***IITG Project 2013-2014***

***OpenStack Implementation***

Christopher Rigby christopher.rigby@sunyorange.edu

Cartmell Warrington cartmell.warrington@sunyorange.edu

Our project implements an OpenStack cluster aimed at providing a virtual machine environment for the deployment of laboratory modules on preconfigured OS images. Our architecture has several major benefits:

* providing preconfigured laboratory images limits the amount of time necessary for students to set up the laboratory environment, allowing more laboratory time to be devoted to learning objectives
* it provides a platform for the development of such laboratory modules, which can be shared with learning partners
* it allows us to introduce relevant, cutting edge technology such as virtualization environments, infrastructure-as-a service, and software-defined networking components to technology students.

For those interested in replicating our project, this document outlines the technologies used and some of the issues encounted.

The primary technology used was ***OpenStack***, which is an open source “cloud computing” platform. OpenStack’s web site is located at:

 http://www.openstack.org/

We chose to deploy the “Grizzly” release, however we are currently experimenting with the Icehouse version, and may migrate to that release soon.

We primarily deployed on the ***Ubuntu*** Linux Operating System:

 http://www.ubuntu.com/

Our particular platform was ***Ubuntu Server 12.04 LTS***

 http://releases.ubuntu.com/12.04/

There are precompiled OpenStack packages available for Ubuntu, see the Installation Guide section below for links and resources.

The following ***Installation Guides*** were valuable in the installation and configuration of our environment:

 http://docs.openstack.org/grizzly/openstack-compute/install/apt/content/

 https://github.com/mseknibilel/OpenStack-Grizzly-Install-Guide

However, the ***openvswitch*** packages that come with Ubuntu seem to have problems with later kernels. An easy fix is to obtain the source and build them yourselves. If, like us, you implement on Ubuntu, you can build Debian packages and install them via dpkg.

 http://openvswitch.org/download/

Our OpenStack architecture consisted of:

* 1 compute node
* 1 network node
* 18 compute nodes

As to our hardware build, each node consisted of:

* Motherboard
* Intel i7 quad core processor
* 32 gigs RAM
* 2 NIC cards (3 for the network node)

This configuration is robust enough to support upwards of 80 powerful virtual machines (multiple processors and several gigs of RAM per VM). We can run several classes concurrently without putting a strain on cluster resources.

Once the cluster is operational, there are several places one can obtain prebuilt OS images:

* http://docs.openstack.org/image-guide/content/ch\_obtaining\_images.html
* https://launchpad.net/cirros/+download
* http://cloud-images.ubuntu.com/
* http://fedoraproject.org/en/get-fedora#clouds
* https://github.com/rcbops/oz-image-build

For those with specific OS requirements, several sites offer instructions and advice on constructing custom images:

* http://docs.openstack.org/image-guide/content/ch\_creating\_images\_manually.html
* http://docs.openstack.org/image-guide/content/ch\_creating\_images\_automatically.html
* http://www.ibm.com/developerworks/cloud/library/cl-openstack-images/
* http://networkstatic.net/building-a-windows-image-for-openstack/

We support a growing collection of Operating System images, some geared toward particular operations

Some examples:

 Software Development

 OS: Ubuntu Desktop

 Software: gcc/g++, JDK, python, perl, clisp, swi-prolog, ruby, Octave, DDD, Eclipse

 Database Design

 OS: Ubuntu Desktop

 Software: MySQL, Postgresql, sqlite3, mongodb

 LAMP Server

 OS: Ubuntu Server

 Software: Apache, MySQL, PHP5

One of our goals is to provide preconfigured operating system images, along with high-quality, self-contained laboratory modules that can be shared with learning partners. Some example of modules that we have already developed are:

*Operating Systems*

* OS installation procedures: Fedora, Ubuntu Desktop, Ubuntu Server, Windows 7
* RAID Arrays – construct software RAID 1 and RAID 5 arrays on Linux and Windows
* File Sharing – configure SaMBa and NFS file shares
* iSCSI Deployment – create and configure iSCSI targets and mount them on a variety of Operating Systems.
* Process Monitoring on Windows and Linux – basic tools for Process Management
* Memory Management on Windows and Linux
* Virtualization Basics – deploy and remotely access different virtual machine instances on OpenStack. Use Eucalyptus and Hybridfox for VM deployment. Introduction to boto scripting and automated deployment and monitoring.
* Networking – sockets and servers in Python. Work in pairs to write very simple client and server scripts in Python. Use Wireshark to capture packets between client and server. Write a Python script that mimics the behavior of the nmap port scanner.

*Linux*

* Installation – perform a full installation of Linux (Ubuntu Desktop and Server, as well as Fedora versions available).
* The Shell and basic commands – introduction to the command line interpreter and basic Linux commands. File permissions.
* Working with the bash shell – advanced commands, piping and redirection. Grep, sed, and awk. Basic shell scripting.
* System Initialization and Linux processes – the Linux boot sequence; process management with ps and top; process scheduling with cron and at.
* Filesystems – Linux file system hierarchy, add raw disks to a system, partition using fdisk, create filesystems, mount and unmount, add entries to fstab.
* Administrative Tools – add and modify user accounts.
* Software Installation – install software with the package manager (apt-get, dpkg), as well as from source code. Perform an installation of the Apache Web server from source.

*Databases*

* MySQL introduction – the MySQL shell, the help system, users and permissions, create a database, grant privileges.
* MySQL Workbench – administration and monitoring with the Workbench tool. Create an E-R diagram from an existing database.
* Backups and Replication – backup and restore a database using mysqldump. Configure simple primary-secondary replication between two databases.

And several more.

We are interested in collaborating on sharable laboratory content, and are enthusiastically seeking feedback on our efforts, particularly concerning the scope and quality of our platform and materials. For further information please contact us.