Extending a Desktop Computing Grid with Cloud Resources

Ben Cotton
Purdue University
155 S. Grant st
West Lafayette, Indiana 47907
bcotton@purdue.edu

Andrew Howard
Cray, Inc.
12676 Jade Run #2
Beavercreek, OH 45431
ahoward@cray.com

Preston Smith
Purdue University
155 S. Grant St
West Lafayette, Indiana 47907
psmith@purdue.edu

ABSTRACT

This paper discusses the work at Purdue University to extend scientific computing capacity by scavenging cycles from desktop machines around campus. Using the Condor[1] software developed at the University of Wisconsin and VMWare virtual machines, Windows desktops in instructional labs can be dynamically provisioned as Linux compute nodes to provide additional capacity for research computing. The goal of this work is to provide distributed computing resources that match the needs of faculty and student researchers without the need to purchase additional equipment. Initial testing efforts have lead to a sustained pool over over 100 virtual machines available for computation. Work to date has been focused on testing and proof-of-concept, with future work planned to make heavier computational use and provide additional functionality. Additional work may also include the automatic provisioning of virtual machines based on the real-time utilization of existing Condor pools.

1. INTRODUCTION

Condor is used on approximately 37,000 cores across the Purdue University West Lafayette and Calumet campuses in varying roles. For the central community clusters, Condor is used to scavenge cycles unused by the PBSPro scheduler. Smaller, special-purpose clusters use Condor as the primary scheduler. Instructional labs, administrative offices, and academic departments run Condor, primarily on Windows desktops, to provide scavenged cycles. All of the separate Condor pools flock to each other, allowing jobs to flow freely across campus to whatever resources are available. In addition, pools flock with outside institutions as part of the Dia Grid[2] project. This allows institutions to make excess capacity available to peers.

Windows slots are in idle (i.e. not being used by local users or Condor jobs) state more than 90% of the time, on average. Nearly one fifth of the cores in Purdue's Condor pools run the Microsoft Windows operating system. While there are some researchers who have code that will work on Windows, the majority of Condor jobs (99.5%) require Linux. The resulting under-utilization represents a significant opportunity to improve the campus grid resource, allowing the university to get the most computation out of the hardware.

Condor supports using various types of virtual machine systems, including VMWare, as compute jobs. Normally, the virtual machine has some self-contained computational payload, however it is also possible to run a base operating system with its own Condor instance as a VM universe job. In this way, a Windows machine can run a Linux VM that would then be available for computational use.

2. WORK PERFORMED

2.1 Network setup

The initial configuration was to create a CentOS 5 virtual machine that contained a minimal OS installation sufficient to run Condor and the IP over Peer-to-Peer (IPOP)[3] network stack. IPOP is a technology developed by the ACIS laboratory at the University of Florida in order to create robust, self-configuring virtual networks over an existing Peer-to-Peer (p2p) network. By using this method, the virtual networks can span existing firewalls and NATs that may otherwise cause issues routing traffic between the virtual compute nodes and physical Condor servers.

One significant change that was made as testing progressed was switching away from IPOP networking to using the Condor Connection Broker (CCB). CCB uses the Condor poolâĂŹs central manager to broker connections, allowing hosts on private networks to freely communicate with outside hosts. By using CCB instead of IPOP, the hosts can use

[1] http://www.cs.wisc.edu/condor

[2] http://www.dia-grid.org

[3] http://byron.acis.ufl.edu/papers/ipdps06ipop.pdf

the host's network and not depend on an additional IPOP server.

2.2 Job submission

Initially, VM images were submitted from a single host and the entire disk image (approximately 2-3 GB) had to be sent over the network at job start and eviction. Initial testing saw over 1000 VM jobs running concurrently. This lead to bottlenecks when many VM jobs were running in a building that only had a 2 Gb/s connection to the campus core. Students using the labs reported the machines were slow to log in when VM jobs were being evicted. At 2.5 GB, a building with 110 lab machines and a 2 Gb/s connection would take over 18 minutes to transfer the VM images at wire speed. This is approximately one-third of the standard class period at Purdue.

In order to improve the performance of VM jobs at start and eviction, the Condor developers added support for pre-staging VM images on execute hosts in Condor version 7.5.3. By pre-staging the VM on local disk, job startup and eviction no longer saturates the network connection. The bulk queueing of VM jobs is also faster since the scheduler does not have to copy the same image out to each execute node.

3. FUTURE WORK

As testing completes and the VM jobs enter production status, future work will focus on adding additional functionality. Since the VM images contain only a minimal install, jobs submitted to a VM must be completely self-contained and transfer all executables and data. Future implementations could have additional disk images containing various applications commonly used by researchers. By allowing the applications to be "bolted on" to the base disk image, updates can be done minimally, reducing the overhead when new images need to be deployed to physical hosts.

Other work relates to the size of the virtual machine pool. One possibility is the use of of pre-configured VMs at partner institutions. Similar work is being done as part of the Extending Science Through Enhanced National CyberInfrastructure (ExTENCI) project[4]. This would increase the overall size of the VM pool and reduce the impact of heavy use periods in instructional labs.

Additionally, it may make sense to dynamically adjust the size of the VM pool based on the utilization in existing physical pools. By only submitting VM images when other pools are near capacity, the Windows cores are left available for Windows-specific jobs or to hibernate. When pools have a low idle core count, VMs could be quickly spun up to meet demand.

[4]https://sites.google.com/site/extenci