Virtual Workspace in the Grid

Katarzyna Keahey, Ian Foster,
Timothy Freeman, Xuehai Zhang,
and Daniel Galron

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.
  - The abstraction imposes a lot of restrictions on the resource consumer.
  - However, these restrictions are usually perfectly acceptable.
  - But…what if they are not?

An Example of Submit Description File

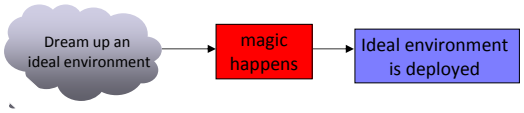
- Executable = foo
- Universe = standard
- Requirements = Memory >= 32 && OpSys == "Linux" && Arch ==“Intel"
- Error = err.$(Process)
- Input = in.$(Process)
- Output = out.$(Process)
- Log = foo.log
- Queue 150

The Missing Link in Grid Computing

- We need to define mechanisms for and *dynamic* deployment and management of remote environments
- Requirements:
  - Flexibly define an environment
    - The more we can customize it, the more useful it is
  - Deploy and manage such environments
    - Can such environments be deployed securely?
    - How fast/dynamic can this deployment be?
    - How can I control resources allocated to such an environment?

Virtual Workspaces

- Virtual Workspaces: environments that can be made available dynamically the Grid with well-understood properties



Workspace Template Aspects

- Environment Aspect (workspace meta-data)
  - Generic information
    - Name, time to live, etc.
  - Software partition information
    - Software description: OS, “OSG configuration”, “application partition”, etc.
    - Software meta-data is bundled with the actual software and attested by its issuer
  - Services: ssh, GRAM, pre-configured job
  - Deployment independent
- Resource allocation request (deployment time)
  - Memory, disk, networking, etc.
    - *See GGF JSDL standard*
  - On deployment the actual resource allocation information becomes available

Virtual Machines (VMs)

- VMs happen to have:
  - Good isolation properties
    - Generally enhanced security, audit forensics
  - Good enforcement potential
  - Customizable software configuration
    - Library signature, OS, maybe even 64/32-bit architectures
  - Serialization property
    - VM images (include RAM), can be copied
  - The ability to pause and resume computations
    - Allow migration

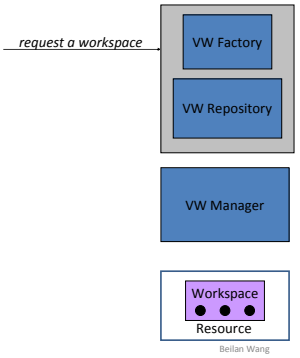
Atomic Workspaces and Virtual Clusters

- Atomic workspace
  - One or more homogeneous workspaces
    - The only differences are in names
- Cluster/aggregate workspace
  - A set of interdependent heterogeneous workspaces
    - Example: a head-node and a set of worker nodes
  - Interdependencies of metadata are expressed through tags and pointers

Supporting Services

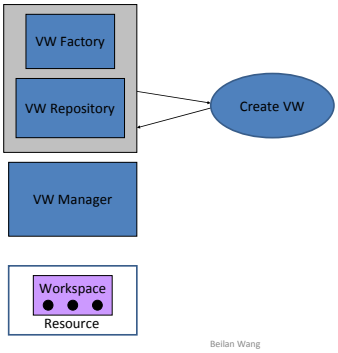
- VW Factory
  - Creates VW
    - A negotiation process may take place
- VW Repository
  - Access to state describing a VW
  - Allows inspection, management, termination, renegotiation and etc.
- VW Manager
  - Service deploying VWs on nodes
  - Operations: stage, deploy/undeploy, start/stop, pause/unpause, checkout VM images
- Once VW is deployed, jobs may be executed

Deploying Workspaces in the Grid



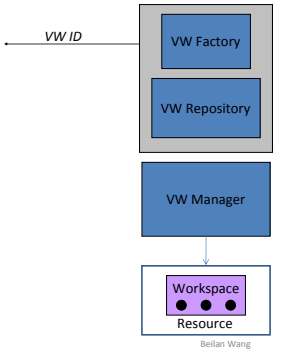
- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

Deploying Workspaces in the Grid



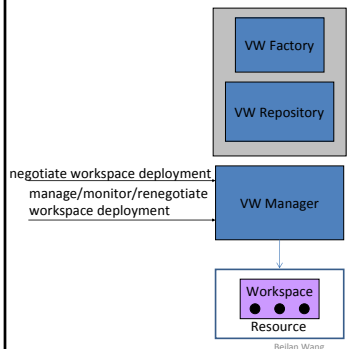
- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

Deploying Workspaces in the Grid



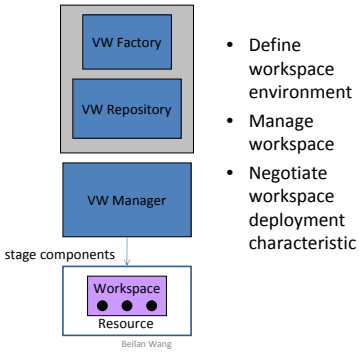
- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

Deploying Workspaces in the Grid



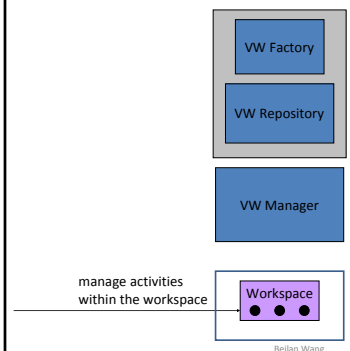
- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

Deploying Workspaces in the Grid



- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

Deploying Workspaces in the Grid



- Define workspace environment
- Manage workspace
- Negotiate workspace deployment characteristic

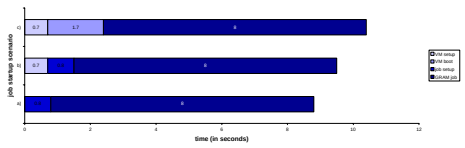
Workspace Structure

- Elements of a workspace
  - Workspace description (meta-data)
    - EPR/name, category, state, etc., also hardware, network, software configuration, etc.
  - Workspace implementation (VM image)
- Workspace instances
  - Workspace meta-data contains a name
  - Instance equality
  - Copying operation: copy image and meta-data, create a new name

Current Implementation

- Current prototype using Globus Toolkit 4
  - Leveraging standard Grid Service features such as lifetime management
- Workspace Wizard (VW Factory)
  - Returns workspace meta-data
  - Very rudimentary implementation
- Workspace Service (VW Repository & VW Manager)
  - Create: takes workspace meta-data and a deployment descriptor
  - Manage:
    - renegotiate resource allocation if necessary
    - Also traditional Grid Service management: TTL, etc.
  - Destroy
    - Different options: pause, shutdown or destroy

Workspace Service: Individual Workspaces



a) GRAM job execution
b) GRAM job execution in a paused Xen VM
c) job execution in a booted Xen VM (pre-configured job)

- Using a paused VM allowed us to "save" on initiation time

Things Not Talked About

- Security
  - Processing of encryption and signing
- Moving images
  - Image size: starting at 1MB, more typically 200 MB and upwards
  - Image diff (Rosenblum)
  - Proximity of image as matching criterion (VMPlant)
  - Mounting partitions (general on-site assembly)
- Scalability
  - Distribute processes among existing VMs
  - Lightweight VMs: Denali
- Clusters
  - Currently in progress: work on virtual cluster, collaboration with the COD team at Duke

Conclusions

- We need mechanisms for *dynamically* deploying and managing environments in the Grid
- Workspaces are a fundamental building block of a Grid environment
  - Workspaces are implemented using wide variety of technologies
    - VMs are a highly promising one: a "computon" for the Grid
  - Workspace aspects
    - Deployment-independent environment definition
    - Deployment-time policy and enforcement negotiation
- Many challenges remain
  - Security and deployment issues
  - Protocols, protocols, protocols
  - Leveraging the opportunities

Questions?