Information and Knowledge Management
ISSN 2224-5758 (Paper) ISSN 2224-896X (Online)
Vol 1, No.1, 2011

SLA Driven Load Balancing For Web Applications in Cloud Computing Environment

More Amar
Computer Engineering Department, Pune University, MAE, Alandi
Pune, Maharashtra 412105, India
amarmore2006@gmail.com

Kulkarni Anurag
Computer Engineering Department, Pune University, MAE, Alandi
Pune, Maharashtra 412105, India
anurag.kulkarni@yahoo.com

Kolhe Rakesh
Computer Engineering Department, Pune University, MAE, Alandi
Pune, Maharashtra 412105, India
rakeshkolhe139@gmail.com

Kothari Rupesh
Computer Engineering Department, Pune University, MAE, Alandi
Pune, Maharashtra 412105, India
kotharirupesh@gmail.com

Yahide Prashant
Computer Engineering Department, Pune University, MAE, Alandi
Pune, Maharashtra 412105, India
yahideprashant@yahoo.com

Abstract

Cloud computing is an emerging topic in the field of parallel and distributed computing. Many IT giants such as IBM, Sun, Amazon, Google, and Microsoft are promoting and offering various storage and compute clouds. Clouds provide services such as high performance computing, storage, and application hosting. Cloud providers are expected to ensure Quality of Service (QoS) through a Service Level Agreement (SLA) between the provider and the consumer. In this research, we develop a heterogeneous test bed compute cloud and investigate adaptive management of resources for Web applications to satisfy a SLA that enforces specific response time requirements. We develop a system on top of EUCALYTPUS framework that actively monitors the response time of the computed resources assign to a Web application and dynamically allocates the resources required by the application to satisfy the specific response time requirements.

Keywords: Eucalyptus, SLA, QOS, Virtualization, Minimum Response Time.

1 Introduction

1.1 Overview:

Cloud Computing is an emerging topic in the field of parallel and distributed computing. Many IT giants such as IBM, Sun, Amazon, Google, and Microsoft are promoting and offering various storage and compute clouds. Clouds provide services such as high performance computing, storage, and application hosting. Cloud providers are expected to ensure Quality of Service (QoS) through a service level agreement (SLA) between the provider and the consumer. Cloud providers need to establish a robust infrastructure that can grow dynamically, with easy maintenance and update.

A cloud is a combination of physically and virtually connected resources. Virtualization allows us to instantiate virtual machines dynamically on physical machines and allocate them resources as needed. Therefore, virtualization is one of the key technologies behind the cloud computing infrastructure. There are several benefits that we expect from virtualization, such as high availability, ease of deployment, migration, maintenance, and low power consumption that help us to establish a robust infrastructure for cloud computing.

EUCALYPUTS is an open source framework for developing clouds. Its primary developers are at the University of California, Santa Barbara. The aim of EUCALYPTUS is to enable academics to perform research in the field of cloud computing. In this research study, I use the EUCALYPTUS framework to establish a cloud and host a simple Web application on a Web farm of virtual machine instances.

1.2 Problem Statement

In Cloud computing, provider and consumer sign a service level agreement (SLA) that defines quality of service requirements. Cloud providers such as Amazon and Google allow consumers to the select resources they need for their cloud-based applications. Usually consumers reserve resources without any optimal prediction or estimation, leading either under or overutilization of reserved resources. Web application owners, in order to ensure the us- ability of their applications, need to maintain a minimum response time for their end users. Web application owners should be able to host their applications on a cloud by specifying a desire Quality of Service (QoS) in terms of response time. Cloud providers should ensure those QoS requirements using a minimal amount of computational resources. When violations of the SLA are predicted or detected, cloud providers should horizontally scale up the hosted Web application to ensure that the specific response time requirement is satisfied. Currently, neither the commercial cloud providers nor the existing open source frameworks support maximum response time guarantees.

1.3 Objectives

The main goal of this research study is to satisfy a cloud consumer that specifies Quality of Service (QoS) requirements in terms of response time by monitoring the average response time of the application and adaptively scaling up the Web application when necessary. Towards reaching this goal, the following are our specific objectives:

1. Understand EUCALYPTUS and establish a heterogeneous compute cloud test bed.
2. Write a simple Web based application and develop a workload generator for it.
3. Host the Web application on the cloud and establish a load balancer for the application.
4. Detect SLA violations by monitoring request response times.
5. Develop software components to horizontally scale up the Web application dynamically.

2 Literature and Technology Review

2.1 *Cloud Computing*

Cloud computing is an emerging technology in the field of parallel and distributed computing. Cloud computing is the delivery of computing as a service rather than a product, whereby shared resources, software, and information are provided to computers and other devices as a utility over the Internet.

Definition:“A Cloud is a type of parallel and distributed system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements established through negotiation between the service provider and consumers".

Storage and compute clouds are two major types of clouds that aim to provide services without exposing the underlying infrastructure to customers. A cloud service has three distinct characteristics that differentiate it from traditional hosting. It is sold on demand, typically by the minute or the hour; it is elastic -- a user can have as much or as little of a service as they want at any given time; and the service is fully managed by the provider (the consumer needs nothing but a personal computer and Internet access). Significant innovations in virtualization and distributed computing, as well as improved access to high-speed Internet and a weak economy, have accelerated interest in cloud computing. Instead of investing and managing personal hardware, users rent virtual machines or services from cloud providers.

2.2 Open source Cloud framework (Eucalyptus)

2.2.1 *EUCALYPTUS:*

EUCALYPTUS is an open source cloud computing framework developed by the University of California, Santa Barbara as an alternative to Amazon EC2. The aim of EUCALYPTUS is to enable academics to perform research in the field of cloud computing. It is composed of several components that interact with each other with well-defined interfaces. EUCALYPTUS therefore allows the open source community to replace or modify any component according to their requirements. Key functionalities of EUCALYPTUS are VM instance scheduling, construction of virtual infrastructure, administrative interfaces, user management of clouds, and definition and execution of service-level agreements. EUCALYPTUS allows the use of Amazon EC2 interfaces for interacting with the cloud. The three main components of EUCALYPTUS are the Node Controller, the Cluster Controller and the Cloud Controller. Each is described below.

2.2.1.1Node Controller:

The Node Controller (NC) is responsible for controlling the execution, inspection, cleanup, and termination of virtual machine instances on a physical machine. A EUCALYPUTS cloud may contain many NCs, but one physical machine only requires one NC. A single NC is therefore able to control all virtual machines executing on a single physical machine.

2.2.1.2 Cluster Controller:

A collection of NCs form Cluster Controller (CC) that typically executes on a head node or server that is able to access both the public and the private network. The CC is responsible for managing a collection of associated NCs.

2.2.1.3 Cloud Controller:

Each EUCALYPTUS-based cloud must include one Cloud Controller (CLC) that is the entry point for users. The CLC is responsible for making global decisions that include user authentication, processing of user or administrative requests, high level decisions for VM instance scheduling, and processing of SLAs

2.3 Virtualization:

Virtualization allows multiple operating systems to execution simultaneously on a physical machine. It is achieved through a virtual machine monitor (VMM). VMMs are also known as hypervisors. VMMs are responsible for keeping track of all activities performed by virtual machines. VMMs are categorized into Type-1 and Type-2 VMMs. Type-1 VMMs run directly on hardware without the need for a host OS, while Type-2 VMMs run on top of a host OS. Server virtualization can be achieved using Type-1 or Type-2 VMMs. Virtualization achieved through a Type-1 VMM is known as paravirtualization, and virtualization achieved through a Type-2 VMM is known as full virtualization.

On the X86 architecture, four privilege levels are available and known as Ring 0, 1, 2, and 3. Privilege levels are used to manage hardware resources to the applications and OS. Ring 0 is the most privilege level and Ring 3 is the least privilege level. Usually, OS instructions execute in Ring 0 and user application instructions execute in Ring 3. Virtualization needs to place VMM in the most privilege level (Ring 0) for managing the share resources for multiple operating systems.

2.3.1 Virtualization platform KVM:

The Kernel-based Virtual Machine (KVM) is a free virtualization platform licensed under GPL version 2, provides full virtualization. KVM contains a Type-2 hypervisor and supports Linux-based host operating systems. It allows Windows or Linux as the guest OS. The Ubuntu community provides KVM support.

A typical KVM installation consists of the following components:

- A device driver for managing the virtualization hardware; this driver exposes its capabilities via a character device /dev/kvm.
- A user-space component for emulating PC hardware; currently, this is handled in the user space and is a lightly modified QEMU process.
- The I/O model is directly derived from QEMU's, with support for copy-on-write disk images and other QEMU features.

3 System working and methodology

3.1 System Overview:

We develop a private network using four workstations to establish a heterogeneous cloud on the CS&IM LAN. We used the EUCALYPTUS framework to establish a cloud with one Cloud Controller (CLC), one Cluster Controller (CC), and three Node Controllers (NCs). We installed the CLC and CC on a front-end node that is on the CSIM LAN and the clouds private network. We installed the NCs on three separate machines connected to the private network. We used EUCALYPTUS to instantiate virtual machines on the node controllers.

Ngnix is a HTTP and mail proxy server capable of load balancing .We used Ngnix as a load balancer because it offers detailed logging and allows reloading the configuration file without termination of existing client sessions. We installed the Ngnix as a load balancer on the front- end node and the Tomcat application server on the virtual machine. We stored the virtual machine image in EUCALYPTUS, which cached it on all three NCs. Our Web application contains one Java servlet that performs a matrix calculation. The Tomcat server hosts the Web application while Ngnix balances the load. We used Httperf to generate a HTTP workload for the Web application. We developed a software component named VLBManager to monitor the average response time of requests by real-time monitoring of Ngnix logs. We developed another software component named VLBCoordinator to invoke the virtual machines on the EUCALYPTUS-based cloud. Both components interact with each other using XML-RPC. The VLBManager detects the violation of the average response time for each virtual machine in the Web farm due to the heterogeneous testbed cloud. Whenever VLBManager detects a violation of the average response time, it signals the VLBCoordinator to invoke another virtual machine and add it to the Web farm.

3.2 System Design:

3.3 Experiment: Adaptive resource allocation to Web application:

In this experiment, we used our proposed system to prevent response time increases and rejection of requests by the web server. . Initially, the load balancer is deployed with one virtual machine (vm1), while vm2 and vm3 are cached to the physical system using eucalyptus. In this experiment, we try to satisfy a service level agreement that enforces a two-second maximum average response time requirement for the sample web application on an undefined load level. We use vlbmanager to monitor the ngnix logs and detect the violations of the sla. We use vlb- coordinator to adaptively invoke additional virtual machines as required to satisfy the sla.

3.4 Results:

The figure 3.3.1 shows the number of requests served by the system. We observe linear growth in the number of requests served by the system with each load level. Figure3.3.2 shows the average response time of each virtual machine as it is adaptively added to the load balancer. Whenever the system detects a violation of the response time requirement from any virtual machine, it dynamically invokes another virtual machine and adds it to the Web farm. We observe continued violation of the required response time due to the latency of virtual machine boot-up.

3.5 Discussion:

Adaptive management of resources on compute clouds for Web application helps us offer

SLAs that enforce specific response time requirements. To avoid the continued violation of the SLA during VM boot-up, it would be better to predict violation of response time rather than to wait until the requirement is violated.

4 Conclusion and Recommendations

4.1 Overview:

Cloud computing is an emerging topic in the field of parallel and distributed computing and requires a huge contribution from the research community. This research study explored cloud computing and identified the advantage of adaptive resource management in heterogeneous clouds for Web applications. We developed a system on top of a EUCALYPTUS cloud that adaptively grows to satisfy specific response time requirements for Web applications.

4.2 Contribution:

This research identified a new kind of Service Level Agreement (SLA) for heterogeneous compute clouds in which the consumer wants specific response time for his or her Web application. We developed a system on top of EUCALYPTUS that actively monitors the response time of a Web application hosted on multiple virtual machines and adaptively manage the Web application's resources.

4.3 Recommendations

The CPU is typically the bottleneck in dynamic content generation and We observed that whenever response time starts to grow, the CPU usage exceeds above 90%. We recommend investigating system behavior by adaptively invoking other virtual machines by actively mon- itoring this behavior. To help overcome the virtual machine boot-up latency problem it would be useful to predict VM response times in advance then invoke additional virtual machines before the system exceeds the response time limit. It should not be necessary to check each VM response time if load balancing is fair instead of round robin. We would only need to check aggregated response time.

We also recommend moving VLBManager and Ngnix to a virtual machine and naming this VM as the head node. Whenever a user needs to deploy a Web application on cloud, the provider needs to deploy two virtual machines, one for the Web application and one for the head node to actively monitor the response time of the Web application hosted on the virtual machine. The head node can detect violations of response time requirements and adaptively deploy more virtual machines to satisfy the response time requirements.

References:

Postfix server configuration

http://www.server-world.info/en/note?os=Fedora_11&p=mail

Creation of certificate

http://www.g-loaded.eu/2005/11/10/be-your-own-ca/

Cloud Computing

http://en.wikipedia.org/wiki/Cloud_computing

http://searchcloudcomputing.techtarget.com/definition/cloud-computing

http://www.ibm.com/cloud-computing/us/en/

Eucalyptus and KVM configuration

http://open.eucalyptus.com/wiki/EucalyptusInstallationSource_v2.0

http://www.howtoforge.com/virtualization-with-kvm-on-a-fedora-11-server

Avi, K., Yaniv, K., Dor, L., Uri, L., & Anthony, L. (2007). KVM: The Linux Virtual Machine Monitor in Ottawa Linux Symposium (pp. 225–230).

Buyya, R., Yeo, C. S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. In HPCC '08: Proceedings of the

2008- 10th IEEE International Conference on High Performance Computing and c ommunications (pp. 5–13). Washington, DC, USA: IEEE Computer Society.

Google Code. (2008). Typica: A Java client library for a variety of Amazon Web services. Available at http://http://workspace.globus.org//[Online; accessed 06-March-2009])

Wikipedia. (2009b). Google App Engine. (Available at http://en.wikipedia.org/w/index.php?title=Google_App_Engine&oldid=273555538[Online; accessed 3-March-2009])

VMWare2007). Understanding full virtualization, paravirtualization, and hardware assist.Whitepaper.

Amazon.com, Inc. (2009). Amazon Elastic Compute Cloud (EC2). (Available at http://aws.amazon.com/ec2/ [Online; accessed 1-March-2009])

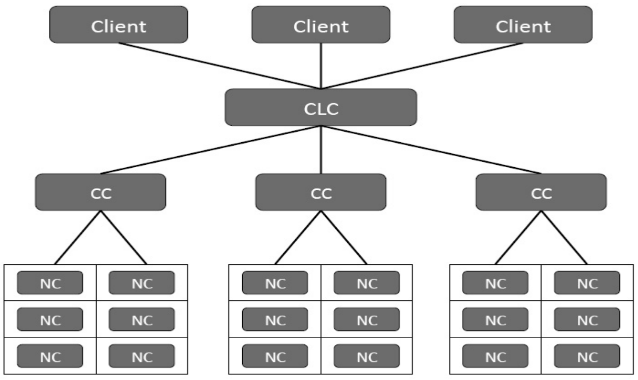

Fig2.2.1: Hierarchical interaction of Eucalyptus components

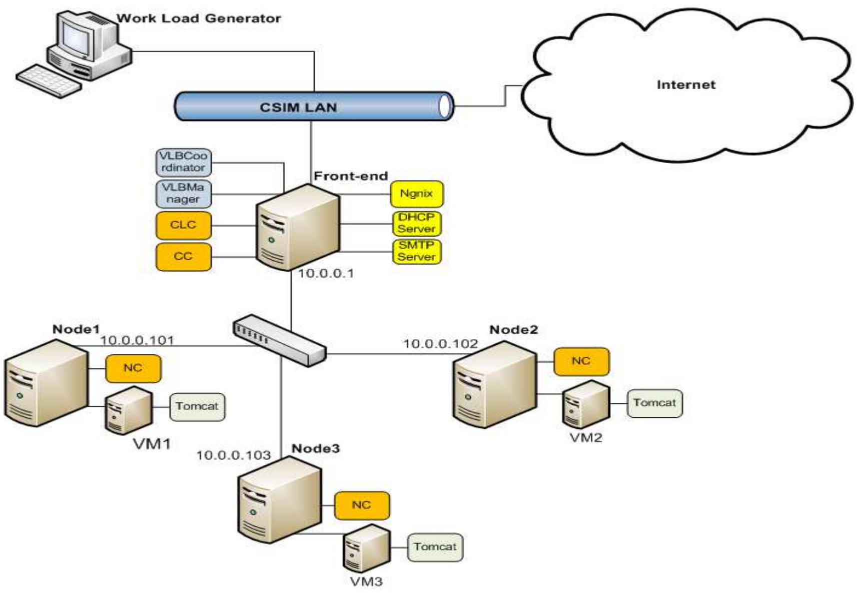

Fig3.2: Final System Architecture

Table: Hardware configuration of physical machines

Node Name	Type	CPU Frequency	RAM
Front-end	Intel Pentium	2.80 GHz	2 GB
Node1	Intel Pentium	2.66 GHz	1.5 GB
Node2	Intel Celeron	2.4 GHz	2 GB
Node3	Intel Core 2 Duo	1.6 GHz	1 GB

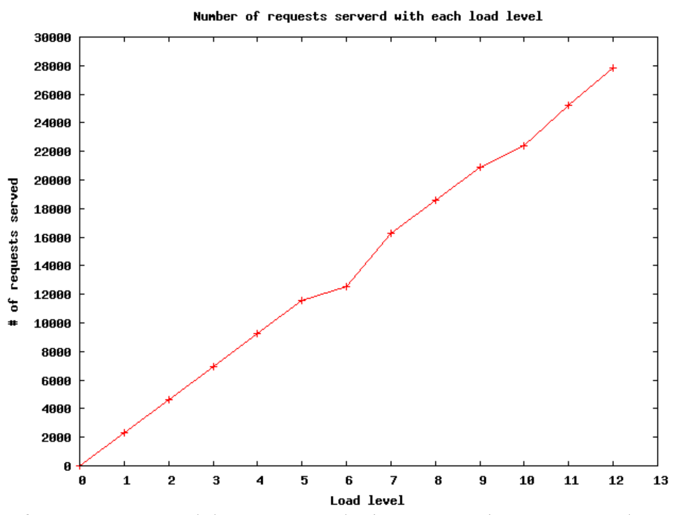

Figure3.3.1: Number of requests served by system during Experiment. We observe linear growth in the number of requests served by system with each load level.

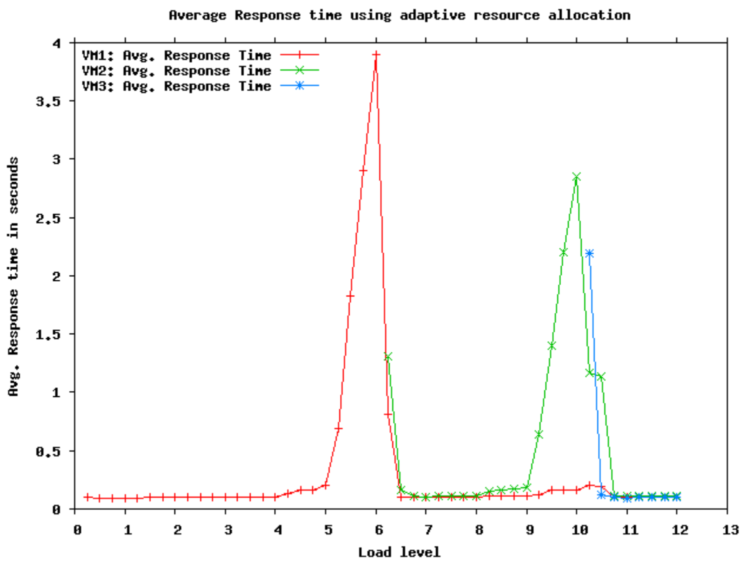

Figure3.3.2: Average response time for each virtual machine during Experiment. Whenever the system detects a violation of the response time requirement from any virtual node, it dynamically invokes another virtual machine and adds it to the Web farm.

This academic article was published by The International Institute for Science, Technology and Education (IISTE). The IISTE is a pioneer in the Open Access Publishing service based in the U.S. and Europe. The aim of the institute is Accelerating Global Knowledge Sharing.

More information about the publisher can be found in the IISTE's homepage:
http://www.iiste.org

The IISTE is currently hosting more than 30 peer-reviewed academic journals and collaborating with academic institutions around the world. **Prospective authors of IISTE journals can find the submission instruction on the following page:** http://www.iiste.org/Journals/

The IISTE editorial team promises to the review and publish all the qualified submissions in a fast manner. All the journals articles are available online to the readers all over the world without financial, legal, or technical barriers other than those inseparable from gaining access to the internet itself. Printed version of the journals is also available upon request of readers and authors.

IISTE Knowledge Sharing Partners

EBSCO, Index Copernicus, Ulrich's Periodicals Directory, JournalTOCS, PKP Open Archives Harvester, Bielefeld Academic Search Engine, Elektronische Zeitschriftenbibliothek EZB, Open J-Gate, OCLC WorldCat, Universe Digtial Library , NewJour, Google Scholar