Virtualization, Grid, Cloud: Integration Paths for Scientific Computing

Or, where and how will my efficient large-scale computing applications be executed?

D. Salomoni, INFN Tier-1 Computing Manager

Davide.Salomoni@cnaf.infn.it

INFN-CNAF

ESC09, 17/10/2009

Outline

Outline

Setting the Scene

Q: Any suggestions you've got for appropriate books for pretty bright kids at a good liberal arts school would be appreciated.

A: You can throw in Voltaire just about anywhere. Make sure, though, if you do, to say, "*in this, the best of all possible worlds, it is simple to see that* ***we'll just have to rage in and adb the bejesus out of this hideously broken program****. And adb, of course, is the best of all possible debuggers.*" The kids'll love this.

(*Anonymous*, HEPCAL, 2002)

The Grid and Scientific Computing

Everybody more of less knows what *The Grid* is – or, at least, what the Grid is (or was) supposed to be.

The document on **Common Use Cases for a HEP Common Application Layer**, a.k.a. *HEPCAL* *, written in 2002, stated:

> *By the name Grid, we understand a widely distributed computing infrastructure, including hardware resources and the corresponding software tools and services, which allow optimal execution of computational tasks, with appropriate access to the distributed data. The implementation of such an infrastructure is beyond the scope of this document. The Grid is assumed to provide proper authentication and authorisation, transparent access to resources, and overall management of the necessary databases.*

*http://lcg.web.cern.ch/LCG/policydocs.htm

The Grid and Scientific Computing

Everybody more of less knows what *The Grid* is – or, at least, what the Grid is (or was) supposed to be.

The document on **Common Use Cases for a HEP Common Application Layer**, a.k.a. *HEPCAL* *, written in 2002, stated:

> *By the name Grid, we understand a widely distributed computing infrastructure, including hardware resources and the corresponding software tools and services, which allow optimal execution of computational tasks, with appropriate access to the distributed data. The implementation of such an infrastructure is beyond the scope of this document. The Grid is assumed to provide proper authentication and authorisation, transparent access to resources, and overall management of the necessary databases.*

*http://lcg.web.cern.ch/LCG/policydocs.htm

A Grid Success Story: EGEE

It is hard to deny that impressive achievements have been reached in the Grid world since HEPCAL was written.

Example Case: the **Enabling Grids for E-sciencE** (EGEE) project*, co-funded by the European Commission. As of October 2009:

- 260 sites across 55 countries.
- More than 150,000 CPU cores available 24x7; 21 PB of disk storage, 41 PB of tape storage; 330k jobs per day.
- Security & Policy in place; 140 registered Virtual Organizations, 14,000 registered users.
- Funding: 32MEuro contributed by the EU for 24 months; total budget is ~47MEuro, with additional ~50MEuro in computing resources contributed by the partners.
- Manpower: ~9000 Person Months, ~50% of which are contributed by the partners.

*http://www.eu-egee.org/

A Grid Success Story: EGEE

It is hard to deny that impressive achievements have been reached in the Grid world since HEPCAL was written.

Example Case: the **Enabling Grids for E-sciencE** (EGEE) project*, co-funded by the European Commission. As of October 2009:

- 260 sites across 55 countries.
- More than 150,000 CPU cores available 24x7; 21 PB of disk storage, 41 PB of tape storage; 330k jobs per day.
- Security & Policy in place; 140 registered Virtual Organizations, 14,000 registered users.
- Funding: 32MEuro contributed by the EU for 24 months; total budget is ∼47MEuro, with additional ∼50MEuro in computing resources contributed by the partners.
- Manpower: ∼9000 Person Months, ∼50% of which are contributed by the partners.

*http://www.eu-egee.org/

How Do You Use The Grid?

No gory details here, but in very general terms:

- You need to be part of a Virtual Organization. If your experiment/collaboration is not part of an existing Grid Virtual Organization, you must set it up or join an existing one.
- You access the Grid via a User Interface, using a personal X.509 certificate. Your jobs will be submitted through this user interface, specifying various parameters via a *Job Description Language*.
- Your job requirements will be matched against the available resources. Supposing these resources are found and available to you, your jobs will sooner or later run somewhere.
- You will be able to control the status of your job and retrieve job output. Provision is made to store, find and retrieve data.

Upcoming relevant events:

- III Corso INFN di formazione per **amministratori** di siti Grid (Catania, Nov 2-6)
- III Scuola INFN per **utenti** LHC della Grid (CNAF, Nov 25-27)

How Do You Use The Grid?

No gory details here, but in very general terms:

- You need to be part of a Virtual Organization. If your experiment/collaboration is not part of an existing Grid Virtual Organization, you must set it up or join an existing one.
- You access the Grid via a User Interface, using a personal X.509 certificate. Your jobs will be submitted through this user interface, specifying various parameters via a *Job Description Language*.
- Your job requirements will be matched against the available resources. Supposing these resources are found and available to you, your jobs will sooner or later run somewhere.
- You will be able to control the status of your job and retrieve job output. Provision is made to store, find and retrieve data.

Upcoming relevant events:

- III Corso INFN di formazione per **amministratori** di siti Grid (Catania, Nov 2-6)
- III Scuola INFN per **utenti** LHC della Grid (CNAF, Nov 25-27)

Outline

1. Technologies for Scientific Computing
   - Grid
   - Cloud
   - Virtualization
2. Integration, A Case Example
   - WNoD: Virtualization of Traditional Farm Resources
   - Extending the WNoD Architecture
   - Status: Where Are We?
   - Distributed Grid Submission and Cloud Services
   - Future Work

The Cloud and Scientific Computing

Like *Grid* was back in 2002, *Cloud* is now enjoying massive popularity in both technical and non-technical fora. And like the term Grid was, the term Cloud is generic enough to fit many bills.

Take e.g. the paper **Scientific Computing in the Cloud***, written Dec 30, 2008:

> *For research groups, cloud computing provides convenient access to reliable, high performance clusters and storage, without the need to purchase and maintain sophisticated hardware. For developers, virtualization allows scientific codes to be optimized and pre-installed on machine images, facilitating control over the computational environment.*

*http://arxiv.org/pdf/0901.0029v1

A Recent Example

October 14, 2009: DOE to explore scientific cloud computing at Argonne, Lawrence Berkeley National Laboratories.*

> *A new program funded by the American Recovery and Reinvestment Act through the U.S. Department of Energy (DOE) will examine cloud computing as a cost-effective and energy-efficient computing paradigm for scientists to accelerate discoveries in a variety of disciplines, including analysis of scientific data sets in biology, climate change and physics.*

> *DOE is funding the project at $32 million.*

To put things in perspective, the EGI (European Grid Initiative) project, the EGEE to-be-sustainable successor, is seeking to get €25 million from the EU in 4 years for a European-wide production infrastructure.

* http://newscenter.lbl.gov/press-releases/2009/10/14/scientific-cloud-computing/

A Recent Example

October 14, 2009: DOE to explore scientific cloud computing at Argonne, Lawrence Berkeley National Laboratories.*

> *A new program funded by the American Recovery and Reinvestment Act through the U.S. Department of Energy (DOE) will examine cloud computing as a cost-effective and energy-efficient computing paradigm for scientists to accelerate discoveries in a variety of disciplines, including analysis of scientific data sets in biology, climate change and physics.*
>
> *DOE is funding the project at $32 million.*

To put things in perspective, the EGI (European Grid Initiative) project, the EGEE to-be-sustainable successor, is seeking to get €25 million from the EU in 4 years for a European-wide production infrastructure.

*http://newscenter.lbl.gov/press-releases/2009/10/14/scientific-cloud-computing/

A Recent Example

October 14, 2009: DOE to explore scientific cloud computing at Argonne, Lawrence Berkeley National Laboratories.*

> *A new program funded by the American Recovery and Reinvestment Act through the U.S. Department of Energy (DOE) will examine cloud computing as a cost-effective and energy-efficient computing paradigm for scientists to accelerate discoveries in a variety of disciplines, including analysis of scientific data sets in biology, climate change and physics.*
>
> *DOE is funding the project at $32 million.*

To put things in perspective, the EGI (European Grid Initiative) project, the EGEE to-be-sustainable successor, is seeking to get €25 million from the EU in 4 years for a European-wide production infrastructure.

*http://newscenter.lbl.gov/press-releases/2009/10/14/scientific-cloud-computing/

Cloud: the Amazon Definitions

Amazon provides a relatively easy to understand definition of the Cloud services they provide, called *Amazon Elastic Compute Cloud* (or Amazon EC2) and *Amazon Simple Storage Service* (or Amazon S3):

> *Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.*

> *Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.*

Cloud: the Amazon Definitions

Amazon provides a relatively easy to understand definition of the Cloud services they provide, called *Amazon Elastic Compute Cloud* (or Amazon EC2) and *Amazon Simple Storage Service* (or Amazon S3):

> *Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.*

> *Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.*

Can This Be Really Used in the Scientific World?

It appears that at least in some cases the answer may be yes. Take for example the Gaia project* of the European Space Agency, whose goal is to survey about a billion stars to make an extremely precise three-dimensional map of our galaxy:

> *For the full 1 billion star project numbers [Gaia Science Operations Development Team] calculated that they will analyze 100 million primary stars, plus 6 years of data, which will require a total of 16,200 hours of a 20-node EC2 cluster. That's an estimated total computing cost of 344,000 Euros. By comparison, an in-house solution would cost roughly 720,000 EUR (at today's prices) – which doesn't include electricity or storage or sys-admin costs. (Storage alone would be an additional 100,000 EUR.)*§

*http://www.esa.int/esaSC/120377_index_0_m.html
§http://aws.typepad.com/aws/2009/06/scaling-to-the-stars.html

Some Cloudy Terminology

Functionality in Cloud services can be described in many ways. We'll focus on the term *service* here.

- **HaaS**, *Hardware as a Service*. You access hardware (e.g., CPUs) that is made available to you. This could generally apply to both Grid and Cloud services.
- **SaaS**, *Software as a Service*. (or Service as a Software?) You directly access the services that interest you.
- **DaaS**, *Data as a Service*. Data Access from various sources in various formats.

Some Cloudy Terminology

Functionality in Cloud services can be described in many ways. We'll focus on the term *service* here.

- **HaaS**, *Hardware as a Service*. You access hardware (e.g., CPUs) that is made available to you. This could generally apply to both Grid and Cloud services.
- **SaaS**, *Software as a Service*. (or Service as a Software?) You directly access the services that interest you.
- **DaaS**, *Data as a Service*. Data Access from various sources in various formats.

Some Cloudy Terminology

Functionality in Cloud services can be described in many ways. We'll focus on the term *service* here.

- **HaaS**, *Hardware as a Service*. You access hardware (e.g., CPUs) that is made available to you. This could generally apply to both Grid and Cloud services.
- **SaaS**, *Software as a Service*. (or Service as a Software?) You directly access the services that interest you.
- **DaaS**, *Data as a Service*. Data Access from various sources in various formats.

Some Cloudy Terminology

Functionality in Cloud services can be described in many ways. We'll focus on the term *service* here.

- **HaaS**, *Hardware as a Service*. You access hardware (e.g., CPUs) that is made available to you. This could generally apply to both Grid and Cloud services.
- **SaaS**, *Software as a Service*. (or Service as a Software?) You directly access the services that interest you.
- **DaaS**, *Data as a Service*. Data Access from various sources in various formats.

Cloudiness Galore

One could then argue that putting the previous definitions together, Cloud delivers Platform as a Service (**PaaS**) functionality.

$$HaaS + Saas + Daas = PaaS$$

In general, I would say that another interesting functionality we need to consider is Cloud Computing as Infrastructure as a Service (**IaaS**), with particular regard to the interconnection of Cloud providers (a Cloud of Clouds).

Outline

Virtualization, or Resource Abstraction

Nothing particularly new *per se*: virtual machines have existed for years, like other virtualization technologies, e.g. virtual memory, virtual storage, virtual networks.

- The IBM M44/44X explored paging systems and the virtual machine concept in 1965.
- The SoftPC software emulator of x86 hardware was introduced in 1988.
- The first version of the open-source XEN was released in 2003.
- FEDERICA (Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures)* is a 30-months (1/2008-6/2010) EU co-funded project to support research in virtualization of e-Infrastructures integrating network resources and nodes capable of virtualization.

What changed is how powerful, ubiquitous, and (relatively) easy to use virtualization technologies are or are starting to be.

*http://www.fp7-federica.eu/

Virtualization, or Resource Abstraction

Nothing particularly new *per se*: virtual machines have existed for years, like other virtualization technologies, e.g. virtual memory, virtual storage, virtual networks.

- The IBM M44/44X explored paging systems and the virtual machine concept in 1965.
- The SoftPC software emulator of x86 hardware was introduced in 1988.
- The first version of the open-source XEN was released in 2003.
- FEDERICA (Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures)* is a 30-months (1/2008-6/2010) EU co-funded project to support research in virtualization of e-Infrastructures integrating network resources and nodes capable of virtualization.

What changed is how powerful, ubiquitous, and (relatively) easy to use virtualization technologies are or are starting to be.

---

*http://www.fp7-federica.eu/

Virtualization, or Resource Abstraction

Nothing particularly new *per se*: virtual machines have existed for years, like other virtualization technologies, e.g. virtual memory, virtual storage, virtual networks.

- The IBM M44/44X explored paging systems and the virtual machine concept in 1965.
- The SoftPC software emulator of x86 hardware was introduced in 1988.
- The first version of the open-source XEN was released in 2003.
- FEDERICA (Federated E-infrastructure Dedicated to European Researchers Innovating in Computing network Architectures)* is a 30-months (1/2008-6/2010) EU co-funded project to support research in virtualization of e-Infrastructures integrating network resources and nodes capable of virtualization.

What changed is how powerful, ubiquitous, and (relatively) easy to use virtualization technologies are or are starting to be.

*http://www.fp7-federica.eu/

Overview: How Good or Bad Virtualization Is?

There are many competing ways to answer this question, and they may depend on the particular perspective or focus one likes to adopt (e.g. performance, resiliency, ease of deployment, overall cost savings.)

For what regards **performance**, we recently* compared both qualitatively and quantitatively XEN† and KVM§.

Some details:

- Host O/S: Intel E5420, SL 5.2 x86_64, kernel 2.6.18-92.1.22.el5
- VM O/S: SLC 4.5 i386, kernel 2.6.9-67.0.15.EL.cern
- KVM version 83, XEN version 3.2.1

---

* A.Chierici and R.Veraldi, *A comparison between xen and kvm*, CHEP09, http://www.particle.cz/conferences/chep2009/

† http://www.xen.org/

§ http://www.linux-kvm.org/page/Main_Page

Overview: How Good or Bad Virtualization Is?

There are many competing ways to answer this question, and they may depend on the particular perspective or focus one likes to adopt (e.g. performance, resiliency, ease of deployment, overall cost savings.)

For what regards **performance**, we recently* compared both qualitatively and quantitatively XEN† and KVM§.

Some details:

- Host O/S: Intel E5420, SL 5.2 x86_64, kernel 2.6.18-92.1.22.el5
- VM O/S: SLC 4.5 i386, kernel 2.6.9-67.0.15.EL.cern
- KVM version 83, XEN version 3.2.1

---

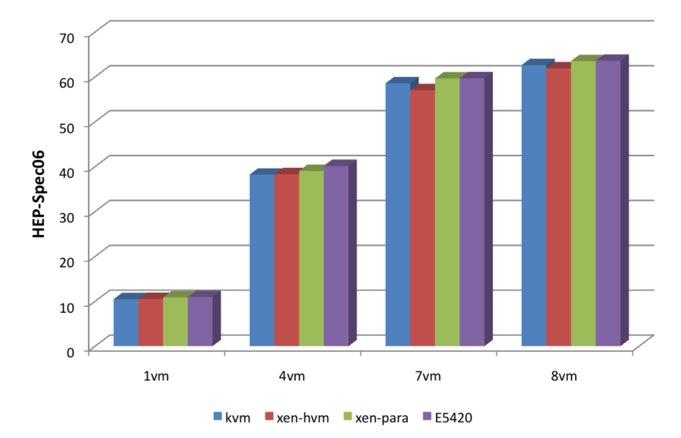
*A.Chierici and R.Veraldi, *A comparison between xen and kvm*, CHEP09, http://www.particle.cz/conferences/chep2009/

†http://www.xen.org/

§http://www.linux-kvm.org/page/Main_Page

Benchmarks: HEP-SPEC06 (1)

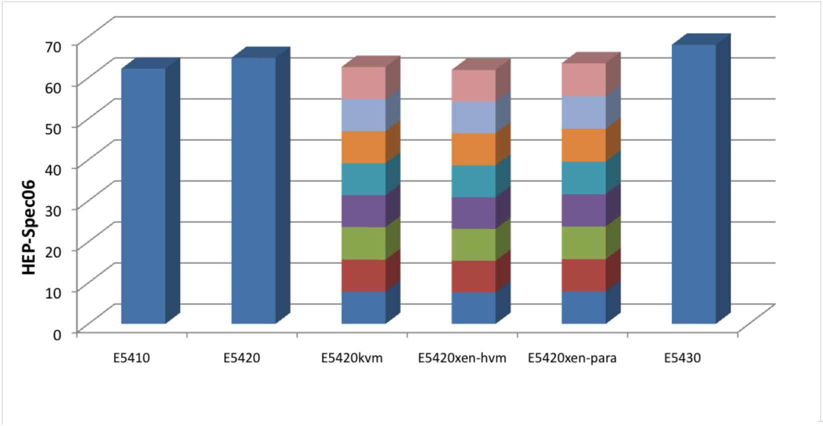
The plot shows HEP-SPEC* performance for various hypervisor and physical hardware combinations while an increasing number of virtual machines runs concurrently on the host.



*http://www.italiangrid.org/grid_operations/site_manager/HEP-SPEC06

Benchmarks: HEP-SPEC06 (2)

This is a comparison of HEP-SPEC performance of 8 VMs against performance on physical CPUs.



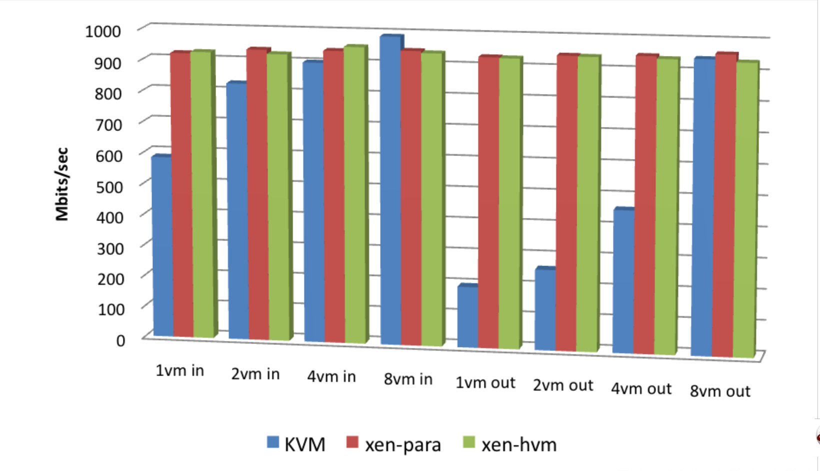
Benchmarks: HEP-SPEC06 (3)

The following table shows the HEP-SPEC percentage loss due to virtualization compared to physical CPUs.

Virtualization Technology	% loss from non Emulated CPU (E5420, 8vm)
E5420kvm	3,42
E5420xen-hvm	4,55
E5420xen-para	2,02
E5410 vs. E5420	4,07

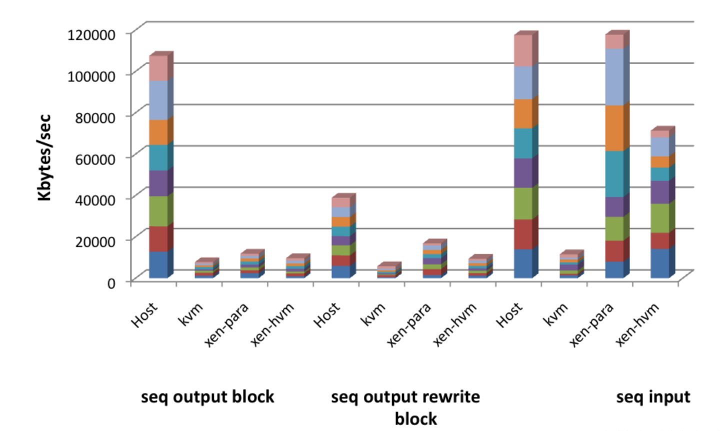
Benchmarks: network

Network performance was tested with `iperf` measuring both inbound and outbound throughput, with a 256KB TCP window size, 5 parallel connections, for 15 minutes.



Benchmarks: disk I/O

Disk access speed, measured with `bonnie++`, shows that a lot is still to be improved, in particular for what regards I/O writes (and in particular for KVM). The tests were performed with a disk-on-a-file rather than with the more performant disk-on-a-partition approach.



Virtualization: Some Considerations

During our tests KVM proved great reliability: it never crashed and integrated seamlessly into our computing farm, without requiring any special effort to the system administrators.

There are several areas where performance improvements are expected and needed; clock drift in VMs is still sometimes an issue; in some of these areas, like in I/O through the adoption of paravirtualized block devices (`virtio`) with KVM, things have already somewhat improved.

While reliability is certainly a key issue (we had several unexpected behaviors with XEN), usability is another one: how easily can you deploy and manage *many* VMs? Or, how can you setup complex networking configurations? (There seems to be some interesting development on the horizon for this, like `netcf`*).

So, virtual machines can be used in production services today, but one still needs to know fairly in detail how to deal with them, and work still needs to be done on the technology.

*https://fedorahosted.org/netcf/

Virtualization: Some Considerations

During our tests KVM proved great reliability: it never crashed and integrated seamlessly into our computing farm, without requiring any special effort to the system administrators.

There are several areas where performance improvements are expected and needed; clock drift in VMs is still sometimes an issue; in some of these areas, like in I/O through the adoption of paravirtualized block devices (`virtio`) with KVM, things have already somewhat improved.

While reliability is certainly a key issue (we had several unexpected behaviors with XEN), usability is another one: how easily can you deploy and manage *many* VMs? Or, how can you setup complex networking configurations? (There seems to be some interesting development on the horizon for this, like `netcf`*).

So, virtual machines can be used in production services today, but one still needs to know fairly in detail how to deal with them, and work still needs to be done on the technology.

*`https://fedorahosted.org/netcf/`

Virtualization: Some Considerations

During our tests KVM proved great reliability: it never crashed and integrated seamlessly into our computing farm, without requiring any special effort to the system administrators.

There are several areas where performance improvements are expected and needed; clock drift in VMs is still sometimes an issue; in some of these areas, like in I/O through the adoption of paravirtualized block devices (`virtio`) with KVM, things have already somewhat improved.

While reliability is certainly a key issue (we had several unexpected behaviors with XEN), usability is another one: how easily can you deploy and manage *many* VMs? Or, how can you setup complex networking configurations? (There seems to be some interesting development on the horizon for this, like `netcf`*).

So, virtual machines can be used in production services today, but one still needs to know fairly in detail how to deal with them, and work still needs to be done on the technology.

*`https://fedorahosted.org/netcf/`

Virtualization: Some Considerations

During our tests KVM proved great reliability: it never crashed and integrated seamlessly into our computing farm, without requiring any special effort to the system administrators.

There are several areas where performance improvements are expected and needed; clock drift in VMs is still sometimes an issue; in some of these areas, like in I/O through the adoption of paravirtualized block devices (`virtio`) with KVM, things have already somewhat improved.

While reliability is certainly a key issue (we had several unexpected behaviors with XEN), usability is another one: how easily can you deploy and manage *many* VMs? Or, how can you setup complex networking configurations? (There seems to be some interesting development on the horizon for this, like `netcf`*).

So, virtual machines can be used in production services today, but one still needs to know fairly in detail how to deal with them, and work still needs to be done on the technology.

---

*`https://fedorahosted.org/netcf/`

Why Integration, Then?

From the previous slides, it has hopefully emerged that Virtualization on the one hand, and Grid and Cloud on the other are technologies that should mutually integrate.

- **Virtualization** provides the flexibility to create ad-hoc services.
- **Grids and Clouds** provide mechanisms to exploit the services with different interfaces and use cases.

All three should be viewed in the larger context of an open, global, interconnected Virtual Infrastructure (an **Infrastructure as a Service**).

Like many times in the past, 95% of the problem is. . . **agreement**.

Why Integration, Then?

From the previous slides, it has hopefully emerged that Virtualization on the one hand, and Grid and Cloud on the other are technologies that should mutually integrate.

- **Virtualization** provides the flexibility to create ad-hoc services.
- **Grids and Clouds** provide mechanisms to exploit the services with different interfaces and use cases.

All three should be viewed in the larger context of an open, global, interconnected Virtual Infrastructure (an **Infrastructure as a Service**).

Like many times in the past, 95% of the problem is. . . **agreement**.

Why Integration, Then?

From the previous slides, it has hopefully emerged that Virtualization on the one hand, and Grid and Cloud on the other are technologies that should mutually integrate.

- **Virtualization** provides the flexibility to create ad-hoc services.
- **Grids and Clouds** provide mechanisms to exploit the services with different interfaces and use cases.

All three should be viewed in the larger context of an open, global, interconnected Virtual Infrastructure (an **Infrastructure as a Service**).

Like many times in the past, 95% of the problem is. . . **agreement**.

Why Integration, Then?

From the previous slides, it has hopefully emerged that Virtualization on the one hand, and Grid and Cloud on the other are technologies that should mutually integrate.

- **Virtualization** provides the flexibility to create ad-hoc services.
- **Grids and Clouds** provide mechanisms to exploit the services with different interfaces and use cases.

All three should be viewed in the larger context of an open, global, interconnected Virtual Infrastructure (an **Infrastructure as a Service**).

Like many times in the past, 95% of the problem is. . . **agreement**.

Outline

Common Requests to Computing Resource Providers

- Operating System:
  - "I want SLC3". *"Hey, no, my application [only runs | is only certified] on Ubuntu 9.04."* "What? Forget it! I need `afs` and SL5."
- Applications:
  - "I *absolutely* need you to install application X.Y version Z on all your nodes."
  - "Please don't change *that* system library!" ("and don't you dare to upgrade the kernel!")
- Intra-VO requirements may also apply, e.g. different sets of users belonging to the same VO may have different needs.
- The INFN Tier-1 currently supports ∼20 Virtual Organizations.
  - Optimize resource usage, e.g. avoid static allocations, try not to waste CPU cycles, do not [buy | set-up] dedicated infrastructures to fix requirements/problems.
  - Maintain full control of the site.
  - Do not change established workflows.

Common Requests to Computing Resource Providers

- Operating System:
  - "I want SLC3". *"Hey, no, my application [only runs | is only certified] on Ubuntu 9.04."* "What? Forget it! I need `afs` and SL5."
- Applications:
  - "I *absolutely* need you to install application X.Y version Z on all your nodes."
  - "Please don't change *that* system library!" ("and don't you dare to upgrade the kernel!")
- Intra-VO requirements may also apply, e.g. different sets of users belonging to the same VO may have different needs.
- The INFN Tier-1 currently supports ∼20 Virtual Organizations.
  - Optimize resource usage, e.g. avoid static allocations, try not to waste CPU cycles, do not [buy | set-up] dedicated infrastructures to fix requirements/problems.
  - Maintain full control of the site.
  - Do not change established workflows.

Common Requests to Computing Resource Providers

- Operating System:
  - "I want SLC3". *"Hey, no, my application [only runs | is only certified] on Ubuntu 9.04."* "What? Forget it! I need `afs` and SL5."
- Applications:
  - "I *absolutely* need you to install application X.Y version Z on all your nodes."
  - "Please don't change *that* system library!" ("and don't you dare to upgrade the kernel!")
- Intra-VO requirements may also apply, e.g. different sets of users belonging to the same VO may have different needs.
- The INFN Tier-1 currently supports ∼20 Virtual Organizations.
  - Optimize resource usage, e.g. avoid static allocations, try not to waste CPU cycles, do not [buy | set-up] dedicated infrastructures to fix requirements/problems.
  - Maintain full control of the site.
  - Do not change established workflows.

Common Requests to Computing Resource Providers

- Operating System:
  - "I want SLC3". *"Hey, no, my application [only runs | is only certified] on Ubuntu 9.04."* "What? Forget it! I need `afs` and SL5."
- Applications:
  - "I *absolutely* need you to install application X.Y version Z on all your nodes."
  - "Please don't change *that* system library!" ("and don't you dare to upgrade the kernel!")
- Intra-VO requirements may also apply, e.g. different sets of users belonging to the same VO may have different needs.
- The INFN Tier-1 currently supports ∼20 Virtual Organizations.
  - Optimize resource usage, e.g. avoid static allocations, try not to waste CPU cycles, do not [buy | set-up] dedicated infrastructures to fix requirements/problems.
  - Maintain full control of the site.
  - Do not change established workflows.

Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

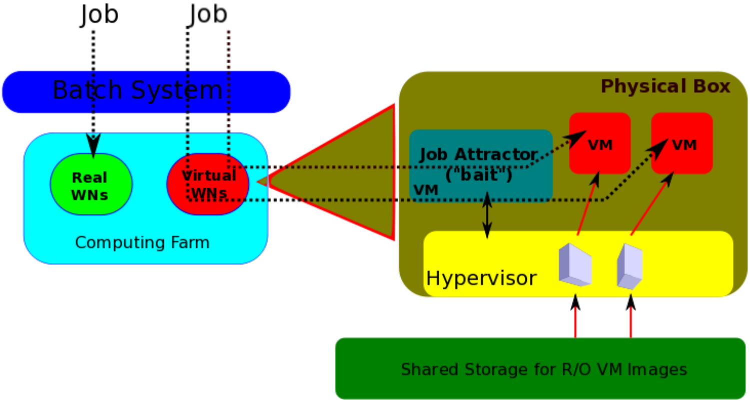
Use Virtual Machines as Compute Nodes

Making them available *on-demand*, i.e. when needed

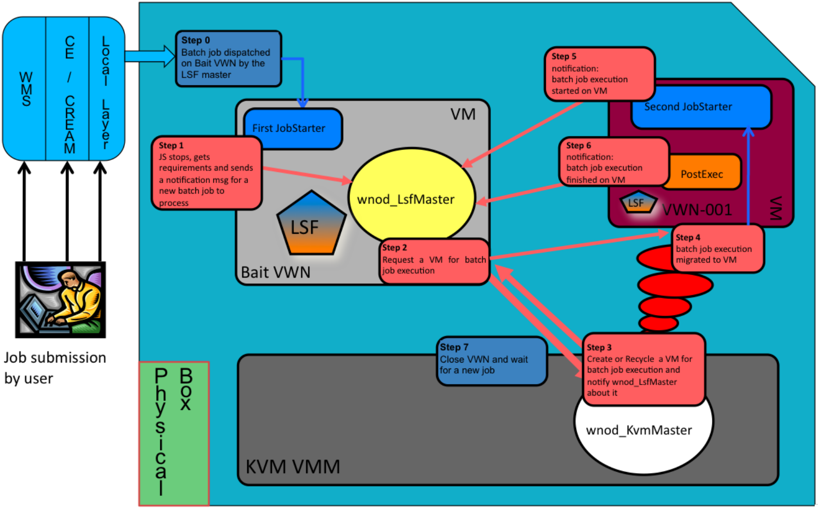
The main use case we initially considered within INFN was to provide dynamic virtual execution environments. We wrote a software called virtual *Worker Nodes on Demand* (WNoD), with the following characteristics:

- Decouple installed software from physical hardware.
- WNoD VMs can run different images (e.g., different O/S, or same O/S but with different package sets) for different (sets of) users.
- VMs are created *on-demand*, i.e. at job execution time, so resources are always dynamically shared.
- VM images are initially retrieved from shared storage (`gpfs` at the INFN Tier-1) and then stored locally.
- VM images are **never** obtained directly by users, and are always under the control of the resource provider.
- WNoD *does not alter* user procedures for job submission.

Worker Nodes on Demand High-Level Overview



The Worker Nodes on Demand Process Flow



Outline

Dynamic Selection of Services beyond Local Site Resources

WNoD originally provided a solution to dynamically manage computing environments for *local* or *grid* jobs in a strictly site-specific fashion. We found it interesting to extend this mechanism to:

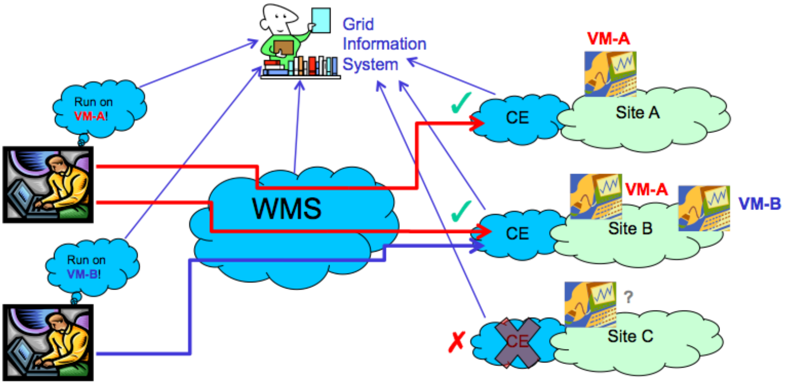
- Select VM images through standard Grid jobs to use custom (e.g., per-VO, per-role) resources.
- Provide cloud-like services, e.g. standard computing cores accessible via regular `ssh`.

Dynamic Selection of Services beyond Local Site Resources

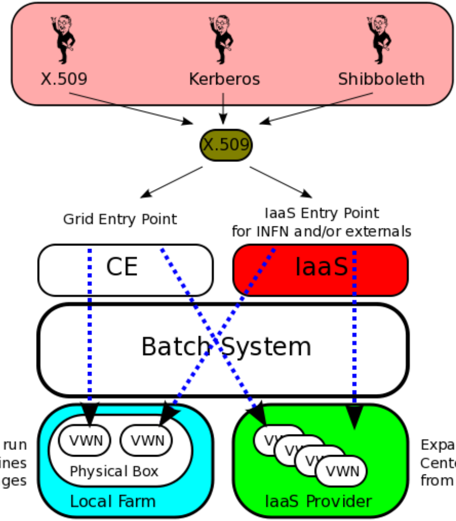
WNoD originally provided a solution to dynamically manage computing environments for *local* or *grid* jobs in a strictly site-specific fashion. We found it interesting to extend this mechanism to:

- Select VM images through standard Grid jobs to use custom (e.g., per-VO, per-role) resources.
- Provide cloud-like services, e.g. standard computing cores accessible via regular `ssh`.

Dynamic Selection of Resources through Standard Grid Jobs



Make Grid and Cloud Services Converge around Dynamically Provisioned Resources



Users transparently run jobs on virtual machines executing known images

Expand capacity of Computing Centers through provisioning from IaaS providers

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Which Use Cases Are Now Covered, Then?

- Selection of per-customer (e.g. per-user, per-VO) images for WNs, either locally or via the grid.
- Simplification of WN installation and upgrade procedures.
- Provisioning of on-demand farms, e.g. to build interactive analysis facilities, or just to provide servers to customers.
- Support of *peak requests* resorting to external resource providers.
- Provisioning of pre-packaged on-demand services, e.g. dynamic web server / squid / StoRM front-end farms.
- Simplification of grid or cloud usage through support of several authentication methods.

Outline

1. Technologies for Scientific Computing
   - Grid
   - Cloud
   - Virtualization

2. Integration, A Case Example
   - WNoD: Virtualization of Traditional Farm Resources
   - Extending the WNoD Architecture
   - Status: Where Are We?
   - Distributed Grid Submission and Cloud Services
   - Future Work

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently ~280 production VMs (~10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently $\sim$280 production VMs ($\sim$10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently ~280 production VMs (~10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently ∼280 production VMs (∼10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently ∼280 production VMs (∼10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

WNoD Deployment

WNoD are being deployed in production at the INFN Tier-1:

- Before deployment, we ran more than 400k test jobs. There are currently ~280 production VMs (~10% of the total Tier-1 capacity).
- SL5 only runs on WNoD. Several VOs run on WNoD only, some (e.g. ETICS) on VO-specific SL5 images.
- ATLAS and CDF transparently run SLC 4.4 on both traditional and virtual WNs.
- CMS successfully tested SL5.3 on WNoD and is expected to run on it soon (on either SLC4.4 or SL5.3). ATLAS, Alice and LHCb will start tests of SL5.3 shortly.
- Thanks to WNoD, **migration to SL5** is as flexible as it can be: tell us when you're ready, and we'll direct you to the images you want, without static allocations. Also, **kernel upgrades** do not require the reboot of multi-core nodes (push a new image to all or a subset of hypervisors).
- Monitoring and accounting already supporting WNoD.

Outline

Integration with gLite Job Submission

VM images may be selected through standard grid tools:

- Available VM images are published in the Grid Information System using the Glue attribute `SoftwareRunTimeEnvironment`.
- Selection of VM images is done by users via standard JDL statements, e.g.

```
CeRequirements = "(Member(\"vm_etics_slc44\",
  other.GlueHostSoftwareApplicationRuntimeEnvironment))"
```

- Users (or sets thereof) must be authorized by resource providers to locally use the selected VM images through proper WNoD configuration.
- This works today with the latest CREAM and WMS builds.

Integration with gLite Job Submission

VM images may be selected through standard grid tools:

- Available VM images are published in the Grid Information System using the Glue attribute `SoftwareRunTimeEnvironment`.
- Selection of VM images is done by users via standard JDL statements, e.g.

```
CeRequirements = "(Member(\"vm_etics_slc44\",
  other.GlueHostSoftwareApplicationRuntimeEnvironment))"
```

- Users (or sets thereof) must be authorized by resource providers to locally use the selected VM images through proper WNoD configuration.
- This works today with the latest CREAM and WMS builds.

Integration with gLite Job Submission

VM images may be selected through standard grid tools:

- Available VM images are published in the Grid Information System using the Glue attribute `SoftwareRunTimeEnvironment`.
- Selection of VM images is done by users via standard JDL statements, e.g.

```
CeRequirements = "(Member(\"vm_etics_slc44\",
  other.GlueHostSoftwareApplicationRuntimeEnvironment))"
```

- Users (or sets thereof) must be authorized by resource providers to locally use the selected VM images through proper WNoD configuration.
- This works today with the latest CREAM and WMS builds.

Integration with gLite Job Submission

VM images may be selected through standard grid tools:

- Available VM images are published in the Grid Information System using the Glue attribute `SoftwareRunTimeEnvironment`.
- Selection of VM images is done by users via standard JDL statements, e.g.

```
CeRequirements = "(Member(\"vm_etics_slc44\",
  other.GlueHostSoftwareApplicationRuntimeEnvironment))"
```

- Users (or sets thereof) must be authorized by resource providers to locally use the selected VM images through proper WNoD configuration.
- This works today with the latest CREAM and WMS builds.

The Cloud: Providing ssh Access to Compute Nodes

- We are writing a layer to interface WNoD with cloud APIs, like OCCI (Open Cloud Computing Interface) or Amazon EC2.
- You request a computing resource, get an IP address back, `ssh` to the resource, do your work, and then dispose of the resource.
- This works today with a very basic user interface and a subset of the proposed OCCI API. It dynamically instantiates a VM taken from the standard set of Tier-1 resources and allocates it to the requester through WNoD and the LRMS.

The Cloud: Providing ssh Access to Compute Nodes

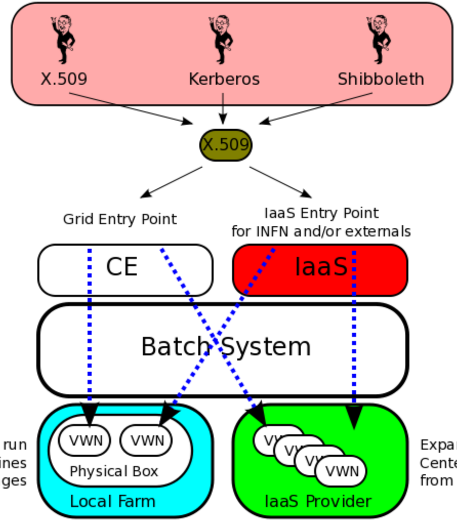
- We are writing a layer to interface WNoD with cloud APIs, like OCCI (Open Cloud Computing Interface) or Amazon EC2.
- You request a computing resource, get an IP address back, ssh to the resource, do your work, and then dispose of the resource.
- This works today with a very basic user interface and a subset of the proposed OCCI API. It dynamically instantiates a VM taken from the standard set of Tier-1 resources and allocates it to the requester through WNoD and the LRMS.

The Cloud: Providing `ssh` Access to Compute Nodes

- We are writing a layer to interface WNoD with cloud APIs, like OCCI (Open Cloud Computing Interface) or Amazon EC2.
- You request a computing resource, get an IP address back, `ssh` to the resource, do your work, and then dispose of the resource.
- This works today with a very basic user interface and a subset of the proposed OCCI API. It dynamically instantiates a VM taken from the standard set of Tier-1 resources and allocates it to the requester through WNoD and the LRMS.

Make Grid and Cloud Services Converge around Dynamically Provisioned Resources

Users transparently run jobs on virtual machines executing known images



Expand capacity of Computing Centers through provisioning from IaaS providers

Outline

1. Technologies for Scientific Computing
   - Grid
   - Cloud
   - Virtualization

2. Integration, A Case Example
   - WNoD: Virtualization of Traditional Farm Resources
   - Extending the WNoD Architecture
   - Status: Where Are We?
   - Distributed Grid Submission and Cloud Services
   - Future Work

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others…

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others…

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others. . .

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others…

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others…

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others...

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others. . .

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others…

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others. . .

Many Things Are Still To Come

- Full deployment of WNoD at the INFN Tier-1.
- Support for MPI/multicore jobs.
- Grid/Glue schema standardization efforts and integration of the latest WMS builds into the gLite release.
- Full compliance to OCCI standards.
- Network virtualization, security considerations.
- Cloud usage billing.
- (Transparent) Use of external cloud providers for flash requests.
- Integration with other projects.
- Make WNoD as site independent as possible, possibly target other LRMS.
- And others. . .

Summary

- WNoD works today in production at the INFN Tier-1 and is being used to solve real use cases, like migration to a new O/S (SL5) and the provisioning of custom environments.
- Distributed selection of custom execution environments in EGEE has been demonstrated.
- Dynamic integration of **cloud and grid services** using the same set of resources works today and can be exploited to provide new services.

This is just an implementation example of a possible integration strategy. Other ways or methods may be followed, but the key message here is this:

Try to exploit existing and planned investments to cover present and future use cases, dynamically sharing resources as much as possible.

Summary

- WNoD works today in production at the INFN Tier-1 and is being used to solve real use cases, like migration to a new O/S (SL5) and the provisioning of custom environments.
- Distributed selection of custom execution environments in EGEE has been demonstrated.
- Dynamic integration of **cloud and grid services** using the same set of resources works today and can be exploited to provide new services.

This is just an implementation example of a possible integration strategy. Other ways or methods may be followed, but the key message here is this:

Try to exploit existing and planned investments to cover present and future use cases, dynamically sharing resources as much as possible.

Summary

- WNoD works today in production at the INFN Tier-1 and is being used to solve real use cases, like migration to a new O/S (SL5) and the provisioning of custom environments.
- Distributed selection of custom execution environments in EGEE has been demonstrated.
- Dynamic integration of **cloud and grid services** using the same set of resources works today and can be exploited to provide new services.

This is just an implementation example of a possible integration strategy. Other ways or methods may be followed, but the key message here is this:

Try to exploit existing and planned investments to cover present and future use cases, dynamically sharing resources as much as possible.

Summary

- WNoD works today in production at the INFN Tier-1 and is being used to solve real use cases, like migration to a new O/S (SL5) and the provisioning of custom environments.
- Distributed selection of custom execution environments in EGEE has been demonstrated.
- Dynamic integration of **cloud and grid services** using the same set of resources works today and can be exploited to provide new services.

This is just an implementation example of a possible integration strategy. Other ways or methods may be followed, but the key message here is this:

Try to exploit existing and planned investments to cover present and future use cases, dynamically sharing resources as much as possible.