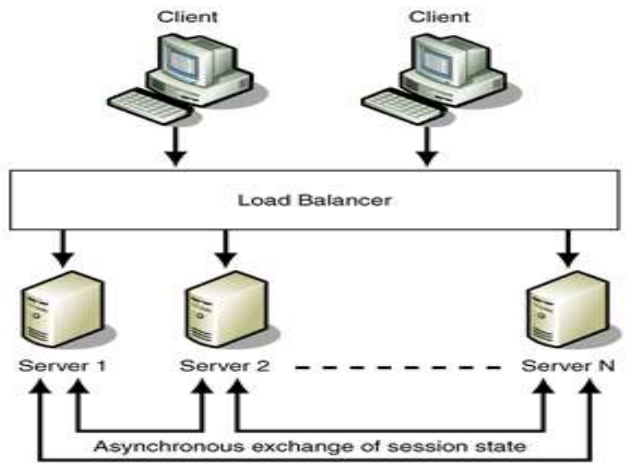

Fig-1: System Architecture

4.1. Input Design

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy.

1.Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

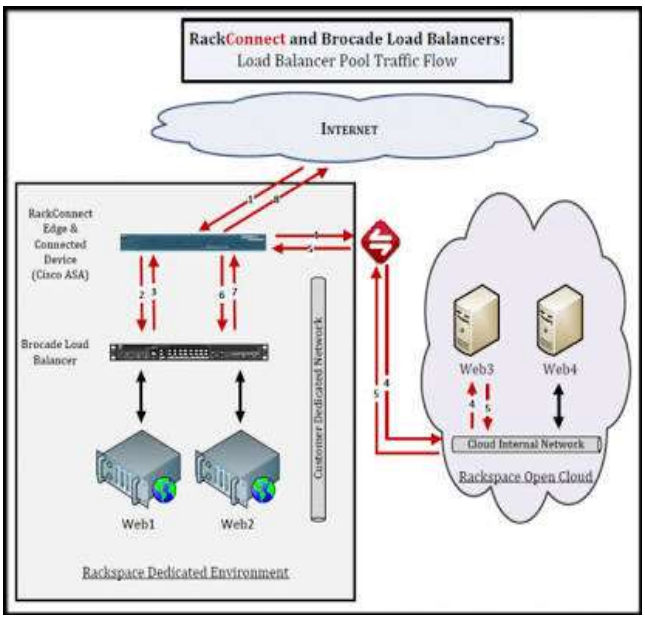

Fig-2: System Process in Cloud

4.2.Output Design

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient

and intelligent output design improves the system's relationship to help user decision-making.

1.Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

5. FEASABILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

5.1. Economical Feasability

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

5.2. Technical Feasability

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

5.3. Social Feasability

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

6.MODULE DESCRIPTION

6.1. Load Balancing in Cloud Computing

"Cloud computing" is a term, which involves virtualization distributed computing, networking, software and web services. A cloud consists of several elements such as clients, data center and distributed servers. It includes fault tolerance, high availability, scalability, flexibility, reduced overhead for users, reduced cost of ownership, on demand services etc. Central to these issues lies the establishment of an effective load balancing algorithm. The load can be CPU load, memory capacity, delay or network load. Load balancing is the process of distributing the load among various nodes of a distributed system to improve both resources utilization and job response time while also avoiding a situation where some of the nodes are heavily loaded while mother nodes are idle or doing very little work. Load balancing ensures that all the processor in the system or every node in the network does approximately the equal amount of work at any instant of time. This technique can be sender initiated, receiver initiated or symmetric type (combination of sender initiated and receiver initiated types). Our objective is to develop an effective load balancing algorithm using Divisible load scheduling or to maximize or minimize different performance parameters (throughput ,latency for Depending on the application requirement).

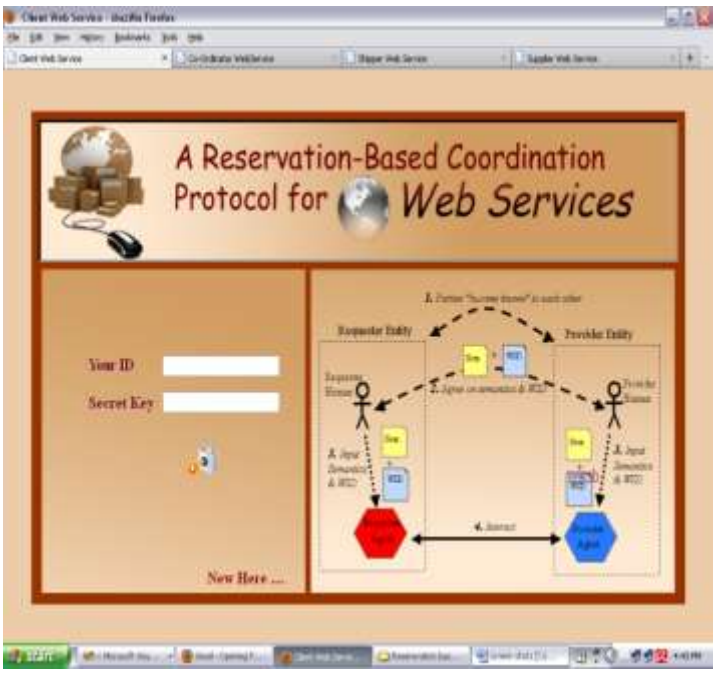

Fig- 3: Login Process

6.2. Idle Servers

Idle and under-utilized servers contribute significantly to wasted energy, see Section survey reports that idle servers contribute 11 million tons of unnecessary CO2 emissions each year and that the total yearly costs for idle servers is billion. An energy-proportional system consumes no energy when idle, very little energy under a light load, and gradually, more energy as the load increases.

6.3. Server Consolidation

Any optimization of cloud performance and energy efficiency by redistributing the workload .Server consolidation policies have been proposed to decide when to switch a server to a sleep state. The reactive policy responds to the current load; it switches the servers to a sleep state when the load decreases and switches them to the running state when the load increases. Generally, this policy leads to SLA violations and could work only for slow varying predictable loads.

To reduce SLA violations one can envision a reactive with extra capacity policy when one attempts to have a safety margin and keep running a fraction of the total number of servers, e.g., 20% above those needed for the current load. The auto scale policy is a very conservative reactive policy in switching servers to sleep state to avoid the power consumption and the delay in switching them back to running state. This can be advantageous for un predictable policies. The moving window policy estimates the workload by measuring the average request rate in a window of size seconds uses this average to predict the load for second(_+2). And so on. The predictive linear regression policy uses a linear regression to predict the future load.

6.4.Energy Proportional Systems

In an ideal world, the energy consumed by an idle system should be near zero and grow linearly with the system load. In real life, even systems whose energy requirements scale linearly, when idle, use more than half the energy they use at full load. Data collected over a long period of time shows that the typical operating regime for data centers. servers is far from an optimal energy consumption regime. An energy-proportional system consumes no energy when idle, very little energy under a light load, and gradually, more energy as the load increases. An ideal energy-proportional system is always operating at 100% efficiency. The energy efficiency of a system is captured by the ratio performance per Watt of power." During the last two decades the performance of computing systems has increased much faster than their energy efficiency

Energy proportional systems, In an ideal world, the energy consumed by an idle system should be near zero and grow linearly with the system load. In real life, even systems whose energy requirements scale linearly, when idle, use more than half the energy they use at full load. Data collected over a long period of time shows that the typical operating regime for data center servers is far from an optimal energy consumption regime.

The dynamic range is the difference between the upper and the lower limits of the energy consumption of a system as a function of the load placed on the system. A large dynamic range means that a system is able to operate at a lower fraction of its peak energy when its load is low

7.CONCLUSION

The realization that power consumption of cloud computing center is signicant and is expected to increase substantially in the future motivates the interest of the research community in energy-aware resource management and application placement policies and the mechanisms to enforce these policies. Low average server utilization and its impact on the environment make it imperative to devise new energy-aware policies which identify optimal regimes for the cloud servers and, at the same time, prevent SLA violations

A quantitative evaluation of an optimization algorithm or an architectural enhancement is a rather intricate and time consuming process; several benchmarks and system configurations are used to gather the data necessary to guide future developments. For example, to evaluate the effects of architectural enhancements supporting Instruction-level or Data-level Parallelism on the processor performance and their power consumption several benchmarks are used the results show different numerical outcomes for the individual applications in each benchmark. Similarly, the effects of an energy-aware algorithm depend on the system configuration and on the application and cannot be expressed by a single numerical value

REFERENCES

[1]User Interfaces in C#: Windows Forms and Custom Controls by Matthew MacDonald.
[2]Applied Microsoft® .NET Framework Programming (Pro-Developer) by Jeffrey Richter.
[3]Practical .Net2 and C#2: Harness the Platform, the Language, and the Framework by Patrick Smacchia.
[4]Data Communications and Networking, by Behrouz A Forouzan.
[5]Computer Networking: A Top-Down Approach, by James F. Kurose.
[6]Operating System Concepts, by Abraham Silberschatz.
[7]M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” University of California, Berkeley, Tech. Rep. USB-EECS-2009-28, Feb 2009.
[8]“The apache
Cassandra project,” http://cassandra.apache.org/.
[9]L. Lamport, “The part-time parliament,” ACM Transactions on Computer Systems, vol. 16, pp. 133–169, 1998.
[10]N. Bonvin, T. G. Papaioannou, and K. Aberer, “Cost-efficient
and differentiated data availability guarantees in data clouds,” in Proc. of the ICDE, Long Beach, CA, USA, 2010.
[11]O. Regev and N. Nisan, “The popcorn market. online markets
for computational resources,” Decision Support Systems, vol. 28, no. 1-2, pp. 177 – 189, 2000.
[12]A. Helsinger and T. Wright, “Cougaar: A robust configurable
multi agent platform,” in Proc. of the IEEE Aerospace Conference,
2005.

[13]J. Brunelle, P. Hurst, J. Huth, L. Kang, C. Ng, D. C. Parkes,
M. Seltzer, J. Shank, and S. Youssef, "Egg: an extensible and economics-inspired open grid computing platform," in Proc. of the GECON, Singapore, May 2006.
[14]J. Norris, K. Coleman, A. Fox, and G. Candea, "Oncall: Defeating
spikes with a free-market application cluster," in Proc. of the International Conference on Autonomic Computing, New York, NY, USA, May 2004.
[15]C. Pautasso, T. Heinis, and G. Alonso, "Autonomic resource
provisioning for software business processes," Information and Software Technology, vol. 49, pp. 65–80, 2007.
[16] A. Dan, D. Davis, R. Kearney, A. Keller, R. King, D. Kuebler,
H. Ludwig, M. Polan, M. Spreitzer, and A. Youssef, "Web services on demand: Wsla-driven automated management," IBM Syst. J., vol. 43, no. 1, pp. 136–158, 2004.
[17] M. Wang and T. Suda, "The bio-networking architecture:a biologically inspired approach to the design of scalable, adaptive, and survivable/available network applications," in Proc.
of the IEEE Symposium on Applications and the Internet, 2001.
[18] N. Laranjeiro and M. Vieira, "Towards fault tolerance in web services compositions," in Proc. of the workshop on engineering fault tolerant systems, New York, NY, USA, 2007.
[19] C. Engelmann, S. L. Scott, C. Leangsuksun, and X. He, "Transparent symmetric active/active replication for servicelevelhigh availability," in Proc. of the CCGrid, 2007.
[20]J. Salas, F. Perez-Sorrosal, n.-M. M. Pati and R. Jim´enez-Peris, "Ws-replication: a framework for highly available web services," in Proc. of the WWW, New York, NY, USA, 2006,

