Applications of Virtualization Techniques in the Grid Context

Yves Kemp
DESY IT

GridKa School 2007
What is virtualization?

- Definition from Enterprise Management Associates:
  - technique for **hiding physical characteristics** of computing resources from the way in which other systems, applications, or end users interact with those resources.
  - making a single physical resource appear to function as multiple logical resources
    - server, operating system, application, storage device
  - making multiple physical resources appear as a single logical resource.
    - storage devices or servers

- Grid Computing is about Virtualization of resources!
  - Hides physical characteristics by introducing a standardized layer of abstraction → Multiple resources appear as one single logical resource.

This talk is about OS virtualization and its applications in the Grid field
Overview

- Not a coherent talk
  → Collection of ideas about uses of virtualization:

  - Consolidation
    - Server, Grid Services
    - Computing centers
  - Testing & Deployment
  - Computing nodes
  - Tools
Consolidation

- Server consolidation via Virtualization established in the IT world
  - Services might have to run on separate OS instances
  - Leads to server sprawl
  - Virtualization saves space, energy, hardware costs, maintenance...
  - Virtualization enables higher QoS, new features:
    - Redundancy, security, migration, …
Examples:

- **University of Karlsruhe: EKP**
  - Small site: service nodes (CE, SE, MON) not under heavy load
  - One single powerful machine, with failsafe hardware hosts up to 8 service nodes
  - Using Xen

- **Experience: Good!**
  - Started with Grid Services, now virtualizing the other server infrastructure (ldap, print server…)
  - Two identical server, shared Distributed Raid Block Device enables live migration
  - More reliable hardware, OS deployment eased, admins can concentrate on other things

Büge et. al. eScience 06
Consolidation, examples contd.

- Grid-Ireland Setup:
  - Operations Centre at the Trinity College Dublin:
    - Provides top-level services (RB, LFC, VO...)
    - Provides and manages Grid-Gateways for 17 sites in Ireland
  - Local site admins only manage their Worker Nodes
  - All local grid services running in VMs (Xen) in one physical box
- Experiences:
  - Massive expansion of Grid sites
  - Custom testbeds for developers
  - Management tools needed!!!
Consolidation, examples contd.

- MetaCenter (Brno, Prague, Pilsen & CESNET)
  - EGEE in a box: 7 Xen domains running different services

- Example of not-yet-virtualized site: DESY-HH
  - Some production service nodes under heavy load: CE, SE components, …
  - Some services (RB) different independent boxes
  - Investigating possibility of “spreading one VM over multiple boxes”
Consolidating whole Clusters

- One cluster might be “too small” for one application
  - Aggregation of clusters
  - Dynamic re-partitioning of clusters
- Also formation of sub-clusters possible
- MPI over WAN cluster
WAN, Multi-Site MPI using XEN

- **General setup:**
  - Xen 3.0.2, Linux 2.6.16

- **Variations:**
  - Connection over LAN (Gbit)
  - WAN via PacketiX
  - WAN via OpenVPN

- **Results:**
  - Virtualization overhead: 0-20% on 4-128-node clusters
  - Overhead smallest when compute-intensive
  - Migration of VMs possible

- **Is this a model for federated Tier-Centres?**

---

Tatezono et.al XHPC06
Dynamic Virtual Clustering

- **Idea:** Use existing clusters and dynamically “reassemble” them for different applications
  - Using virtualization (Xen)
  - Provide always needed OS
  - VMs in correct network
  - Integrated in the batch system (Moab)
  - Capacity of Load Balancing over cluster boundaries

- **Implementation details:**
  - Batch server dynamically adds or removes VMs from Torque resource manager
  - VM image staged to local disk, started, and deleted after job execution
  - Modifications to the Moab scheduler (together with developers)

Emeneker et. al, XHPC06
Testing and Deployment

- Virtues of Virtualization:
  - Fast and flexible deployment of machines
    - Faster installation than physical machine installation through image management
  - Different OS flavors on one/few machines
  - Snapshots: Save state of a machine before major intervention, easy roll-back
  - Enables complex testing and deployment workflows
  - Always clean and predictable environment
  - Development for upcoming platforms (emulation)
dCache build service @ DESY

- **Purpose**
  - Unified build service for dCache and Desy code
    - No more builds on developers machines
  - Secure and up to date build environment
  - Automated test deployments Suite

- **Design**
  - CVS, busybox, apt-get, xen-image-manager.py
  - Modular and simple
  - Fast: Reinstall 45-90 sec.
  - Automatic regression tests possible

1) Publish CVS tag into RDBMS
2) Triggers installation
3,4) System updated, build dependencies installed
5) Build state recorded in DB
6,7) Packages made available

Owen.Synge@desy.de
vGrid: Virtualization in gLite certification

- Certification testbed
  - ~60 machines @ CERN plus several other sites
  - All gLite services present
  - Daily regression tests
  - Installation (rpm) and configuration of patches

- Problems
  - Simultaneous Certification of several patches can cause conflicts
  - Patches often fail at RPM install or configuration
  - Testing: Switch quickly between different versions

- Solution:
  - 10 SLC4 machines with Xen 3.0.1, LVM
  - 28 hostnames/IP numbers
  - Heavily in use since October 2006
  - SLC3/4 images, users install gLite services on them
  - No scheduler: Users decides where to install

- Management using SmartFrog

- vGrid Portal at Cern: http://vgrid.web.cern.ch/

Omer.Khalid@cern.ch
Virtualization on the Worker Nodes

- Surprising idea: Virtualization costs performance, but many benefits:
  - More OS types and flavors can be supported, also old OS on new hardware possible
  - Each job runs in his own OS instance, does not affect other jobs: security through encapsulation
  - Separation of local and grid environment/users
  - Desktop harvesting?
  - Each job might get a clean system at start: No trojans
  - Buy a general purpose cluster, and use it for many different purposes
  - Job migration and checkpointing: Interesting for MPI and very long jobs
  - Distributed administration: Local admin installs VMM, generic Virtual Machine provided by user or third party

- One of the key issues: Integration into a batch system!
At Karlsruhe University:

- All nodes have two OS running all the time
- The OS needed gets all CPU and RAM resources
- Sharing all resources

- Using Xen: No noticeable performance loss due to virtualization:
  - Around 3-4% loss for CMS software
- Even performance gain is possible:
  - AMS group could benefit from 64 bit, but 32 bit common agreement
  - Galprop runs 22% faster in a virtual 64-bit machine than on 32-bit native system!

→ A overall performance gain can be possible (at least no drastic performance losses)
Integration into Batch system

- Batch system must know about the partitioning of the nodes, and must steer resource allocation
- Torque/Maui running
- Ansatz: Do not change any line of code in existing products!
- Written additional daemon

Problem: Writing a second scheduler, concurrent to Maui

Büge, Kemp, Oberst et. al. XHPC 06
Magrathea

- Small change to PBSpro (scheduler)
- Additional daemon (Magrathea): running on each physical machine
- One VM/node active (all resources), others might start: preemption
- Using PBS attributes to distinguish free/running/occupied machines

J. Denemark et. al., CGW06 and Desy Workshop Jan.07
Changing Moab

- **Arizona State University with Cluster Resources**
  - ASU has different clusters: interconnect with private, high-bandwidth network
  - Dynamic Virtual Clustering:
    - Deploys VMs in a (multi-)cluster to execute jobs
    - Software stack put into VMs and used anywhere
    - Scheduler deploys VMs to run user jobs
  - Implementation:
    - Moab scheduler modified: create and control VMs
    - VMs created for each job, customized at boot
    - VM disk images in central location
    - Using Xen (also considered Vmware and UML)
  - Results: better job throughput

http://hpc.asu.edu/dvc/
Other followed this way:

- ASU changed Moab for their purpose
- First HEP site evaluation this solution:
  - Simon Fraser University (Canada)
  - Atlas (Grid) on WestGrid (local jobs) resources
- Atlas and local jobs on same hardware!
  - Different OS and software stack
- Three different jobs types:
  - Local MPI Jobs: in non-virtualized environment
  - Local serial jobs: XEN openSuse-10.2
  - Atlas jobs: XEN SLC4 with LCG middleware
- Software
  - Recent Torque version >= 2.0
  - Moab cluster manager version >=5.0
  - Modifications to LCG software
- Test suite:
  - Moab starts Xen: up to 4 VMs per 4-core host
  - Moab waits: VM starts, OS updated, torque client starts; then submit atlas job
  - Communication between Moab master and slave efficient and stable:

Chechelnitskiy, CHEP07
Others:

- **University of Marburg:**
  - Extension of SGE: XGE
    - Backfilling for short parallel jobs in cluster filled with serial jobs using Virtualization techniques
    - Tested and used on MARC cluster (VTDC06 workshop)

- **University of Lübeck (Bayer et al)**
  - Dynamically installing RunTime Environments
  - Combination with virtualization in early state
  - Used in the ARC community

- **Commercial uses like Amazon Elastic Compute Cloud EC2** (using VMware)
  - ...
Globus Virtual Workspaces

- Other focus:
  - Previous solutions hide virtualization from the user
  - Globus: User encapsulates his environment in a VM and deploys it on remote resources
    - Authorized clients can deploy and manage workspaces on-demand via the GT4 Virtual Workspace Service
    - Currently using Xen

- Very promising techniques as very tight integration into the Middleware
  - Enables a real world-wide running of the same OS
  - The local admins do not have to care about users OS

- Has yet to be tested on large scale (Proof-of-concept comprises 5 nodes)

http://workspace.globus.org/
Administrative tools

- Management of VMs often an issue
- Many tools have emerged
  - Creation of VMs
  - Starting/Pausing/Stopping one/many VMs
  - Managing complete virtual clusters
- Solutions like VMware have some build-in tools
- XEN only provides basic management tools
  - Need to tailor own management tools
Example of a light-weight tool:

- xen-image-manager.py
  - Developed by Owen Synge for his purposes at Desy
- Small and simple python script
- Manages configuration of Xen domains
- Manages snapshotting of domains
- Scriptable Virtualization abstraction
  - Hide Virtualization implementation
  - Could be extended to work with other techniques
- Presents available hosts and images

http://trac.dcache.org/trac.cgi/wiki/manuals/xen-image-manager.py
Grid-Irelands Virtualisation tools

- **GridBuilder**
  - For interactive use
  - Manage VMs config
  - VM creation from templates
  - Web front-end
    http://gridbuilder.sourceforge.net

- **Quattor and Xen**
  - Quattor fabric management suite for OS installation and management

- **Xen support**
  - Describe state of VM host
  - Install VM guest automatically
  - Each service managed by components: Ncm-xen

- **Network bootloader for para-virtualized Xen-VMs**
  - Pypxeboot allows PXE installation of VMs

Childs et. al.
Summary and outlook

- Lots of topics not mentioned
  - KVM (Kernel-based Virtual Machine): Interesting to follow
  - Commercial deals around Xen: XenSource & Citrix, …

- Future of Virtualization in Grid
  - Many theory and proof-of-principle papers
  - Now we need mass-deployment in production systems

- My own appreciation:
  - Virtualization already solved many problems: Consolidation,…
  - Virtualization of Worker Nodes will solve many open CPU and security issues in Grid Computing. Soon!
  - Time to move focus from “CPU virtualization” to “storage virtualization”?

Thanks to all contributors and especially Owen Synge!