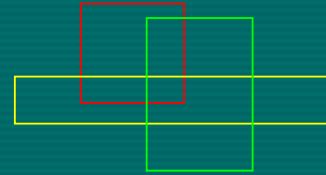


HPC Virtual Machine Resource Management

Wesley Emeneker and Amy Apon
University of Arkansas



Motivation

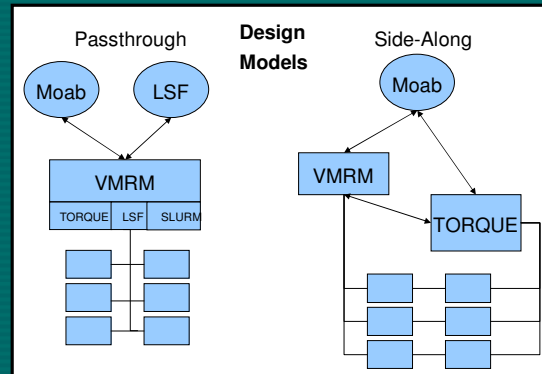
Intelligently using VMs has improved some multi-cluster's workload performance [1]. Transparently deploying and managing VMs in clusters is still difficult. There are many tools for deploying and managing VMs in data-center environments, but these tools do not meet HPC needs for VM Resource Management (VMRM).

To HPC needs, VMRM must:

- plug into the existing scheduling and job management infrastructure
- cannot significantly change the way users interact with the system
- provide new and desirable capabilities
- support both serial and parallel jobs

Planned Capabilities

- Transactional creation of hundreds or thousands of unique VMs from a single master image
- Transparent software environment switching
- Transparent parallel and serial checkpointing
- Hybrid Linux/Windows clusters
- Job migration to different physical nodes



VMRM Basics

- Implemented as a daemon that the scheduler (or other processes) can control
- Creates, destroys, and monitors VMs
- Migrates, checkpoints, and preempts VMs on demand

Progress

- XML is the selected communication protocol
- The VMRM acts as a daemon and accepts network messages
- Approximately 1500 lines of Python, including unit tests have been written
- Items that can be selected for each individual image include IP and hostname ranges, allowed users, memory and CPU limits

Comparison of Models

- Passthrough model
 - VMRM needs intimate knowledge of how to interact with job manager
 - Must rewrite jobs to use VMs created on real nodes
 - Transparent to scheduler
 - More complicated to implement and more housekeeping required to interact with the scheduler and job manager
- Side-Along model
 - Scheduler must be modified to interact with VMRM
 - Job manager might be modified to be aware of VMs
 - Simpler to implement, but requires scheduler modifications

Future Work

- Secure authentication and communication!
- Support for more VMs than Xen
- Image caching, migration, and checkpointing all have stubs, but are not yet implemented

Reference

[1] Wesley Emeneker and Daniel Stanzione, "Dynamic Virtual Clustering," Cluster 2007.