Evolution of OSG to support virtualization and multi-core applications

(Perspective of a Condor Guy)

Dan Bradley

University of Wisconsin

Workshop on adapting applications and computing services to multi-core and virtualization,
CERN, June 2009

VMs in OSG

- VM worker nodes
  - enthusiastic reports from site admins
  - transparent to users
- VM middleware pilots
  - just a more dynamic version of above
  - on-demand creation of worker nodes or services
  - still transparent to grid users
- VM user pilots/jobs
  - VO or user provides VM
  - at prototype stage

VM Worker Node Example Purdue: Linux in Windows

- Using GridAppliance
- Run Condor in Linux VM
  - consistent execution environment for jobs
  - VM is transparent to job
  - IPOP network overlay to span firewalls etc.
  - sandboxes the job
- Hope to incorporate as many as possible of the 27k computers on campus

VM middleware pilot: "Wispy" Science Gateway

- Purdue's VM cloud testbed built with Nimbus
- Used by TeraGrid Science Gateways program
  - Prototype for creating OSG clusters on TeraGrid resources
  - VM looks like an OSG worker node

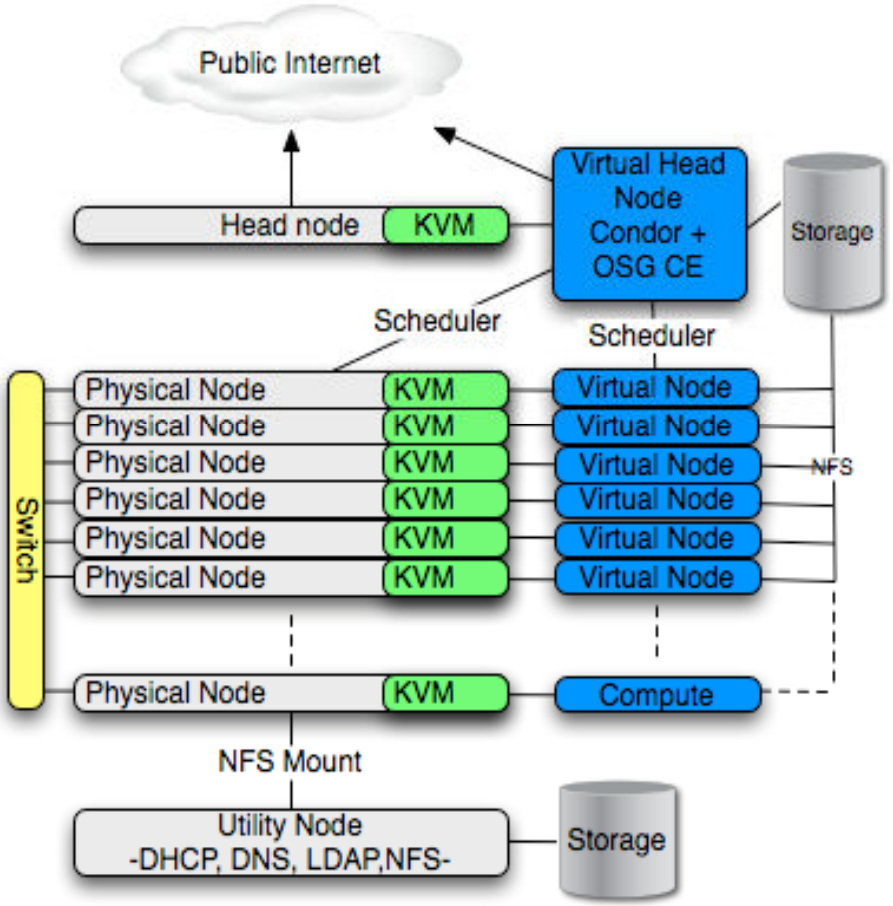
CLEMSON
SCHOOL OF COMPUTING

Clemson VO Cluster

- KVM virtual machines
- OSG CE in Virtual machines
- Condor as the LRM
- Condor within the VM image

- NFS share
- And PVFS setup

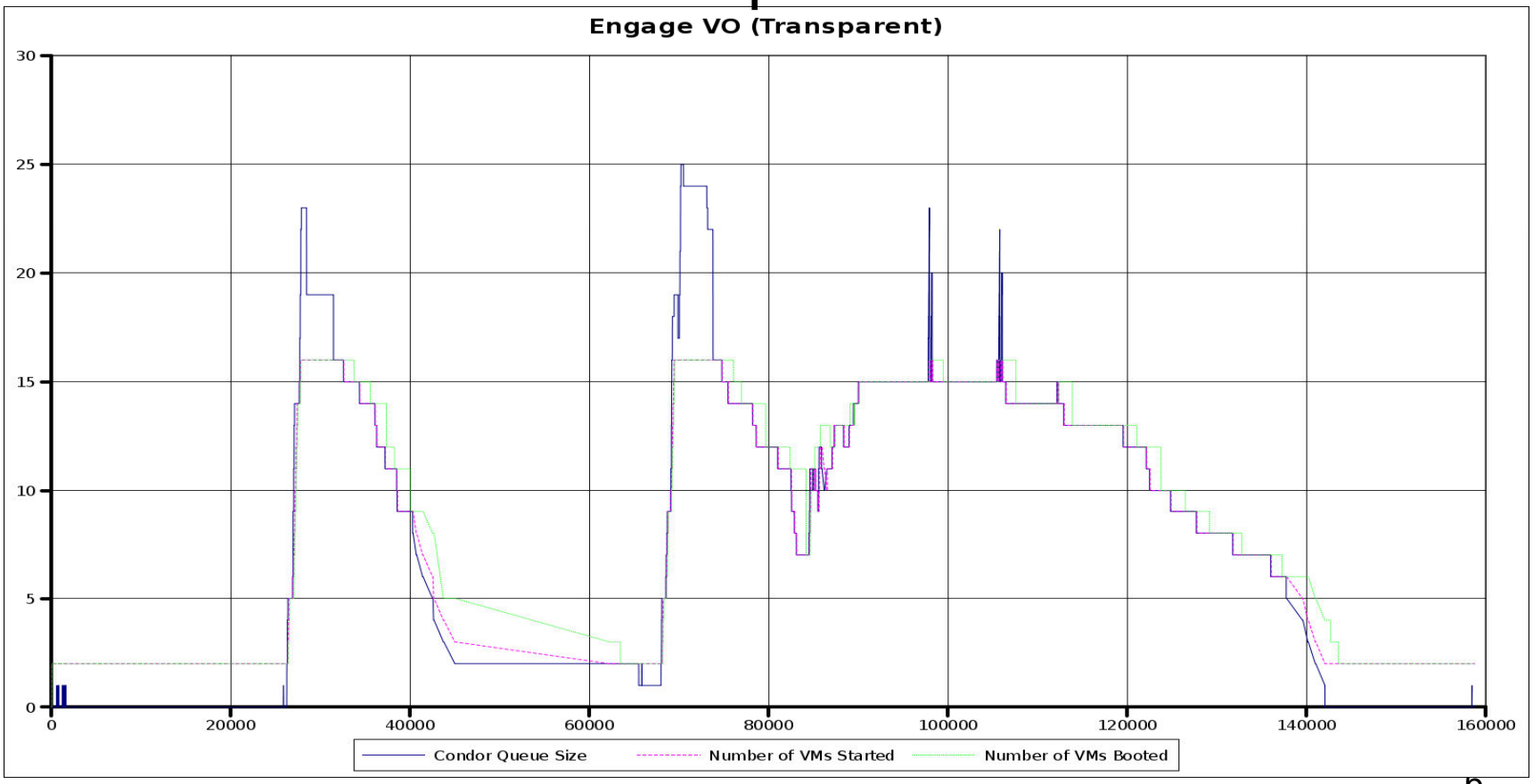
- KVM offers a snapshot mode that gives us ability to use a single image file. Writes are temporary



CLEMSON
SCHOOL OF COMPUTING

Results

- Engage VO
- Site Clemson-Birdnest on OSG Production
- Virtual cluster size responds to load



Routing excess grid jobs to the clouds

Examples:

- Clemson's "watchdog"
  - monitor job queue
  - run EC2 VMs as extra Condor worker nodes
- Condor JobRouter
  - with plugins, can transform "vanilla" job into VM or EC2 job
    - e.g. RedHat's MRG/EC2 approach
  - or operate more like watchdog, generating pilots
- reminds me of glideinWMS pilot scheduling
  - should be nearly directly applicable

VM Instantiation

- Nimbus, VMWare, Platform, RedHat, …
  - likely many solutions will be used

- Condor can run VMs using familiar batch system interface
  - start/stop/checkpoint VMs as if they are jobs
  - VMWare, Xen, KVM
  - can submit to EC2 interfaces (e.g. Nimbus)
  - working example:
    - MIT CSAIL VMWare jobs running on Wisconsin CHTC cluster

multi-core jobs in OSG

- MPI grid jobs are doable
  - tends to require site-specific tuning
  - doesn't directly extend to multi-core in all cases
- For multi-core, want standard way to request
  - whole CPU or whole machine
  - or N cores with M memory

… but currently, not standardized

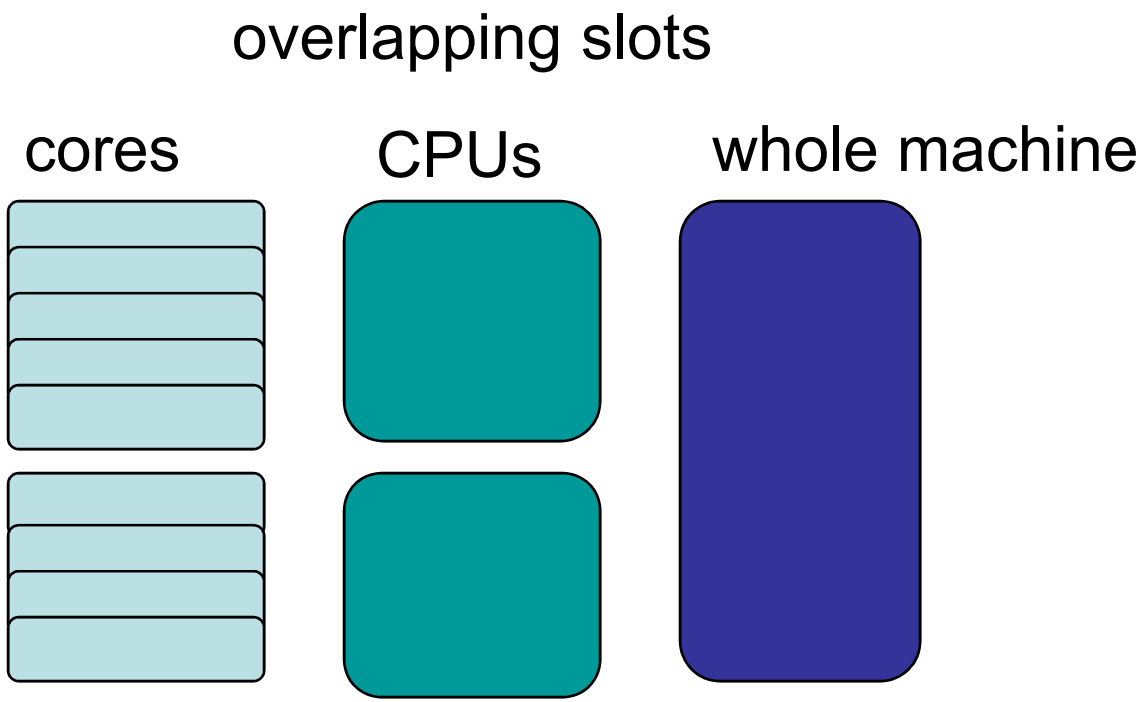
  - Example
    - PBS RSL: (jobtype=single)(xcount=N)(host_xcount=1)
    - Condor:
      - site-specific (custom ClassAd attributes)
      - not currently configured at most sites

Example of multi-core in OSG

- Wisconsin GLOW site
  - Several users require whole machine
  - workflow is many small MPI jobs
  - running on dual 4-core CPUs, 12 GB RAM
- Choices
  - statically configure one-slot machines
  - or mix whole-machine and single-core policies

Condor mixed-core config

- batch slot can represent whole machine or CPU or GPU or whatever
- slots can represent different slices of same physical resources



Condor mixed-core config

- Slot policy controls interaction between overlapping slots
  - can use job suspension to implement reservation
  - job can discover size of slot at runtime via environment
  - cpu affinity can be set
  - next development release improves accounting: weighted slots
  - example config from Condor How-to: http://nmi.cs.wisc.edu/node/1482

Condor variable-core

- dynamic slicing of machine into slots also possible
  - create new slot from leftovers after matching job
  - current issue: fragmentation
    - steady stream of small jobs can starve large jobs
    - doesn't reserve/preempt N small slots to create big slot
    - instead, must periodically drain machine to defragment

Summary

- VM worker nodes
  - already happening
- VM pilots/jobs
  - happening at resource-provider level
  - user-provided VMs more complex
    - need agreement of sites
    - need to deploy common interfaces
    - need logging of pilot activity (like glexec)
- multi-core jobs
  - grid interface needs improvement
  - need standard Condor configuration across sites