

# Poster Abstract: Local Cloud Deployment in a Limited IP Setting

Jeffrey M. Galloway  
 Department of Computer Science, University of Alabama  
 Tuscaloosa, AL  
 jmgalloway@crimson.ua.edu

**Abstract**—This abstract is intended to be an introduction to our current research on cloud and distributed computing. The focus of our research began with the deployment of a cloud computing system. Realizing the implications of deploying such a system in an environment with limited networking resources (IP addresses), we decided to find a solution that would work giving our cloud only one public IP address. Users will be able to access cloud resources through a simple web interface and maintenance of the cloud will be contained with private networking resources.

**Index Terms**—Cloud Computing, IP Networking, Resource Distribution.

## I. INTRODUCTION

THE cloud computing architecture is becoming a dominant contender in the distributed systems paradigm. Its differences from the client/server architecture are based in its heavy use of resource elasticity. The cloud architecture uses virtualization technology in distributed data centers to allocate resources to users as they need them. Cloud computing has emerged from the previous industry standards like grid and cluster computing. Although cloud architectures are similar to these distributed systems, the resources are usually maintained by a single entity and might not be used by customers to complete similar goals. Depending on the level of control the user has (SaaS, PaaS, IaaS), jobs can differ greatly from customer to customer.

Following in the footsteps of cloud leaders such as Amazon and Google, open source communities have provided software packages that allow individuals to deploy their own local cloud. The topic of this paper is concerned with deployment of locally maintained clouds using limited networking resources.

## II. PROBLEM STATEMENT

Customers may be reluctant to move their data to off-site cloud vendors due to access regulations, cost, and trust issues. Instead, new software has become available for them to build their own personal cloud computing system. This gives the customer the ability to understand the issues and benefits of using cloud technology for distribution of resources before making the move to an enterprise vendor. There is a major problem with deploying a cloud locally; network resources in terms of IP addresses are usually expensive and therefore limited for smaller organizations.

We are proposing a solution that needs only one public IP address for resource distribution to users of the cloud. This

approach will provide benefits of more than just efficient use of network resources. By having only one public address associated with the cloud system, security vulnerabilities are decreased. This is due to the fact that all incoming and outgoing traffic to the cloud will be associated with one IP address. This address can be monitored more efficiently than if the entire cloud system were implemented on public IP addresses. In the case of our cloud construct proposition, no new network infrastructure will be needed in the organization.

## III. LOCAL CLOUD IMPLEMENTATION

Currently, our local cloud consists of seven Dell OptiPlex desktop computers, one network attached storage device, one Ethernet switch, and one Linksys NAT router. The underlying operating system of our cloud is Ubuntu Server 10.10 64-bit, which includes the Eucalyptus 2.0 open-source cloud architecture software [5]. Following the current specifications of deploying Ubuntu Enterprise Cloud (UEC) [1,2,4], we have configured the following machines to make up our cloud.

### A. Cloud Controller

This machine is the front end user interface to our cloud setup. It is hosted on a Dell OptiPlex 745 desktop (Core 2 Duo 2.8 GHz, 2GB RAM, 80GB hard drive). Its purpose is to provide the web interface to users while interacting with the rest of the components in the cloud setup. The cloud controller monitors the availability of resources of various components in the local cloud and monitors the running instances currently deployed on the node controllers.

### B. Cluster Controller/Storage Controller

The current deployment of UEC requires the cluster controller and storage controller to be on the same physical machine. These are hosted on a Dell OptiPlex 755 (Core 2 Duo 2.8 GHz, 2GB RAM, 160GB hard drive).

The cluster controller determines on which node controller a deployed instance will run. It also allows network access to the instances running on the node controllers. The cluster controller is responsible for load balancing virtual machines across all node controllers.

The storage controller gives persistent storage access to instance users. This service is similar to the elastic block storage service from Amazon Web Services.

Our current setup has only one cluster, therefore we have only one cluster controller.

### C. Walrus Storage Controller

Our walrus controller is hosted on a Dell OptiPlex 620 (Pentium D 2.93 GHz, 2GB RAM, 120GB hard drive). The

walrus controller stores machine images that can be launched as virtual machines in the local cloud. We are currently hosting two versions of Ubuntu, 9.10 and 10.04 for users to access.

#### D. Node Controller

The node controller is a machine with VT extensions on the CPU(s) used to host the running virtual machines in the local cloud. Our current setup has two node controllers. The first node controller is hosted on a Dell OptiPlex 755 (Core 2 Duo 2.8GHz, 2GB RAM, 160GB hard drive). Our second node controller is hosted on a Dell OptiPlex 960 (Core 2 Duo 2.93GHz, 2GB RAM, 160GB hard drive).

#### E. Backup Device

The backup solution for our local cloud is a Dell OptiPlex 270 (Pentium 4 3.2GHz, 2GB RAM, 120GB hard drive). We use this device to run a scheduled backup on all other devices in our setup. The device mounts folders to our NAS (Netgear Stora 1TB), which stores our nightly backups.

#### F. Client Device

This device is a Dell OptiPlex 280 (Pentium 4 2.8GHz, 2GB RAM, 120GB hard drive). This machine is used for testing the cloud. Instances can be launched, and cloud components can be accessed by ssh.

The router used in our setup is the Linksys WRT54GL. This device has been flashed with version 24 of DD-WRT [3]. This gives us more control over our network through the manipulation of ports through ssh connections, vpn access, and DHCP services. The open nature of the router will become important in the deployment of our local cloud.

As of now, users with private addresses have the ability to launch virtual machines in our cloud setup. This is an inconvenience since either the user has to be physically connected to our private network, or they need vpn accounts to access our cloud from a public address. The future work section will explain the direction of our research as it relates to connecting users to our cloud resources using a single public IP address.

### IV. PRELIMINARY RESULTS

Our idea of a local cloud setup was built with scalability in mind. Use of single machines, when available, has given insight on the network setup of a local cloud. With our current setup, we can run as many virtual machines as our node controller's resources allow us. Collectively we have four physical CPU cores, 4GB of RAM, and 320GB of hard drive space across two node controllers. We have the ability to add node controllers whenever we acquire new machines.

Kernel-based Virtual Machine (KVM) is the virtualization software used by default in UEC. The equation below gives the maximum number of virtual machines KVM will allow in any setup:

$$\# \text{ of VM's} = 8 * (\text{actual number of CPU cores})$$

Figure 1 is given from the cloud controller as it keeps track of the available resources in the cloud.

AVAILABILITYZONE	fluffy	192.168.1.100			
AVAILABILITYZONE	-vm types	free / max	cpu	ram	disk
AVAILABILITYZONE	-m1.small	0008 / 0016	1	192	10
AVAILABILITYZONE	-c1.medium	0006 / 0014	1	256	10
AVAILABILITYZONE	-m1.large	0003 / 0008	1	512	15
AVAILABILITYZONE	-m1.xlarge	0001 / 0004	2	1024	20
AVAILABILITYZONE	-c1.xlarge	0000 / 0001	4	2044	20

Figure 1: Available resources on the local cloud.

The total number of virtual machines our setup allows:

$$8 * (\# \text{ of actual cores} / 2)$$

This equation ensures that no single CPU will be overloaded (at most 4 vm's per core) and virtual machines will execute with sufficient processing resources. Other resources also play a role into the number of virtual machines available for launch such as RAM and hard drive space.

### V. WORK IN PROGRESS

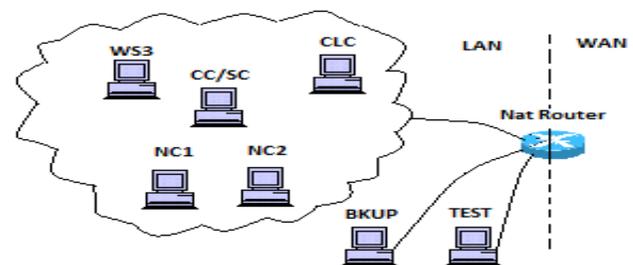
We are in the process of generating scripts that execute when a vm launches and terminates. Because of the single IP address, these scripts are needed to maintain a log file that lets users know which port and IP address to use when their vm has launched. The script that executes when a vm terminates updates the log file to report which ports have been freed and can be used in the next vm request.

We also only have one cluster of node controllers in our cloud setup. When more computing resources become available, we plan on expanding our cloud to multiple clusters maintained in different locations. The use of vpn enabled routers to connect these multiple clusters will be used.

Another aspect that should be considered is the implementation of a custom user interface that makes the launching of vm's easier for the user. This UI would be hosted in the web browser, and should tell the user information about their current number of vm's running, amount of resources using (quota should be applied), port numbers used to communicate with vm's, etc.

### VI. POSTER DESIGN

Layout of the poster will include: abstract, motivation, work in progress and references. All figures in this abstract, along with additional graphs will be displayed.



### REFERENCES

- [1] S. Wardley, E. Goyer, N. Barcet, "Ubuntu Enterprise Cloud Architecture," Canonical Ltd. August, 2009.
- [2] J. D., K. Murari, M. Raju, S. RB, Y. Girikumar, "Eucalyptus Beginner's Guide – UEC Edition (Ubuntu Server 10.04 – Lucid Lynx)," CSS. v1.0. May 2010.
- [3] <http://www.dd-wrt.com>
- [4] <http://www.ubuntu.com/cloud/private>
- [5] <http://open.eucalyptus.com/>