Experiences with Eucalyptus: Deploying an Open Source Cloud.

Authors:
Rick Bradshaw, Argonne National Laboratory, bradshaw@mcs.anl.gov
Piotr T Zbiegiel, Argonne National Laboratory, pzbiegiel@anl.gov

Tags: cloud, cluster, Eucalyptus, security, scalability, support, HPC, virtual machines, research

Abstract

With the recent trend of exploiting resources of the cloud, we have embarked on a journey to deploy an open source cloud using Eucalyptus[1]. During the past year we have learned many lessons about the use of Eucalyptus and clouds in general. The area of security provides significant challenges in operating a cloud, the scalability supposedly inherent in clouds isn't a given, and the process of supporting cloud users is different and more complicated than supporting desktop or even HPC users.

1. Introduction

Our initial investigation of Eucalyptus was as an attempt to see how we could adapt our current on-demand compute cluster to provision virtual machines. We already had an existing solution for raw hardware provisioning, but we wanted to use Eucalyptus to provide virtual machines to users through a well-defined interface. This early research showed that Eucalyptus could be adapted to work within our current development cluster.

When it came time to do a public implementation for the Magellan project[2], we continued to use Eucalyptus because it has many features important to the goals of the project. Most of the information for this paper comes from our experiences in deploying Eucalyptus on our Magellan cluster. One of the major goals of Magellan is investigating high-performance computing (HPC) in the cloud. It was built as a traditional HPC cluster with an Infiniband network attaching all of the nodes but running a Eucalyptus cloud software stack instead of a more traditional HPC cluster batch system.

The team tasked with the deployment comprised HPC and network administrators, cyber security, and user services staff, a mixture chosen because of the common issues we face when deploying any new service to users. The first major issue we will discuss is scalability. When one thinks of a cloud one envisions thousands of machines, so the software should be ready to handle that type of workload. The second issue is security. Any service open to the public has this issue on its list of concerns; but when dealing with random users with root privileges the situation is more problematic. The third issue we will cover is user support, and how the problems can become time-consuming for both user support staff and the users themselves. The overall process has produced a working cloud, and we are much wiser for the effort.

2. Scalability

Early on while working with Eucalyptus we put the system through a number of scalability tests. The goal was to identify sizing limitations for Eucalyptus cluster controllers. Eucalyptus permits a cloud system to include a number of cluster controllers. Our testing was limited to a single cluster controller and it identified several areas of limitation.

The Eucalyptus network model forwards all traffic from the virtual machines running in each cluster through the cluster controller. This setup allows the system to implement security groups more easily however it also creates a potential network bottleneck for the running virtual machines. Even a moderate amount of network traffic spread across many virtual machines can saturate the cluster controller's network bandwidth.

Testing of the virtual machine capacity revealed that Eucalyptus had a limit to the number of simultaneously running virtual machines. Because of a message size limit in a communication protocol, we were able to identify a hard limit to the number of simultaneously running virtual machines on a cluster. This limit falls between 750 and 800 running virtual machines and results in the cluster being unable to service additional instance requests.

Many of the cluster controller operations are iterative. While this setup does not define hard limits for the system, it does mean that as the cluster is extended, certain operations take longer to complete. When the cluster was expanded to more than 200 node controllers, there was a noticeable deterioration of performance during certain operations. For instance terminating large numbers of running virtual machines can cause delays for other operations such as new virtual machine requests.

3. Security

Security on a machine like Magellan is a key concern. Users have the ability to load and start up their own versions of various operating systems, and they retain administrative access to those virtual machines. While building and testing the cluster, we identified several areas that we felt needed to be addressed to ensure security.

Security testing clearly showed that the virtual machines could communicate far more broadly than we had anticipated. The cluster controller that handles all network communications for the virtual machines keeps tight control of ingress communications using security groups; however, egress filtering is limited. In addition, the cluster controller has blanket IP masquerade rules to facilitate communication by virtual machines that do not have publicly routable addresses. This feature often makes it difficult to identify the source of traffic exiting the cloud. The issue that raised most concern was that virtual machines could contact not only the node controller on which they were running but also all other node controllers with the help of the IP masquerading rules on the cluster controller. While this situation does not lead directly to a compromise, it does expose additional attack surface to the virtual machines running on the cloud. Considering the difficulty of tracking communications within the cloud we felt this issue required a solution. We identified proper iptables rules to filter traffic exiting the virtual machines.

The risk of massive amounts of traffic being generated by a user whether malicious or not, is real and serious. Since the virtual machines have fairly open egress they can send massive amounts of data. While we do not wish to impede data movement into and out of the cloud, we do need to identify legitimate and illegitimate data movements and provide a way to actively

slow or stop the illegitimate traffic. An IDS system with active response capabilities was implemented to watch traffic passing through the Eucalyptus cluster controller.

One area of debate for the team has been end-user virtual machine image management. Currently, Eucalyptus allows normal users to upload and register new disk images but it does not allow them to register kernel or ramdisk images. Users must rely on the administrators to make kernels available for use and then couple their disk images with the correct kernel. The concern stems from how Eucalyptus handles newly uploaded images. By default, Eucalyptus registers new images as public and available for all users. Eucalyptus provides commands that can be run to change those permissions but unless the user takes extra steps the image will be available to all users of the machine. This raises a number of issues. First, there is the concern that users may upload an image that, unbeknownst to them, is made public. That image may contain information that they considered sensitive but it is now made available to all users. Second, a malicious user could upload an image loaded with malware (keystroke loggers, back doors, etc.) and it will be made public for use by all users of the system. If other users unwittingly use one of these images, their virtual machines instances would be compromised.

Virtual machine image management can also affect the operational side of Eucalyptus. If users can post new kernels, ramdisks, and disk images that are all made public, normal end users will be presented with a dizzying array of images every time they run *euca-describe-images*. They may choose an image at random that doesn't work or is compromised by the original poster, leading to support requests and security incidents. Administrators will have to regularly review the list of images and cull if necessary.

The debate over this issue continues. We do not want to limit scientific inquiry, but it must be balanced with security and operational concerns that affect the system as a whole.

4. User Support

In order to choose a support plan for our users we looked to the eucalyptus model of support. Eucalyptus has two types of software available: a pay for version with company support and a free version with user forum support. We chose the latter, to which we added a few extra features. We generally want the source code to tweak or to integrate one of our other pieces of software. A user writable wiki with updated how-tos, as well as user mailing lists, not just for help desk support, but for community support were created to help build a commuity. As one would assume, documentation is critical when trying to support this style of operation. Outside of documentation is the issue of updates to the cloud. Once users on the system are trying to teach themselves proper usage, how can the support staff test updates to the software. In the past, maintenance days were used to update or upgrade machines, but with a cloud it seems counter intuitive to have downtime, so there needs to be a development environment large enough to be able to test at a reasonable scale.

Users tend to stick to what they know and understand, so many find it a huge jump to move from supporting their own applications or data sets to supporting an entire OS and tool chain to make it possible to run their applications or analyze their data sets. The small switch from "here are the options you need for that compiler" to "here are the steps that *should* get your compiler installed" can cause a large burden on the support structure for a cloud. Now users need to understand and be able to deal with the intricacies of OS administration. Along these lines is the education of users about all the security policies and protocols is also an extra burden and one

that most users simply do not care about. They have a specific piece of science they need to compute and they used to be able to simply submit jobs to handle that, Now, however, they have to meet all the requirements that are normally taken care of behind the scenes by the administrative or security staff. In most cases, by the time the users know enough to be good users of the cloud, they are junior sysadmins. It is difficult to support something for which the user support staff may in turn need support. The response cycle can be variable based on the bugs and the severity of the bug. A situation that certainly can ruin the time to deployment for many projects.

Arguably, there is an upside to this support model. Users ultimately do get an unchanging environment to run their software in, which should then be portable across any instance of a cloud. Moving from different research clouds and even to Amazon EC2 should be fairly effortless and maintainable over the long term.

5. Conclusions

Over the past year the team at Argonne has worked to build and maintain a scientific cloud system based on Eucalyptus. Three major challenges have been scalability, security, and user support of the cloud. Early on review of the architecture and testing of the Eucalyptus software revealed several scalability issues. Some of these could be addresses while others have persisted to this day in the cloud system. With practically no access to user's virtual machines efforts to improve security on the system have concentrated on network-based solution to identify and limit malicious activity. User support can create a large work load depending on the level of support promised to the user community. Merely providing users with documentation and how-tos helps but isn't enough. A community-based support model has helped ease the burden on the user support organization while providing timely help to users of the system.

In the future we will be expanding our development environment to test other cloud stacks including OpenStack[3] and Nimbus[4] which are both open source cloud stacks. We also are planning on integrating our Infiniband network into our cloud by using I/O forwarding technology to allow GPFS mounts on virtual machines, and if possible we will get ethernet bridging to work over IB and then the entire cloud could make use of the high speed network.

This research used resources of the Argonne Leadership Computing Facility at Argonne National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under contract DE-AC02-06CH11357, funded through the The American Recovery and Reinvestment Act of 2009.

6. References

1. Eucalyptus. (n.d.). Eucalyptus Community. http://www.eucalyptus.com
2. Magellan Cloud. (n.d.). Magellan | Argonne's DOE Cloud Computing. http://magellan.alcf.anl.gov
3. OpenStack (n.d.). OpenStack Open Source Cloud Computing Software. http://www.openstack.org
4. Nimbus (n.d.). Nimbus. http://www.nimbusproject.org