KVM on Clusters: Tackling the Disk I/O Bottleneck for HPC Virtualization

Benjamin Eckart*, Xubin He
Department of Electrical and Computer Engineering
Tennessee Technological University, Cookeville, TN
{bdeckart21,hexb}@tntech.edu
* Graduate Student

Hong Ong, Stephen L. Scott
Computer Science and Mathematics Division
Oak Ridge National Laboratory, Oak Ridge, TN
{hongong,scottsl}@ornl.gov

Abstract

Extending virtualization technology into high-performance, cluster platforms generates exciting new possibilities. For example, runtime environments can be extended and tailored to each user or application, which in turn enables broader library support and customization as well as cross-cluster compatibility. Dynamic job consolidation on compute nodes can be used for more efficient hardware utilization. Virtual machines can be suspended to and resumed from disk and can even be migrated to other clusters, preserving complete states and configurations. Live migration of virtual machines can also augment the fault tolerant aspects of the system by moving applications away from failing nodes. *The question remains, however, whether the benefits of a virtualized cluster environment outweigh the potential hazards of additional overhead. This concern is especially critical for I/O-bound HPC applications, which have been notoriously difficult to make efficient within virtual machines.*

Much work has been investigated, with very promising results, in the area of CPU efficiency for use in virtualized environments. However, I/O efficiency in virtualized environments, specifically with respect to disk I/O, remains little understood and hardly tested. Our research goal is to analyze the properties of disk I/O in virtualized cluster environments in order to identify, study, and eventually minimize all possible performance bottlenecks. As part of our ongoing research, we have developed an extensible performance analysis framework for characterizing disk I/O workloads across virtualized clusters. This toolkit contains two components: *Controller* and *Mine*. The Controller component utilizes a modular design to provide a plugin framework for performance tools, to transparently deploy and synchronize monitoring tools across clusters, and to aggregate results under a common database scheme. The Mine component parses data from the database, based on XML templates, and interactively generates graphics and in-depth statistics. *Though it is still at the embryonic stage, our framework has been used successfully in many test cases concerning virtual disk I/O efficiency in relation to cache behavior, paravirtualized driver support, and CPU utilization statistics.*

This poster focuses on the disk I/O characteristics of the Kernel Virtual Machine (KVM). KVM has several desirable properties warranting further research into its feasibility as a virtualization solution for clusters. KVM was developed using processor-based support in a deliberate attempt to write a lightweight hypervisor. Unlike Xen and other popular virtualization frameworks, KVM is written as a module for the Linux kernel, making it relatively simple to deploy, as opposed to other hypervisors that require kernel recompilation and/or drastic changes to the underlying operating system. Using our framework within KVM, we are able to conduct extensive tests to determine an accurate characterization of disk I/O in relation to these factors: multicore, disk scheduling, paravirtualization vs. full-virtualization techniques, caching effects, and networked storage configurations. *Our poster will reflect our ongoing virtual I/O research, displaying the uses of this novel framework for the HPC virtualization community, and discuss preliminary analysis of our results on a small-scale cluster configuration.*