Two Approaches for the Management of Virtual Machines on Grid Infrastructures

Antonio Juan Rubio Montero
antonio.rubio@ciemat.es

Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas.
Avda. Complutense 22, 28040 Madrid, Spain.

D. Tapiador
daniel.Tapiador@sciops.esa.int

European Space Astronomy Centre/ESA
Villanueva de la Cañada, 28629 Madrid, Spain.

E. Huedo
ehuedo@fdi.ucm.es
R. S. Montero
rubensm@dacya.ucm.es
I. M. Llorente
llorente@dacya.ucm.es

Facultad de Informática
Universidad Complutense de Madrid
28040 Madrid, Spain.

Problems to Deploy Science Software in Grids

Common characteristics of science analysis software

- Frequently released → imposes new configurations
- Never released → bounded to old platforms
- Developed for a unique hardware/software architecture
- Should be deployed in all the Grid resources.
- Should require resource appropriation.

They Difficult (growing costs)

- Software development and testing.
- Portability.
- Support for several platforms (OS, hardware).
- Backward compatibility.
- Distribution of patches and new versions.

Objectives

Provide necessary isolation to:

- execute binary distributions of scientific software **without modification** on several platforms and architectures.
- Performance partitioning: executions of a user does not affect others.
- Free control of assigned hardware resources by the user.
- Free system configuration by the user.
- Reutilization of configurations.

No intrusive with actual Grid Infrastructures:

- Will **not require additional grid middleware** to be installed**.**
- Will grant **compatibility** among production Grid Infrastructures**: EGEE, TeraGrid…**

Provide necessary QoS to:

- get dynamically resources on demand for a time period.
- Possibility of interactive control
- Utility computing.

Our Solution: GridWay and Virtual Machines

Virtual machines provide

- Abstraction from the hardware of a computer.
- Isolation to run “unmodified” **OSs** with its own configurations in a single computer.
- Eventual use of resources.
- Server consolidation.

GridWay is a grid meta-scheduler that allows

- Unattended, reliable and efficient execution of jobs in heterogeneous Grids.
- Coordinated use of resources from several Infrastructures based in Globus, like EGEE, TeraGrid, IrisGrid….
- Adaptive job scheduling.
- User-friendly interface (CLI and DRMAA API).

Scientific Software can be installed in a VM and executed on the Grid by deploying the VM OS image, taking advantage of all virtualization and GridWay features.

First Approach: Straightforward Deployment

GridWay performs this deployment in phases:

- Scheduling.
- Preparation (***Prolog*** and ***pre-Wrapper***).
- Management (***Wrapper***).
- Finalization (***Epilog***).

GridWay performs the scheduling:

Holding a dynamic host list:

- The characteristics and state of the grid resources.
- Updated by querying grid information systems (MDS2 or MDS4)

The host list is filtered and sorted according to

- VM requirements in a template.
- User-supplied rank expression

First Approach: Preparation Tasks

Prolog phase

Makes a remote experiment directory in the front-end cluster node.

Sends all input data necessary for the analysis.

Can send VM images.

Uses GridFTP or RFT.

pre-Wrapper phase (optional)

It's executed in the front-end node before the *Wrapper* phase.

Used to create special configurations

- e. g. to get VM images from repositories localized by RLS
- Check consistence previously downloaded images

First Approach: Management and Finalization

The *Wrapper* program performs all necessary interactions with VMs in a worker node. Its execution is managed by GridWay through pre-WS or WS GRAM.

***Wrapper* steps:**

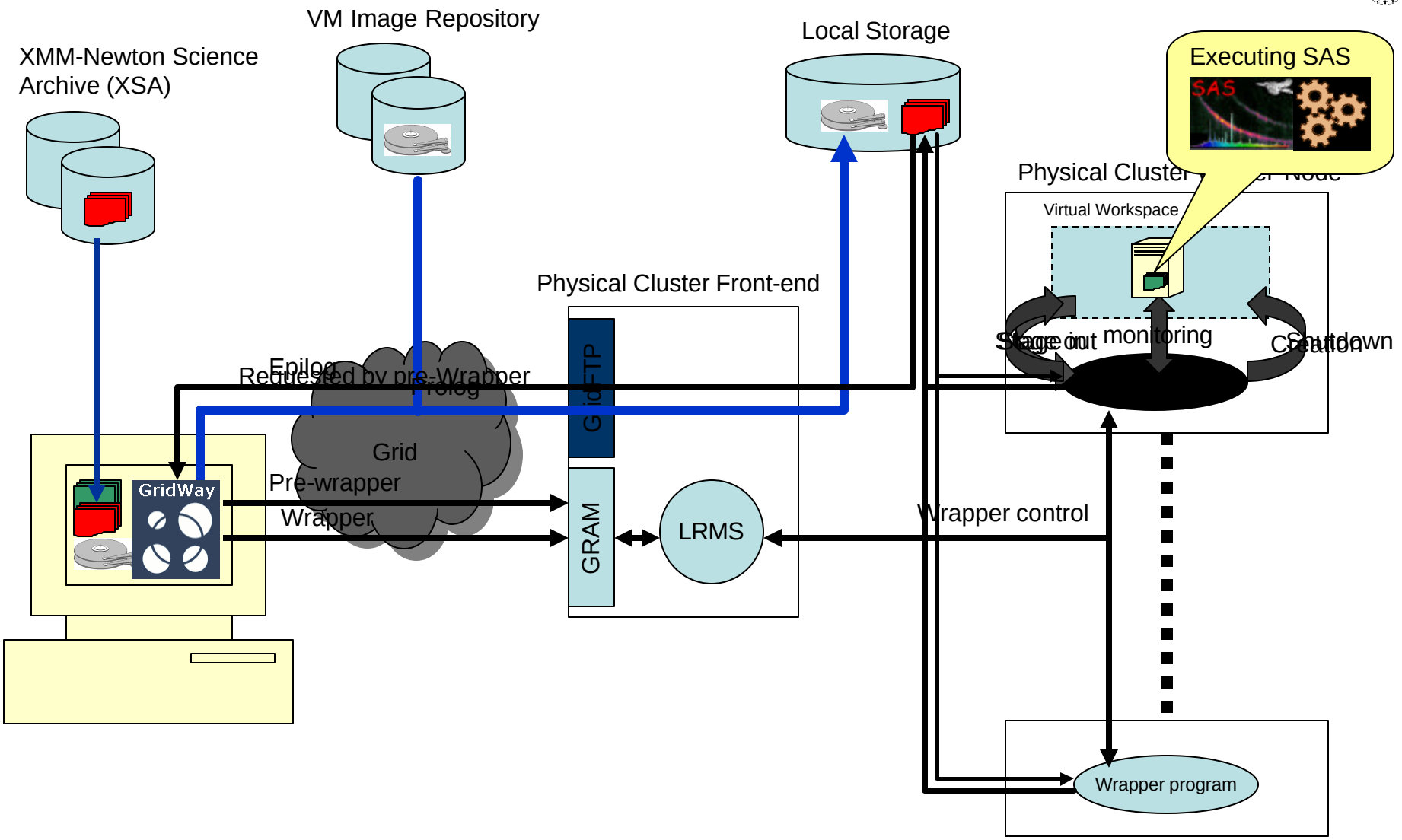
1) Checks the availability and configuration of VM images.
2) Boots or restores the VM from a context file.
3) Waits for the VM activation by testing its services.
4) Copies all input data for the experiment into the VM if needed.
5) Executes the scientific application.
6) Copies output files to the physical cluster file system.
7) Shuts down, pauses or suspends to a context file the VM.

***Epilog* phase**

Transfers back output data of experiment to the user by GridFTP.

Removes the experiment directory.

The Straightforward Mechanism in Action



First Approach: Some Sample Results

We have tested this mechanism with

A HTC application: XMM-Newton SAS 6.5.0

A research cluster:

- OpenPBS as LRMS,
- NFSv3, 100Mbps: note this storage model penalizes the results.
- PIV 3.2Ghz, 2GB RAM, Worker nodes with Xen 3.0 testing (2.6.12 kernel).

We have obtained

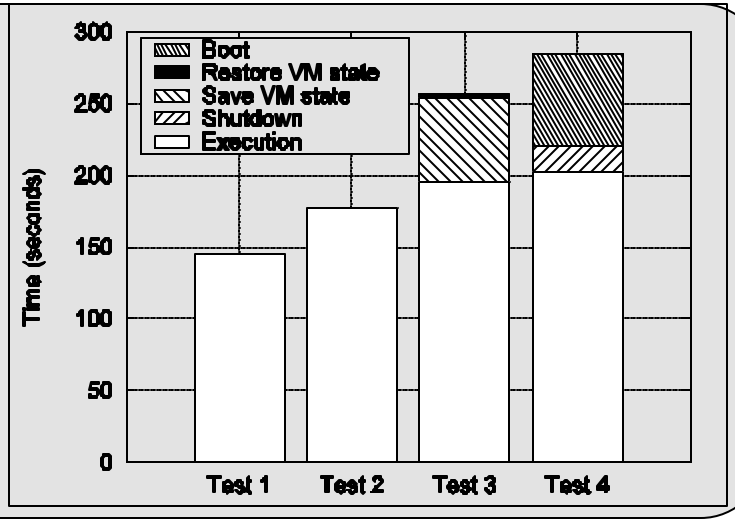
20% overhead in SAS analysis time

64 seconds in boot time

18 seconds in shutdown time

3 seconds in restore from a context file

59 seconds in save context to a file.



Overhead does not change with the size of SAS task to process.

A long analysis (hours) only will be affected by the 20% overhead.

Straightforward Approach: Conclusions

This solution

- Provides a straightforward method of software deployment on production Grid Infrastructures like **EGEE**.
- Scientific **software must not be ported** to several platforms.
- Does **not require** the **installation** of the **scientific software in remote resources**.
- Inherits the isolation and security from virtual machines.
- Does **not require additional Grid middleware** to be installed.
- Potentially compatible with other virtualization tools as VMWare, UML...

Drawback: Single use per deployment**.**

→ Only a group jobs can be submitted join the request of a single VM.

→ **Only Suitable for HTC** applications (as XMM-Newton SAS).

→ It cannot be used for **server consolidation**.

→ Deployment for interactive use is not supported (workshop, temporary research laboratory).

Second Approach: Deployment based on VWS

Virtual Workspace concept

- Abstraction of a whole execution environment.
- A single or many VMs and its virtual network connections simulating a small datacenter.
- Created and destroyed on demand.
- VW administrators are independent of real datacenter administrators or from other VW.

The Globus Virtual Workspace Service

- Manages virtual workspaces in a pool of Xen hosts.
- Remote client can securely negotiate and manage a virtual resource allocation (memory, number of CPUs, time required).
- WSRF compliant.
- Searches compatibility with EGEE by means of OSG Edge Services project.

The aim is to submit jobs to the workspaces deployed by VWS as if they were a physical machine bounded to a LRMS.

Layout of the VWS-based Approach

Whole scheme comprises three parts

- Grid Resource: Middleware services (GRAM, MDS, VWS, etc) and the LRMS.
- GridWay Meta-scheduler (no modifications needed).
- The workspace module (workspace manager and driver).

The Workspace Driver

Manages a VW interfacing Virtual Workspace Service:

- Requests a amount of resources for a limited time period (memory).
- Monitors VW execution (running, corrupted, stopped…).
- Performs common operations on virtual machines (boot, pause, resume…).

Can handle several VWs at the same time.

The Workspace Manager

Interacts with the driver and **translates** the requests coming from the user interface.

Takes info from GridWay for its own scheduling module.

Uses transfer drivers (if needed) to upload VM images.

VWS-based Approach: Loop of the job execution

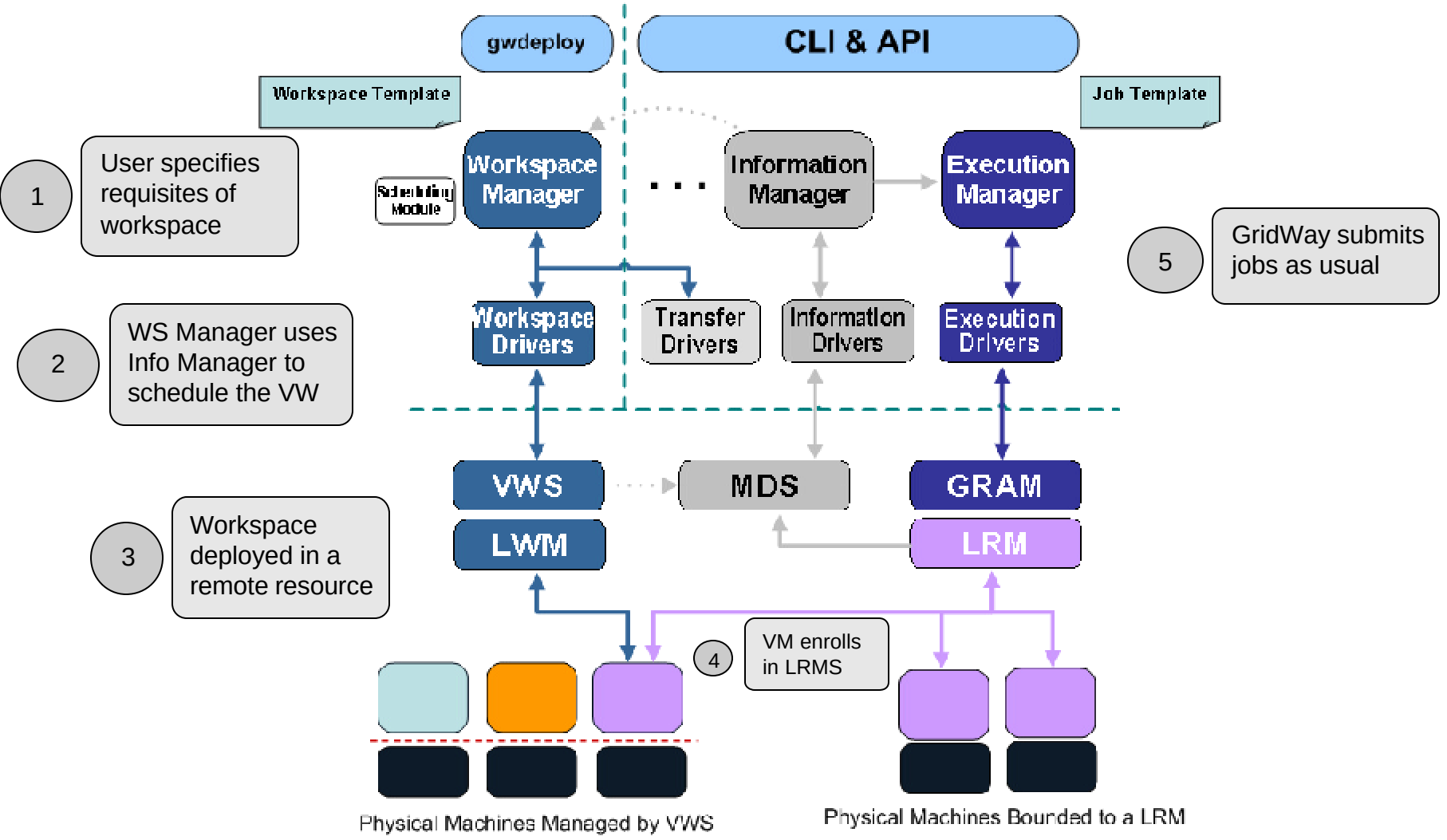
Virtual workspace ←→ GridWay connection

- The virtual machine image is pre-configured to enroll into the LRMS at boot time.
- MDS publishes a new slot in the corresponding queue.
- WS Manager controls VWS state (corrupted, stopped by the site administrator…) through WS Driver.

Complete loop of a job execution:

- The user specifies the features of the workspace.
- The Workspace Manager translates this request and schedules to the best resource (by means of the Workspace Driver).
- The virtual machine enrolls in the LRMS.
- The LRMS publishes this new queue/node.
- GridWay realizes this recently enabled node/queue and submits jobs to it.

VWS-based Approach: General Overview



VWS-based Approach: Conclusions

Benefits

- Effective temporal assignation of resources.
- Suitable for HPC and HTC.
- In the future can be used for **server consolidation** and **interactive mode**.
- Modularity: Workspace Manager will manage other drivers apart of VW driver for Globus.
- Does **not require** the **installation** of the **scientific software in remote resources**.
- Inherits the isolation and security from virtual machines.
- Potentially compatible with other virtualization tools as VMWare, UML...

Drawbacks

VWS is strongly bounded to Globus architecture → only can be used in EGEE in mixed projects as Open Science Grid (Edge Services).

VWS software is in a very initial release.

Future work

In Straightforward deployment

- Perform production tests in EGEE.
- Optimize management of images.
- Adjust EGEE Information Systems to schedule efficiently VMs.
- Performance study of some experiments (ALICE, LHCb…) to minimize its costs in computational resources by means of VMs.

In VWS-based approach

- Exploring all possibilities that VWS offers not bounded only to computational problems:
  - Dynamic server provisioning (databases, web…).
  - Virtual Clusters.
  - Interactive jobs.