Virtual Machines and Grid Computing

Borja Sotomayor
University of Chicago (Dept. of CS)
borja@cs.uchicago.edu

Index

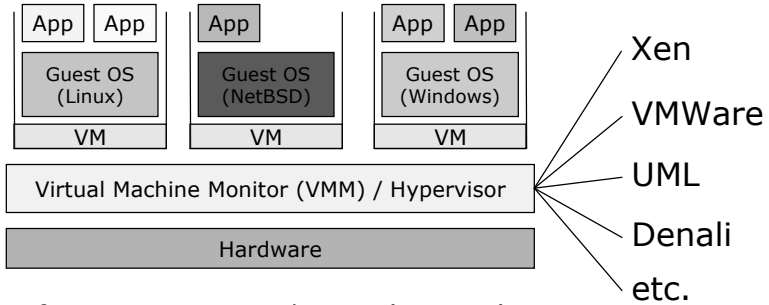
- Virtual Machines and Grid Computing
- Virtual Workspaces
- GT4 Workspace Service

Index

- Virtual Machines and Grid Computing
- Virtual Workspaces
- GT4 Workspace Service

Virtual Machines

- VM technology allows multiple virtual machines to run on a single physical machine.



Performance: Paravirtualization (e.g. Xen) is very close to raw physical performance

Virtual Machines and Grid Computing

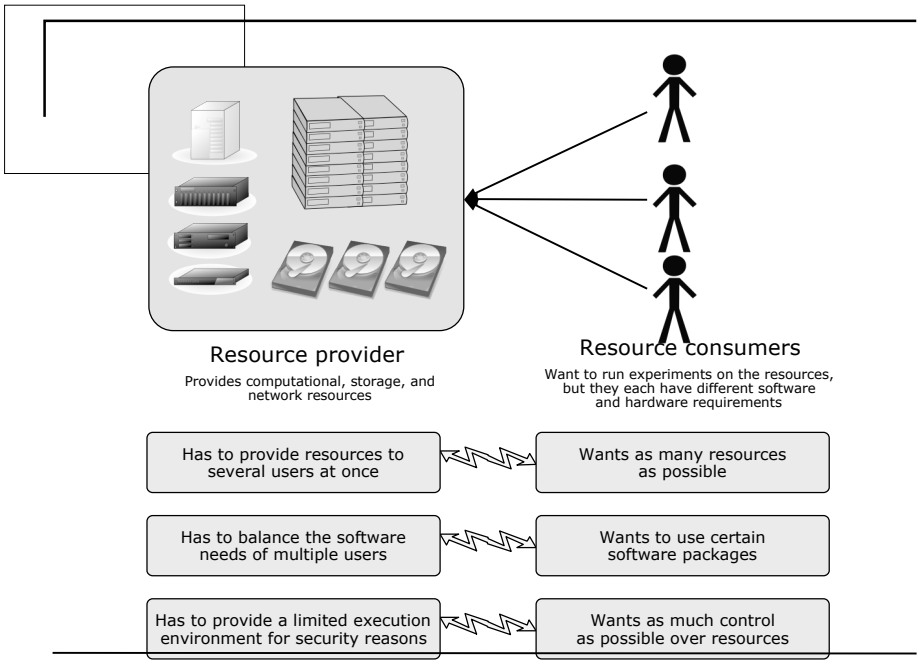
- VMs have several qualities that make them an appealing technology in Grid systems:
  - Security and isolation
  - Customization of execution environment
  - Resource control
  - Site independence
- Grid sites could perform their work on dynamically deployed VMs.

Index

- Virtual Machines and Grid Computing
- Virtual Workspaces
- GT4 Workspace Service

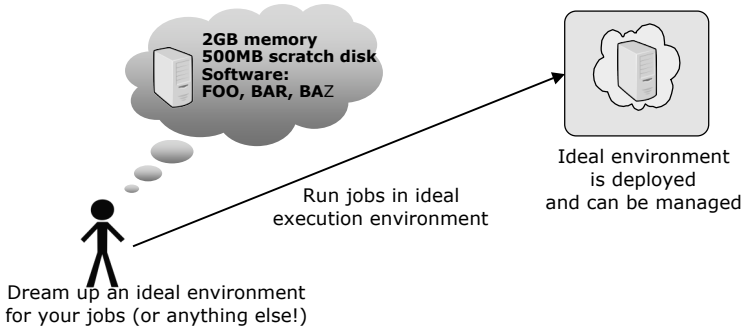
What is a workspace?

- What do I do if I want to "run something in the Grid"?
  - We have to encapsulate our work in a job abstraction.
  - This abstraction imposes a lot of restrictions on the resource consumer.
    - For example: Resource consumers can specify what libraries their jobs require, but they (usually) can't have those libraries dynamically installed when the job is submitted. If the library is not already available in the resources, they generally have to make a prior arrangement with resource providers to make sure that software is installed.
  - However, these restrictions are usually perfectly acceptable.
  - But... what if they're not?



What is a workspace?

- In some cases, resource consumers need *execution environments to be dynamically deployed on remote resources.*
- *The virtual workspace is an abstraction for such an execution environment.*

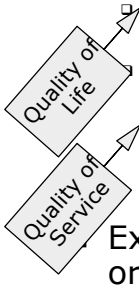


What is a workspace?

- A virtual workspace is an execution environment that can be deployed dynamically and securely on the Grid.
  - *Dynamic*: Started and stopped on an on-demand basis.
  - *Secure: Environment encapsulated inside the workspace and the user submitting the workspace are both trustworthy.*
- Of course, this is not an entirely new idea... It is possible to create custom execution environments by:
  - Dynamically setting up cluster nodes:
    - CoD: http://www.cs.duke.edu/nicl/cod/
    - bcfg: http://trac.mcs.anl.gov/projects/bcfg2/
  - Providing access to existing installation
    - Dynamic Accounts: http://workspace.globus.org/da/
  - Refining site configuration
    - Pacman: http://www.archlinux.org/pacman/

What is a workspace?

- Main two aspects of workspaces:
  - *Environment definition*: Users get exactly the (software) environment they need.
  - *Resource allocation*: Provision and guarantee all the resources the workspace needs to function correctly (CPU, memory, disk, bandwidth, availability), allowing for dynamic renegotiation to reflect changing requirements and conditions



- Existing implementations either don't provide both, or...
  - *Quality of life*: Setting up a new software environment takes a long time, and still doesn't give the resource consumer *full control.*
  - *Quality of service*: Little or no enforcement.

What is a workspace?

What to do?

Virtual Machines are a promising vehicle to achieve higher quality workspaces

Use cases

- Use cases that drive our work on virtual workspaces:
  - Virtual labs
  - Event-driven applications
  - Batch jobs with strict software requirements

Virtual labs

- A CS department wants to teach a course on Parallel Programming.
- Unfortunately, they do not have a computing cluster the students can use. And, even if they do, the cluster admin won't give the students root access.
- Solution: Deploy a virtual workspace that provides the students with a ready-to-use cluster.

Event-driven applications

- Some applications need large amounts of computational power the instant a specific event arrives.
- Science comes at you fast!
- Current solutions include job preemption or allowing urgent requests to skip to the front of the queue.
- Solution: VMs allow seamless suspend/resume of execution environments. "Urgent environment" can be placed on standby, activated when the event arrives, and all other VMs can be paused.

Strict software requirements

- Resource consumers can have strict software requirements:
  - Jobs that will only run with a specific version of a library.
  - Legacy software.
  - Special access privileges
- Managing multiple software requirements can be problematic for the resource provider
  - Conflicting software requirements.
  - Switching between software environments can be costly (e.g., reimaging cluster nodes)
  - Avoid malicious use of resources.
- Solution: Encapsulate desired software environment inside a VM.

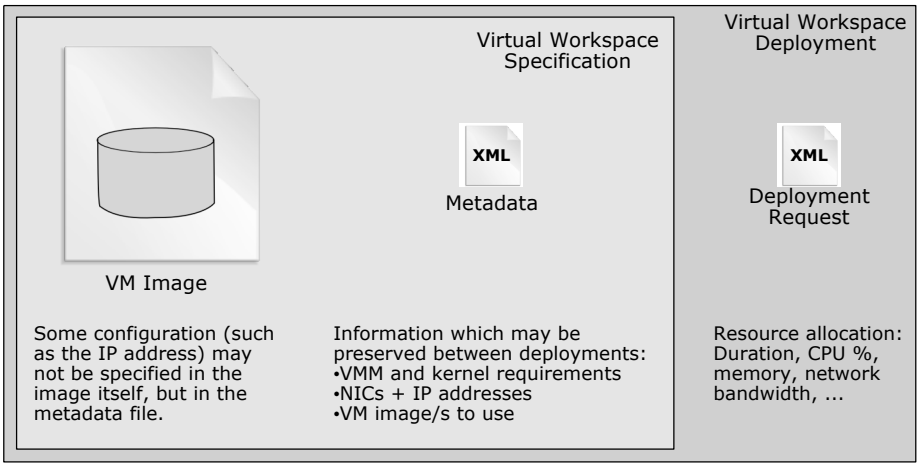
Index

- Virtual Machines and Grid Computing
- Virtual Workspaces
- GT4 Workspace Service

GT4 Workspace Service

- The GT4 Virtual Workspace Service (VWS) is a VM-based workspace implementation.
  - GT4 WSRF frontend
    - WSRF: Web Services Resource Framework
    - Collections of specifications (maintained by OASIS) for *state management with web services ("stateful" web services)*
  - Xen-based, but other VMMs could potentially be used.
  - http://workspace.globus.org/

Virtual Workspace
Specification and Deployment



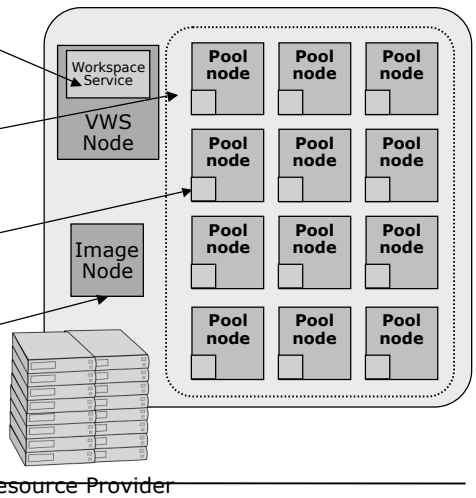
GT4 Workspace Service
Architecture

The workspace service has a WSRF frontend that allows users to deploy and manage virtual workspaces

The VWS manages a set of nodes (typically a cluster). This is called the *node pool*.

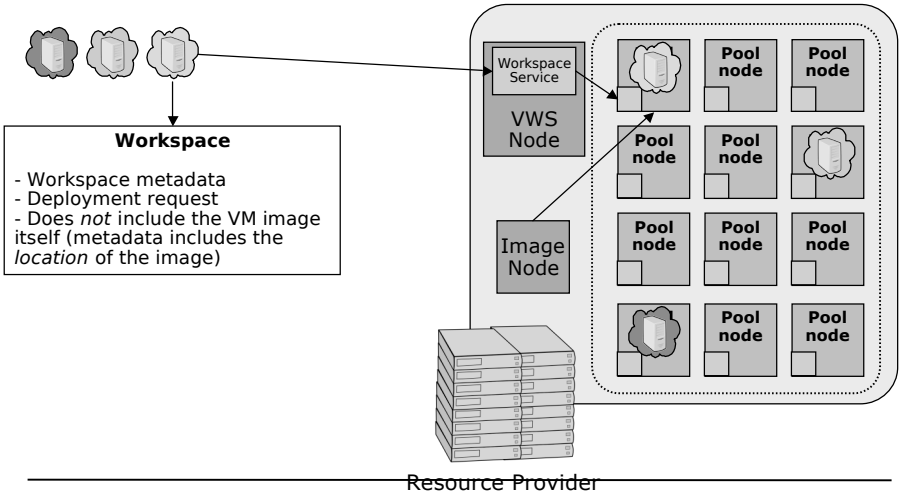
Each node must have a VMM (Xen) installed, along with the *workspace backend* (software that manages individual nodes)

VM images are stored in a separate node.



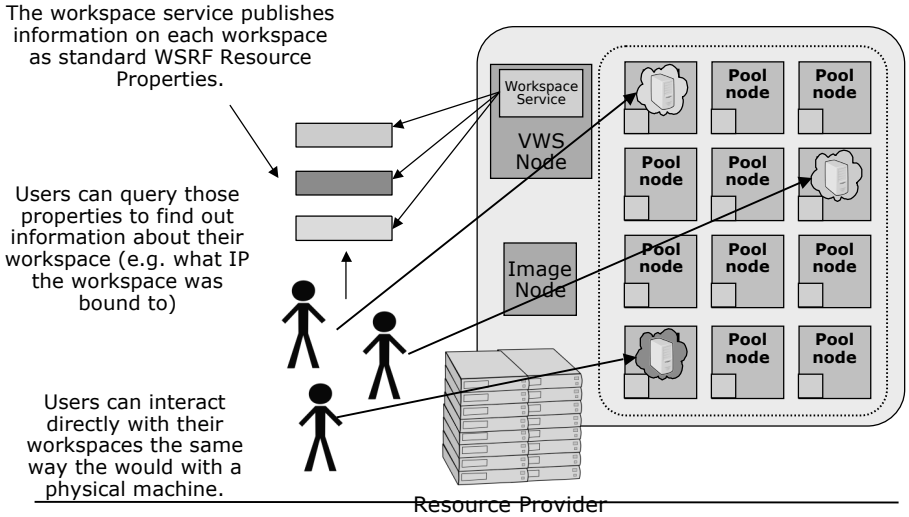
GT4 Workspace Service

Typical Deployment (1/2)



GT4 Workspace Service

Typical Deployment (2/2)



GT4 Workspace Service

- Workspace Service is a Globus *incubator project.*
  - http://workspace.globus.org/
- Current release TP1.2.2 (01/04/07)
  - Provides the functionality described in the previous slides (managing a pool of nodes, and deploying single-machine workspaces to those nodes)
  - We are currently working on supporting virtual clusters.

Questions?

Borja Sotomayor
University of Chicago (Dept. of CS)
borja@cs.uchicago.edu