Grid deployment using virtual machines
MetaCenter use-case

Miroslav Ruda\textsuperscript{1,2} Jiri Denemark\textsuperscript{2,3}

\textsuperscript{1}Institute of Computer Science
Masaryk University

\textsuperscript{2}CESNET
Czech Republic

\textsuperscript{3}Faculty of Informatics
Masaryk University

Hamburg, 2007
MetaCenter

- academic grid infrastructure in Czech Republic
- consists of centers at different universities
  - Masaryk University in Brno
  - Charles University in Prague
  - West Bohemian University in Pilsen
  - and at CESNET
- hardware – around 750 CPUs
  - mostly Xeon/Opteron SMP clusters
  - SGI Altix servers
  - Opteron 16way servers
- dedicated network between sites
  - 10Gbps ethernet
  - DWDM – optical network
- participating in EGEE/EGEEII with another 250 CPUs
MetaCenter II

- software – production grid
  - shared filesystem – AFS
  - shared batch system – PBSPro
  - uniform environment – modules
  - common user management tools – Perun
  - integrated monitoring – Ganglia

- usual grid motivation
  - sharing resources
  - load balancing of jobs
  - redundancy and robustness
  - allow cooperation among scientists from different universities
  - allow experiments which exceed borders of one site
Virtual machines

- virtual machines can provide
  - several machines, with different OS or Linux flavor on the same machine
  - migration
  - suspend/resume

- could enhance MetaCenter (or general grid) in several ways
  - migration $\Rightarrow$ better scheduling, robustness
  - suspend/resume $\Rightarrow$ checkpointing
  - CPU/memory allocation $\Rightarrow$ interactive jobs
  - several virtual domains $\Rightarrow$ possibility to run different images for different groups, support different grid middleware
  - isolation $\Rightarrow$ provide illusion of dedicated cluster
virtual machines can provide

- several machines, with different OS or Linux flavor on the same machine
- migration
- suspend/resume

could enhance MetaCenter (or general grid) in several ways

- migration ⇒ better scheduling, robustness
- suspend/resume ⇒ checkpointing
- CPU/memory allocation ⇒ interactive jobs
- several virtual domains ⇒ possibility to run different images for different groups, support different grid middleware
- isolation ⇒ provide illusion of dedicated cluster
Virtual machines

- Virtual machines can provide
  - several machines, with different OS or Linux flavor on the same machine
  - migration
  - suspend/resume

- Could enhance MetaCenter (or general grid) in several ways
  - Migration ⇒ better scheduling, robustness
  - Suspend/resume ⇒ checkpointing
  - CPU/memory allocation ⇒ interactive jobs
  - Several virtual domains ⇒ possibility to run different images for different groups, support different grid middleware
  - Isolation ⇒ provide illusion of dedicated cluster
virtual machines can provide
- several machines, with different OS or Linux flavor on the same machine
- migration
- suspend/resume

could enhance MetaCenter (or general grid) in several ways
- migration ⇒ better scheduling, robustness
- suspend/resume ⇒ checkpointing
- CPU/memory allocation ⇒ interactive jobs
- several virtual domains ⇒ possibility to run different images for different groups, support different grid middleware
- isolation ⇒ provide illusion of dedicated cluster
Virtual machines

- virtual machines can provide
  - several machines, with different OS or Linux flavor on the same machine
  - migration
  - suspend/resume

- could enhance MetaCenter (or general grid) in several ways
  - migration ⇒ better scheduling, robustness
  - suspend/resume ⇒ checkpointing
  - CPU/memory allocation ⇒ interactive jobs
  - several virtual domains ⇒ possibility to run different images for different groups, support different grid middleware
  - isolation ⇒ provide illusion of dedicated cluster
Virtual machines

- virtual machines can provide
  - several machines, with different OS or Linux flavor on the same machine
  - migration
  - suspend/resume

- could enhance MetaCenter (or general grid) in several ways
  - migration \(\Rightarrow\) better scheduling, robustness
  - suspend/resume \(\Rightarrow\) checkpointing
  - CPU/memory allocation \(\Rightarrow\) interactive jobs
  - several virtual domains \(\Rightarrow\) possibility to run different images for different groups, support different grid middleware
  - isolation \(\Rightarrow\) provide illusion of dedicated cluster
1 Motivation
   • MetaCenter
   • Virtual machines

2 Current usage of virtual machines
   • Elementary usage
   • Service consolidation
   • EGEE/MetaCenter integration
   • Job preemption, interactive jobs

3 Future plans
   • Deployment issues
   • Short term plans
   • Long term plans
Studied use-cases

- portability tests, running services in different Linux distributions
- sharing of one machine by several services – service consolidation
- different Linux flavors running on the same worker node – EGEE/MetaCenter integration
- preemption, interactive jobs
Elementary usage

- running different Linux distributions on the same machine
  - environment for software development
  - for portability tests (EGEE LB service)
  - for simulation of distributed environment
  - some software may require specific Linux distribution
- usually first use-case, very useful to familiarize with virtual machines tools
- in our case Xen, Vserver and OpenVZ
Xen vs. Vserver

- **Xen – paravirtualization**
  - useful for complete encapsulation
  - support for complete Linux distributions
  - perfect solution for service consolidation
  - may not be necessary for worker nodes, but currently used for EGEE/MetaCenter integration

- **Vserver – one kernel space**
  - higher number of virtual machines with small overhead
  - useful when just one or few services must be running – perfect for development machine
  - may be better solution for preemptive use-case (two domains of the same flavor)
  - better on NUMA architecture

- **adoption curve similar, with slightly different problems**
  - Xen – kernel modules, AFS
  - Vserver – standard system daemons, INADDR_ANY binding, loopback
Xen performance results

- good results on small SMP machines – minimal delay for CPU, memory, disk intensive applications
- bad results for fast networks – one CPU is required for bridging on full speed gigaethernet
- bad NUMA support – on 16way Opteron machine slowdown from 5 to 13 minutes
- initial tests with the HVM not encouraging
Xen performance results

- good results on small SMP machines – minimal delay for CPU, memory, disk intensive applications
- bad results for fast networks – one CPU is required for bridging on full speed gigaethernet
- bad NUMA support – on 16way Opteron machine slowdown from 5 to 13 minutes
- initial tests with the HVM not encouraging
Xen performance results

- good results on small SMP machines – minimal delay for CPU, memory, disk intensive applications
- bad results for fast networks – one CPU is required for bridging on full speed gigaethernet
- bad NUMA support – on 16way Opteron machine slowdown from 5 to 13 minutes
- initial tests with the HVM not encouraging
Xen performance results

- good results on small SMP machines – minimal delay for CPU, memory, disk intensive applications
- bad results for fast networks – one CPU is required for bridging on full speed gigaethernet
- bad NUMA support – on 16way Opteron machine slowdown from 5 to 13 minutes
- initial tests with the HVM not encouraging
Xen overhead

- active use of memory
  - dom0
  - every running domU needs at least 100MB
- disk partitions dedicated to different VMs
  - not easy (read-only) sharing of root filesystems
  - required splitting of scratch partitions
- fast network can be dedicated to one domU or bridged
Service consolidation

- primary motivation – efficient use of hardware
  - EGEE in a box
  - 7 domains running all EGEE services in different VM (WMS, LB, Myproxy, VOMS, CE, WN . . . )
  - different EGEE service require different setup, packages, are not compatible
  - used for certification and pre-production testbed
  - but also for production WMS for the VOCE

- 2xXeon 3.0GHz (4 CPUs with HT), 6 GB RAM, 2x150GB disk

- Xen is perfect solution, overhead is minimal
  - all services running all the time, statical splitting of memory is OK
  - root filesystem is different for different domains
EGEE/MetaCenter integration

- primary motivation – allow coexistence of EGEE and MetaCenter environments
- two images running all the time – Debian/OpenSuse (MetaCenter) and SLC (EGEE)
- EGEE gateway (Computing Element) submits to standard PBS, but to special queue
- dynamic allocation of resources to EGEE and MetaCenter maintained by PBS
- PBS must be aware that two VMs share the same node, but with minimal changes on PBS side \(\Rightarrow\) Magrathea project
- no changes to EGEE software
integrating virtual machines and PBS
- each node can run several VMs at a time
- at most one VM on each node is active
- however, a VM can be activated even if another one is active – preemption
- active VM is provided with “all” physical memory and CPU power

implementation
- PBS cannot recognize real machines from virtual ones
- special PBS attribute to distinguish amongst free, running and occupied machines
- modified PBS scheduler schedules jobs to free machines only
- current state of VMs is maintained by a daemon running on each physical machine
Magrathea – implementation

Grid and virtual machines
Miroslav Ruda, Jiri Denemark

Motivation
MetaCenter
Virtual machines

Current usage of virtual machines
Elementary usage
Service consolidation
EGEE/MetaCenter integration
Job preemption, interactive jobs

Future plans
Deployment issues
Short term plans
Long term plans

Worker node

submission denied

PBS server

job submission

magrathea-master

magrathea-slave

VM
running

submission permitted

VM
magrathea-slave

high-priority

PBS MOM

free
Job preemption, interactive jobs

- primary motivation – adding support for interactive jobs to MetaCenter
  - new class of users who cannot use batch mode
  - new functionality for current users
- two Debian/OpenSuse images running all the time, second accessible only by privileged jobs
- when privileged job is coming, standard domain is suspended – not used now
  - node/job is down for PBS
  - problem with parallel jobs
- given only small fraction of CPU, small real memory
  - currently usable only for sequential jobs, support for parallel jobs will require migration and support on scheduler
1 Motivation
   - MetaCenter
   - Virtual machines

2 Current usage of virtual machines
   - Elementary usage
   - Service consolidation
   - EGEE/MetaCenter integration
   - Job preemption, interactive jobs

3 Future plans
   - Deployment issues
   - Short term plans
   - Long term plans
Deployment issues $\Rightarrow$ motivation for new research

- imagine, that number of your machines grow 5x
  - you will be out of public IP address $\Rightarrow$ IPv6 deployment, (private network, VPN)
  - any solution with scalability problems will become bottleneck
    - installation/management tools for clusters
    - monitoring
    - user management
  - you may find problem with licensed software

- image management $\Rightarrow$ Workspaces integration?
- Infiniband available only in one virtual machine $\Rightarrow$ ??
- security implications – separation of different domains, user supplied images
- monitoring/benchmarking
Short term future plans

- Magrathea extensions
  - more than two virtual domains
  - not all domains running
  - fine-grained resource allocation – virtual domains per job
- improved support for job preemption – parallel jobs
- more flexible EGEE/MetaCenter integration
- better integration with batch system – management of virtual machines
- minimization of overhead
  - Xen
    - memory
    - shared filesystem for several domains
  - shared scratch filesystem – PVFS2?
- Vserver and IPv6
Long term future plans

- efficient sharing of high speed interfaces
- monitoring
  - monitoring and management of hosting VM (dom0)
  - monitoring of services in user VMs, including their batch system
- scheduling support
  - scheduling using features provided by VMs – suspend, checkpointing, migration
  - hierarchy of schedulers is more complicated (meta, batch, workspace, VM, OS scheduler)
- migration
  - local filesystem
  - cooperation with scheduling
- model
  - two planes – real and virtual
  - dynamic mapping of virtual machines to real resources
Thank you for your attention.