Mr. Jason St. John, Purdue University, West Lafayette

Jason St. John earned a Bachelor of Science degree in Computer & Information Technology (Network Engineering Technology concentration) from Purdue University, West Lafayette in December 2010. In January 2011, Jason accepted a position as a Graduate Research Assistant in the Department of Computer & Information Technology at Purdue University working under Dr. Thomas Hacker. Jason is currently pursuing a Master’s Degree in high performance computing systems.

Prof. Thomas J. Hacker, Purdue University, West Lafayette

Thomas J. Hacker is an Associate Professor of computer and information technology at Purdue University in West Lafayette, Ind. His research interests include cyberinfrastructure systems, high-performance computing, and the reliability of large-scale supercomputing systems. He holds a Ph.D. in computer science and engineering from the University of Michigan, Ann Arbor. He is a member of IEEE, the ACM, and ASEE.
Developing Virtual Clusters for
High Performance Computing Using OpenNebula

Abstract

High performance computing cluster kits such as the Rocks Cluster Distribution ("Rocks") and OSCAR were developed to simplify the process of installing, configuring, and managing commodity-based clusters. The recent emergence of virtualization technologies and IaaS cloud computing systems, such as OpenNebula and NIMBUS, have now greatly simplified the process of creating, deploying, and managing dozens to thousands of virtual machine images running on a collection of physical servers. For a graduate course in high performance computing systems, we are using the KVM virtualization system with OpenNebula to create virtual high performance computing systems. In this paper, we describe our experiences in investigating alternatives to OSCAR and Rocks to develop a virtual high performance computing clustering environment built on KVM and OpenNebula. We believe our experiences and results will be useful for other educators seeking to efficiently use limited physical hardware resources to develop an effective course in high performance computing systems.

1 Introduction

Developing and deploying Beowulf-style high performance computing clusters has always been a difficult process for many reasons. Traditionally, Beowulf-style HPC clusters consist of a large number of computing nodes, and each of those nodes needs to have an operating system installed on it. To complicate matters, some HPC jobs require Microsoft Windows, which means that system administrators must potentially configure a subset of the computing nodes to become dual-booting systems. Once all of the nodes are operational, system administrators need to be able to control the nodes in batches, while maintaining the integrity of the systems. Even with shell scripts and other automation tools, this process has traditionally been tedious at best. To mitigate the problems faced with deploying clusters, a number of projects have been created called “cluster kits”. These software toolkits are designed to automate a large portion of the installation and configuration procedures.

Additionally, businesses and universities of all sizes are moving their traditional IT infrastructures to the recently emergent technologies of virtualization and cloud computing. This has created a significantly growing need for knowledgeable workers that are able to design, deploy, and troubleshoot these complex environments. Consequently, IT instructional programs must offer effective courses in teaching these concepts so students are able to develop the skills necessary to meet the growing demand by organizations.
While researching potential virtualization-aware alternatives to traditional cluster kits—which are not necessarily targeted towards virtualization or cloud computing-based environments—we discovered that an infrastructure as a service (IaaS) cloud computing toolkit would be a sensible alternative to cluster kits, while also providing a suitable learning environment. After comparing the various open-source IaaS toolkits available, we chose to evaluate OpenNebula combined with the QEMU-KVM hypervisor in the course.

The course is titled “Advanced High Performance Computing Systems” and is a graduate level course. The course material covers HPC cluster architectures and design, commodity hardware, graphics processing units (GPUs), system reliability, job scheduling, lightweight kernels and operating systems, virtualization, and fault tolerance, among other topics. The course is offered at Purdue University–West Lafayette in the Department of Computer and Information Technology.

2 Existing Problems (In Practice)

The design, construction, and deployment of commodity hardware-based high performance computing clusters remains a difficult task in practice, despite several attempts to create easily installed “cluster kits”. There are numerous existing problems with cluster building in practice, ranging from complex configuration requirements to infrequently updated, automated software tools for installing clusters. There are three general methods, consisting of the following: CFEngine, installing the clusters manually, or using a cluster kit, such as the Rocks Cluster Distribution (“Rocks”) or OSCAR.

CFEngine is a software package designed for large-scale configuration management. CFEngine only supports KVM, VMware, and Xen under a commercial license, making it infeasible for a graduate course. System administrators can install clusters without the use of automated tools (manually), which is a very tedious task even when other tools are used to help automate the process. As an alternative, system administrators can use a cluster kit to assist in host operating system installation via PXE booting to configuring the package selection and networking for the cluster hosts. The three most notable cluster kits are: OSCAR (Open Source Cluster Application Resources), the Rocks Cluster Distribution (“Rocks”), and the now-defunct openMosix project.

OSCAR is a cluster kit developed primarily by Oak Ridge National Laboratory (ORNL) and is a community-driven open-source software project. OSCAR version 1.0 was released in 2001 and is currently on version 6.1.1, which was released in 2011.

Rocks is a distribution of GNU/Linux produced by the San Diego Supercomputer Center (SDSC) explicitly targeted at cluster computing using virtualization. Rocks was previously known as the “NPACI Rocks Cluster Distribution” and was created in 2000 as a partnership between SDSC and NPACI.

According to the openMosix home page, “openMosix is a Linux kernel extension for single-system image clustering which turns a network of ordinary computers into a supercomputer.” On 2008-03-01, openMosix officially hit its end-of-life and is therefore not a suitable piece of technology for a modern HPC course. The LinuxPMI project is essentially a
fork of the old openMosix 2.6 development branch and can be considered the successor of openMosix.

2.1 OSCAR and OSCAR-V

OSCAR’s pace of development appears to be significantly slowing, as evidenced by the drop-off in activity during 2009–2010 on the oscar-checkins\textsuperscript{14}, oscar-devel\textsuperscript{15}, and oscar-users\textsuperscript{16} mailing lists. OSCAR also seems to be losing momentum or showing its age in other areas:

- OSCAR does not officially support Red Hat Enterprise Linux (RHEL) 6—released in 2010—or CentOS 6.
- OSCAR’s IRC channel is inactive.
- OSCAR has not had a single Subversion check-in in over 8 months as of the time of this writing\textsuperscript{14}.
- There are many outdated web sites and pages for OSCAR development, from the original OSCAR development site\textsuperscript{17} to SourceForge to many pages on OSCAR’s current web site.

OSCAR-V was introduced in 2007 as a virtualization-enabled version of OSCAR, that was to be used to build virtual HPC clusters, but OSCAR-V is not currently usable. The OSCAR-V project has not released any software to date. Additionally, OSCAR-V’s Trac home page has been completely overrun by spam, and OSCAR-V’s issue tracker currently has over 32,000 tickets created by spam bots\textsuperscript{4}. Based on this evidence, we have concluded that there are better alternatives to OSCAR and OSCAR-V.

2.2 Rocks Cluster Distribution

As of this writing, Rocks has two significant limitations in its adoption abroad, thus limiting its applicability to a course in the HPC field aiming to avoid “technology lock-in”. The first limitation is that Rocks only supports the Xen hypervisor, and the second limitation is that Rocks only supports RHEL 5 and CentOS 5. Although, the Rocks team from SDSC stated in 2009 that Rocks will switch from Xen to KVM with RHEL 6\textsuperscript{20}, there is not a release of Rocks that supports RHEL 6 to date.

2.3 openMosix/LinuxPMI

LinuxPMI does not have a stable release yet, and the pace of development is slow. LinuxPMI’s support of kernels 2.6.17 and 2.6.28.7—neither of which is provided by default in RHEL or CentOS—significantly limits its usefulness in the HPC field.

While these projects have been effective and useful in the past, we do not feel that these projects meet the needs of a state of the art, large-scale, virtualized infrastructure for HPC.
Emergence of Virtualization and Cloud Computing

In the mid-2000s, virtualization and cloud computing experienced massive growth and continue to grow today. The resurgence of virtualization and the dawn of cloud computing have enabled a transformation in Beowulf-style computing clusters from compute nodes running directly on bare hardware to compute nodes running as virtual machines that can be managed as a cloud. Cloud computing provides numerous advantages, including the following:

- Enables easier management of single-system images for virtual HPC clusters
- The ability to create and easily provision/deploy virtual machine images
- A mixed operating system environment no longer requires dual-booting
- The ability to tailor virtual machine images to the specific requirements of a given job

Because of these advantages, cloud computing—most notably of the infrastructure as a service (IaaS) variety—is and will continue to be a major technology for HPC.

Existing Problems (In Academia)

A major problem in academia is how to best facilitate the teaching of HPC cluster development that does not depend on immature or outdated cluster kits. Because virtualization and cloud computing are changing the way in which businesses and research organizations use HPC, a modern course in HPC systems needs to ensure that students are learning essential skills that can be applied broadly. Additionally, learning how to build clusters—virtual or physical—is not easily taught from a purely lecture-based model and requires an in-depth laboratory component that gives students hands-on experience with virtualization and open-source software.

Solution

Our solution to these problems was to use recycled computer lab desktops (with hardware acceleration for virtualization), in addition to commodity network switches to provide the physical infrastructure needed to deploy a small, virtualized Beowulf-style computing cluster that uses open-source software. A set of assignments was developed that required students to plan, install, configure, and benchmark a virtual HPC cluster managed by OpenNebula. These assignments were the laboratory component of a lecture course that covered the theoretical background of various HPC topics, especially those critical to completing the assignments. These assignments used QEMU with KVM as the hypervisor, and students were given the option of using any GNU/Linux distribution they wanted as the operating system for both the physical and virtual machines.

OpenNebula was chosen to manage and provision the virtual machines used in the cluster. OpenNebula was chosen for several reasons, including the following:

- OpenNebula is open-source software with a strong community and responsive developers.
• OpenNebula uses many software tools and libraries that are standard on almost every GNU/Linux distribution—such as SSH, NFS, Bash shell scripts, etc.—which makes learning how the OpenNebula system works behind the scenes much easier.

• OpenNebula allows for choice—the choice to use any hypervisor that libvirt supports, the choice to use multiple deployment models, the choice of a cloud interface, and the choice of tools to handle scheduling of jobs, among other things.

• OpenNebula is used extensively by CERN\textsuperscript{19} and others with great results.

The set of assignments were intended to provide the following educational outcomes:

• Hands-on experience in developing and deploying a virtual HPC cluster

• Hands-on experience in virtualization, cloud computing technologies, and networking

• Introduction to MPI (message passing interface)\textsuperscript{3}, high performance LINPACK (HPL)\textsuperscript{21}, and HPC system schedulers like TORQUE\textsuperscript{18}

• Students exposed to an IaaS tool via OpenNebula

• Students gain experience in planning, designing, building, and benchmarking deployed VMs within each student’s virtual HPC cluster

6 Our Approach

Each student was assigned 16 recycled computer lab desktop computers and a recycled monitor from Purdue University’s IT department, a commodity Gigabit Ethernet network switch, a few carts to store the computers and monitor on, and a number of other peripherals and cables required to use the equipment. Each desktop computer was a Dell OptiPlex 745\textsuperscript{9} with an Intel Core 2 Duo 6400 processor, 2 GBs of DDR2 RAM, and an on-board Gigabit Ethernet NIC.

The first of the two assignments had the students physically assemble the cluster, prepare the hardware for use, and install and configure a GNU/Linux operating system for virtualization with OpenNebula. The second of the two assignments had the students install and configure the OpenNebula system, including a base virtual machine image that supports MPI and TORQUE, in addition to submitting a job that runs at least one of the OSU Micro-Benchmarks\textsuperscript{7} for evaluating virtual machine performance.

The assignments were progressive in nature, and they were designed such that the general required tasks would be grouped together logically and that the assignment deadlines would function as project milestones. The first milestone was composed primarily of tasks that are necessary to prepare for an OpenNebula deployment, while the second milestone was the actual deployment of the OpenNebula system including the virtual machines and benchmarking.
6.1 Evaluation

All five of the students were able to satisfactorily complete the set of assignments by the end of one semester. Satisfactory completion required students to complete the following:

- Install and configure the assigned desktops as OpenNebula VM hosts using the QEMU-KVM hypervisor.
- Install and configure the TORQUE HPC system scheduler for job submission.
- Create and configure a base virtual machine image as a TORQUE client that supports MPI.
- Using OpenNebula, deploy multiple clones of the base VM image to the configured VM hosts.
- Run at least one MPI benchmark from the OSU Micro-Benchmarks suite across all deployed VMs.
- Submit a screenshot of the benchmarking results to the course instructor.

Students found the deployment of OpenNebula to be challenging, but they found the experience to be rewarding. The students were impressed with what they were able to do with the OpenNebula system, as well as how easy OpenNebula is to manage once it is configured. Despite the challenges faced—both conceptually and technologically—the students still enjoyed the learning process and found themselves pleased with the amount of knowledge learned about GNU/Linux systems, virtualization, and IaaS deployments.

6.1.1 Lessons Learned

Based on observations by the authors and feedback from the students, we came to these conclusions, which we believe improve upon the initial two assignment model introduced in this paper:

- Increase the number of assignments (i.e. milestones), while reducing the scope of each.
• Add a preliminary introductory assignment that requires students to logically design the deployment at a high level, so that students are able to better understand the OpenNebula deployment before they even touch the hardware.

• Create an extra assignment—as a requirement or as extra credit—that has students write a shell script or find a utility to automate shell commands that must be executed on each VM host.

7 Conclusion

While there are many difficulties associated with HPC cluster deployment—both in industry and in academia—the OpenNebula IaaS system has shown itself to be effective at teaching the art of virtual cluster building, as well as managing very large clusters, like those operated by CERN, making the skills learned in the course both practical and useful for the students and their future employers.

Bibliography

[16] oscar-users mailing list archives. URL  


[18] Adaptive Computing Enterprises Inc. TORQUE resource manager. URL  


[20] Philip Papadopoulos, September 2009. URL  
http://marc.info/?l=npaci-rocks-discussion&m=125342538122884.

[21] Antoine Petitet, R. Clint Whaley, Jack Dongarra, and Andrew Cleary. HPL - a portable implementation of  
the high-performance linpack benchmark for distributed-memory computers, September 2008. URL  
http://netlib.org/benchmark/hpl/.

[22] Geoffroy Vallée. OSCAR 6.1.1 has been released!, May 2011. URL  