Exploration of Energy consumption in Cloud computing

Gopinath.S[1],Jegan.R.R[2]
PG Scholar[1], Assistant Professor[2],Department of Electronics and Communication Engineering,
PGP College of Engineering & Technology,
Anna University, Chennai-25, INDIA
E-mail: gopivims@gmail.com[1]

***Abstract*— Cloud Computing is a new paradigm that, just as telephone was first invented at home and evolved to be served from a new service providers, aims to transform computing into an utility. It enables hosting of applications from consumer, scientific and business domains and it is being forecasted that more and more users will rent computing as a service, moving the processing power and storage to centralized infrastructures rather than located in client hardware. This is already enabling startups and other companies to start web services without having to invest upfront in dedicated infrastructure. With energy shortages and global climate change leading our concerns these days, the power consumption of data centers has become a key issue. Unfortunately, this new paradigm also has its own drawbacks such as power consumptions. However existing work controls the power and application level performance in a combined architecture, there is still the consumption of the power by a cloud server could be controlled further.. This paper proposes for minimizing the power consumption by implementing the Scheduling algorithm to the cloud infrastructure. The implementation of such Scheduling algorithms should not violate the SLA's such as throughput and response time of the cloud server.**

Keywords- Virtualization, Power consumption, Scheduling, SLA's.

I. INTRODUCTION

Cloud Computing: An overview

Cloud computing uses third party service (Web service) to perform computing needs. Here C**loud** depicts **Internet.** With cloud computing, users can scale up to massive capacities in an instant without having to invest new infrastructure. Cloud computing is benefit to small and medium-sized businesses. Basically consumers use what they need on the Internet and pay only for what they use.

Cloud computing incorporates Infrastructure as a service (IaaS), Platform as a service (PaaS), and Software as a service (SaaS) as well as Web 2.0.

Cloud computing eliminates the cost and complexity of buying, configuring and managing the hardware and software needed to build and deploy applications, these applications are delivered as a service over the Internet (the Cloud).

Examples: Amazon Web services,
Google apps, etc…

Figure.1 shows that the infrastructure of cloud computing environment,

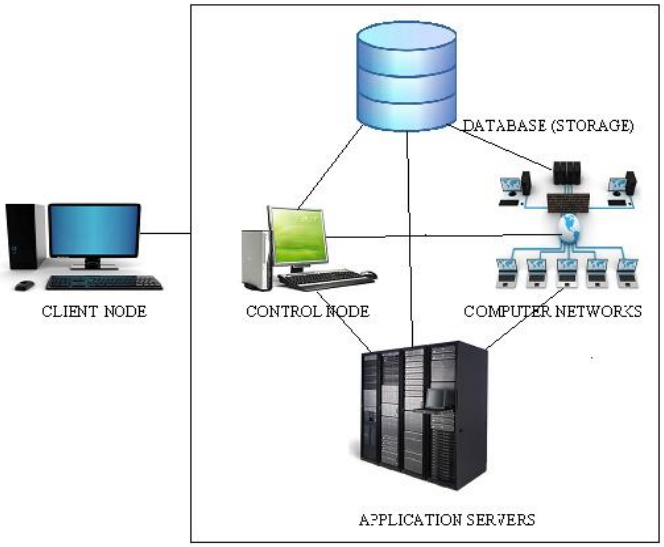

Figure.1: Infrastructure of cloud computing environment

Characteristics of cloud computing:

- Cloud computing is Scalable, scalability is accomplished through load balancing of application instances running separately on a variety of operating systems and connected through Web services. CPU and network bandwidth is allocated and de-allocated on demand. The system's storage capacity goes up and down depending on the number of users, instances, and the amount of data transferred at a given time.
- Involves multitenancy and multitasking; meaning that many customers can perform different tasks, accessing a single or multiple application instances. Sharing resources among a large pool of users assists in reducing infrastructure costs and peak load capacity. Cloud and grid computing provide service- level agreements (SLAs) for guaranteed uptime availability of, say, 99 percent. If the service slides below the level of the guaranteed uptime service, the consumer will get service credit for receiving data late.
- Cloud servers provide a web services interface for the storage and retrieval of data in the cloud. We can store

an object as small as 1 byte and as large as 5 GB or even several terabytes. The data is stored securely using the same data storage infrastructure that same as commercial Web sites.

II. SYSTEM MODELLING

CO-CON Architecture:

In this section, the authors gave a high-level description of the Co-Con coordinated control architecture [1]. An important feature of Co-Con is that it relies on feedback control theory as a theoretical foundation. In recent years, control theory has been identified as an effective tool for power and performance control due to its analytical assurance of control accuracy and system stability. Control theory also provides well-established controller design approaches, e.g., standard ways to choose the right control parameters, such that exhaustive iterations of tuning and testing can be avoided. Furthermore, control theory can be applied to quantitatively analyze the control performance (e.g., stability, settling time) even when the system model changes significantly due to various system uncertainties such as workload variations. This rigorous design methodology is in sharp contrast to heuristic-based adaptive solutions that heavily rely on extensive manual tuning. .Co-Con is a two-layer control solution, which includes a cluster-level power control loop and a performance control loop for each virtual machine.

A. The cluster-level power control:

The cluster-level power controller dynamically controls the total power consumption of all the servers in the cluster by adjusting the CPU frequency of each server with Dynamic Voltage and Frequency Scaling (DVFS)[8].

The cluster-level power control loop is invoked periodically as follows: 1) The cluster-level power monitor (e.g., a power meter) measures the total power consumption of all the servers in the last control period and sends the value to the power controller. The total power consumption is the controlled variable of the control loop. 2) Based on the difference between the measured power consumption and the desired power set point, the power controller computes the new CPU frequency level for the processors of each server, and then sends the level to the CPU frequency modulator on each server. The CPU frequency levels are the manipulated variables of the control loop. 3) The CPU frequency modulator on each server changes the DVFS level of the processors accordingly. The power controller provides an interface to assign weights to different servers. For example, the CPU allocation ratio of each server (i.e., percentage of CPU resource allocated to all the virtual machines on the server) indicates the CPU utilization of the server in the last control period, and can be provided to the controller as weight to give more power to a server whose ratio is higher than the average.

B. Performance Control:

In the second layer, for every virtual machine on each server, we have a performance controller that dynamically controls the application performance of the virtual machine by adjusting the CPU resource (i.e., fraction of CPU time) allocated to it. In this paper, as an example SLA metric, we control the response time of the web server installed in each virtual machine, but our control architecture can be extended to control other SLAs. In addition, we control the average response time to reduce the impact of the long delay of any single web request.

However, our control architecture can also be applied to control the worst-case or 90-percentile response time. We assume that the response time of a web server is independent from that of another web server, which is usually true because they may belong to different customers. Hence, we choose to have a performance control loop for each virtual machine. Our control solution can be extended to handle multitier web services by modeling the correlations between different tiers, which is part of our future work. A cluster-level resource coordinator is designed to utilize the live migration [9] function to move a virtual machine from a server with too much workload to another server for improved performance guarantees.

The performance (i.e., response time) control loop on each server is also invoked periodically. The following steps are executed at the end of every control period:

- The performance monitor of each virtual machine measures the average response time of all the web requests (i.e., controllable variable) in the last control period, and then sends the value to the corresponding performance controller.
- The controller of each virtual machine computes the desired amount of CPU resource (i.e., manipulated variable) and sends the value to the CPU resource allocator. Steps 1 and 2 repeat for all the virtual machines on the server.
- The CPU allocator calculates the total CPU resource requested by the performance controllers of all the virtual machines. If the server can provide the total requested resource, all the requests are granted in their exact amounts. Unallocated resource will not be used by any virtual machines in this control period and can be used to accept virtual machine migration. If the requested resource is more than the available resource, one or more selected virtual machines (running low-priority web services) will be given less resource than requested. If this situation continues for a while, a migration request is sent to the cluster- level CPU resource coordinator to move the selected virtual machines to other servers.
- The cluster-level coordinator tries to find other servers with enough resource and migrates the virtual machines.

C .Co-Ordination of control loops:

Clearly, without effective coordination, the two control loops (i.e., power and performance) may conflict with each other. The CPU frequency manipulated by the power controller will have a direct impact on the application performance of all the virtual machines on the server. The CPU resource allocated by the performance control loops may influence the system power consumption as well. To achieve the desired control functions and system stability, one control loop, i.e., the primary loop needs to be configured with a control period that is longer than the settling time of the other control loop, i.e., the secondary loop. As a result, the secondary loop can always enter its steady state within one control period of the primary control loop. The two control loops are thus decoupled and can be designed independently. The impact of the primary loop on the secondary loop can be modeled as variations in its system model, while the impact of the secondary loop on the primary loop can be treated as system noise.

As long as the two control loops are stable individually, the whole system is stable. In our design, we choose the power control loop as the primary loop for three reasons. First, model variations may cause the secondary loop to severely violate its set point, which is less desirable for the power loop because power limit violations may lead to the shutdown of an entire cluster. Second, the impact of CPU frequency on application performance is usually more significant than the impact of CPU resource allocation on power consumption, and thus, is more appropriate to be modeled as model variations than system noise. Finally, the secondary control loop needs to be designed based on the primary loop. In Co-Con, the control period of the response time control loop is determined based on the estimated execution time of typical web requests such that multiple requests can be processed in a control period.

III. PROPOSED WORK

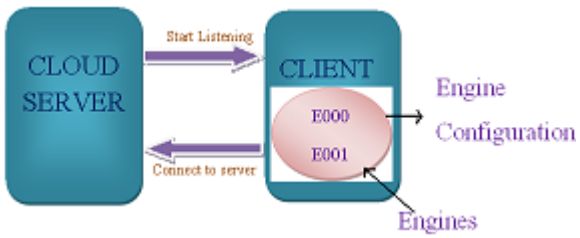

Figure.2 User communicating with the cloud server

Figure.2 shows that schematic of establishing the communication between the cloud server and the client user. The cloud server may have number of client users; each client node has number of virtual machines (VM's). The client nodes can be able to create the VM's depends upon the availability of its own physical resources. In this case we just considered a single client node connected to the cloud server, and the client node has two VM's in it to perform the computing needs.

A. ARCHITECTURE

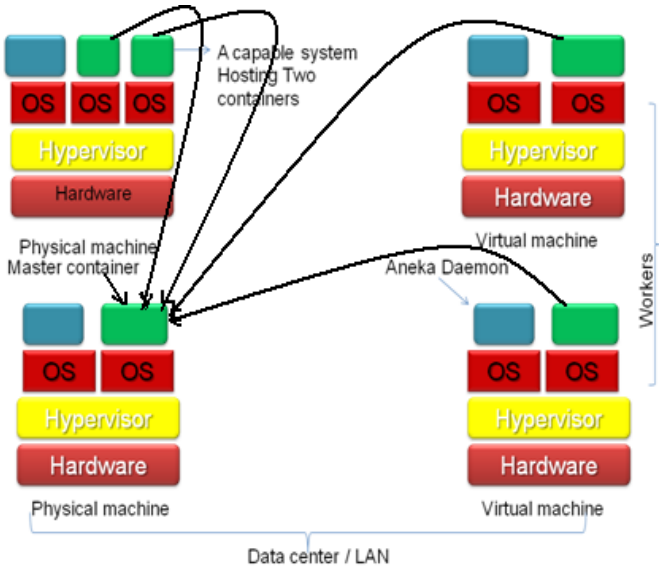

Fig.3 Architecture with the combination of physical machines and virtual machines in a Cloud network

Figure 3 shows the architecture with the combination of physical machines and virtual machines in a cloud network which establishes the connection between the clients to the Cloud server. The cloud consists of number of physical machines and virtual machines. All the nodes in the network are connected to the Server. The master container typically runs on a powerful machine and is responsible for Scheduling jobs to worker nodes. All worker nodes should register with the master. The Aneka Daemon is responsible for installing, starting, stopping and removing containers.

IV. EXPERIMENTS AND RESULTS

Figue.2 shows the performance analysis of the Single cloud server when it is connected to a single client node.

The execution time of the system is calculated by,

$$T = \frac{N * CPI}{f * 10^6} s$$

$$CPI = \frac{f * 10^6}{IPS}$$

$$MIPS = \frac{f}{CPI}$$

Where, T= Total program execution time
N= No.of instructions executed
CPI= Cycles per instruction
f= frequency
IPS= Instructions executed per second
MIPS= Millions instructions per second

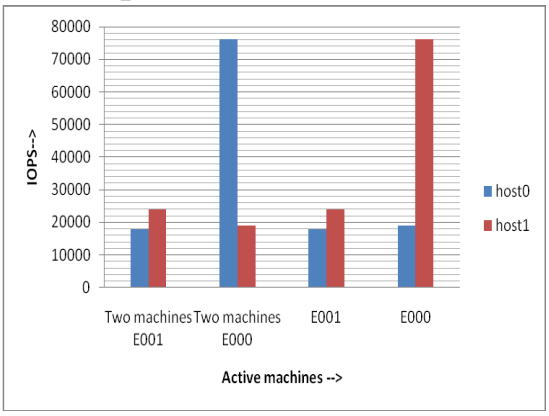

Figure.4 Analysis for Active machines Vs IOPS

Parameters	Simulation Results		
	E001	E000	Two machines running
IOPS	24000	76000	76590
THROUGHPUT (Mbps)	12.05	37.415	37.397
RESPONSE TIME (ms)	0.0561	0.05281	0.52308
SIMULATION TIME (s)	0.9584	0.62199	0.721302

Table.1 performance analysis of Cloud Server

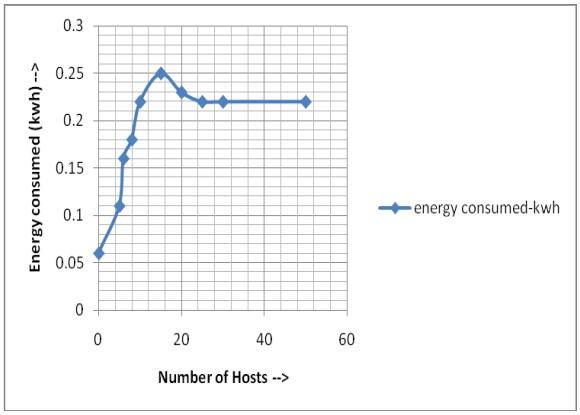

Figure.5 performance of Energy consumption Vs Number of Hosts

In figure.5 the analysis of energy consumed with the number of host is almost proportional for an increase in the density of host and the energy consumed is remains same after a certain density.

When the number of virtual migrations is less, the energy consumed is lesser. Energy consumed per kwh increases proportionally for a certain interval of Virtual machines migrations and then consolidates for the remaining VM migration samples taken as shown in figure.6.

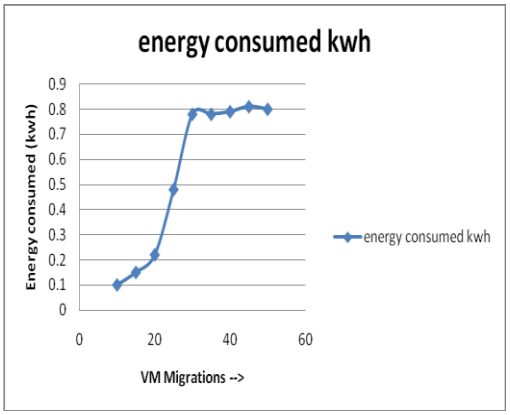

Figure.6

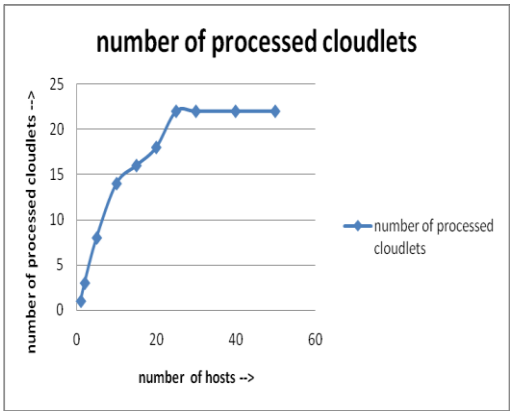

Figure.7

In figure.7 the number of processed cloudlets is minimum when the number of hosts is nil and increases proportionally when number of hosts increases and consolidates after a particular density of hosts.

V. CONCLUSION AND FUTURE WORK

From the examined experiments, the response time of the cloud server have been efficiently studied so as to not violate the above mentioned SLA's, the energy consumed and energy related parameters have been compared with the number of hosts and the virtual machine migrations.

My future work would be to harness the workload and the energy constraints for a heterogeneous scenario like the cloud environment.

VI.REFERENCES

[1] X. Wang and Y. Wang, "Coordinating Power Control and Performance Management for Virtualized Server Clusters, " Proc. IEEE Transactions on Parallel and Distributed systems, VOL 22, NO.2, FEBRUARY 2011.

[2] X. Wang and Y. Wang, "Co-Con: Coordinated Control of Power and Application Performance for Virtualized Server Clusters, "Proc. 17th IEEE Int'l Workshop Quality of Service (IWQoS), 2009.

[3] P.Mell and T.Grance. The NIST definition of Cloud computing. *National Institute of Standards and Technology* 2009.

[4] R. Buyya, C.S. Yeo, and S. Venugopal. Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In *High Performance Computing and Communications, 2008. HPCC'08. 10th IEEE International Conference on*, pages 5–13. IEEE, 2008.

[5] Kyong Hoon Kim, Rajkumar Buyya and Jong Kim "Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters", Grid Computing and Distributed Systems, university of Melbourne, Australia 2009.

[6] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam and Rajkumar Buyya. "Energy-Efficient of Scheduling HPC Applications in cloud computing Environments", IEEE Transactions on Distributed and parallel computing, 2009.

[7] Dzmitry Kliazovich, Pascal Bouvry and Samee Ullah Khan. GreenCloud: a packet-level simulator of energy-aware cloud computing data centers, Springer Science+Business Media, LLC 2010.

[8] X. Wang and M. Chen, "Cluster-Level Feedback Power Control for Performance Optimization,"Proc.Int'l Symp. High-Performance Computer Architecture (HPCA), 2008.

[9] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I.Pratt, and A. Warfield, "Live Migration of Virtual Machines," Proc. Symp. Networked Systems Design and Implementation(NSDI),2005.

