Using Clouds to address Grid Limitations

Giacomo Mc Evoy[1] Bruno Schulze[1]

[1]National Laboratory for Scientific Computing (LNCC)

6th International Workshop on Middleware for Grid Computing - MGC 2008

Outline

Laboratório Nacional de Computação Científica

What is Cloud Computing?

- What Cloud Computing offers
  - Massively scalable IT-related capabilities
  - Hosting different workloads that scale quickly (elasticity)
  - Faster and more transparent management
- How to achieve these
  - Utility computing, pay-as-you-go business model
  - Use of virtualization hypervisor technology
  - Dedicated clusters or commodity hardware

What is Cloud Computing?

- Different from Grid Computing
  - Grids emerge from pre-existing heterogeneous resources
  - Clouds are built according to a pre-defined design
  - Clouds have a reduced interface
  - Clouds can be designed on top of grids

Experiences in e-Science

- Cloud Computing Tested (NSF, HP, Intel, Yahoo!)
  - Research on resource allocation, scheduling and monitoring with cloud services
  - “Service Cell” comprises multiple VMs and constitute building blocks
  - Uses Hadoop and Pig Latin Dataflow Language
- Cloud CAIRN in the CARMEN e-science project (UK)
  - Allows neuroscientists to share, integrate and analyse data
  - Vast amount of data and high throughput required
  - Internally engineered with fast networking between the storage and compute servers

Laboratório Nacional de Computação Científica

Experiences in e-Science

- Eucalyptus open-source cloud computing framework (University of California)
  - Allows research of cloud computing inner architecture
  - Modular design using Web Services
  - Amazon's EC2 interface and Rocks install
- Nimbus cloud computing for e-Science
  - Part of the Science Clouds project by the Globus Alliance
  - Dynamic deployment of virtual clusters
  - Introduced the concept of contextualization to a virtual appliance

Laboratório Nacional de Computação Científica

Some current Grid Limitations

- Complex to develop and mantain applications
  - Broad interface to the application level (to offer flexibility)
  - Small details like libraries, versioning are effort-consuming
  - Provides seamless scaling for batch job execution, but not high-level services
  - Increasing complexity to scale for massive data
- Lacking design enforcements
  - Grid middleware offers access to low level functionality and services
  - Ontology usage is not ensured to access data
  - No control on how many resources can be consumed on a given machine

Features of Virtual Machines

- Add an abstraction layer between hardware and application
- Virtual appliances: Just enough Operating system (JeOS)
- Maximize usage of resources and efficiency by creating VM instances
- Hypervisor manages multiple instances of same base image
- Immutable images and snapshots add flexibility
- Can modify profile of computer resources (RAM, Storage and CPU)

Virtualization of Data

- Three levels of abstraction: Data, Information and Knowledge
- Abstraction will help find and interpret data even if encoding changes
- Ontology can provide a description of Knowledge, needs to be commonplace
- Service Oriented Scenario: User should interact with high-level services

Laboratório Nacional de Computação Científica

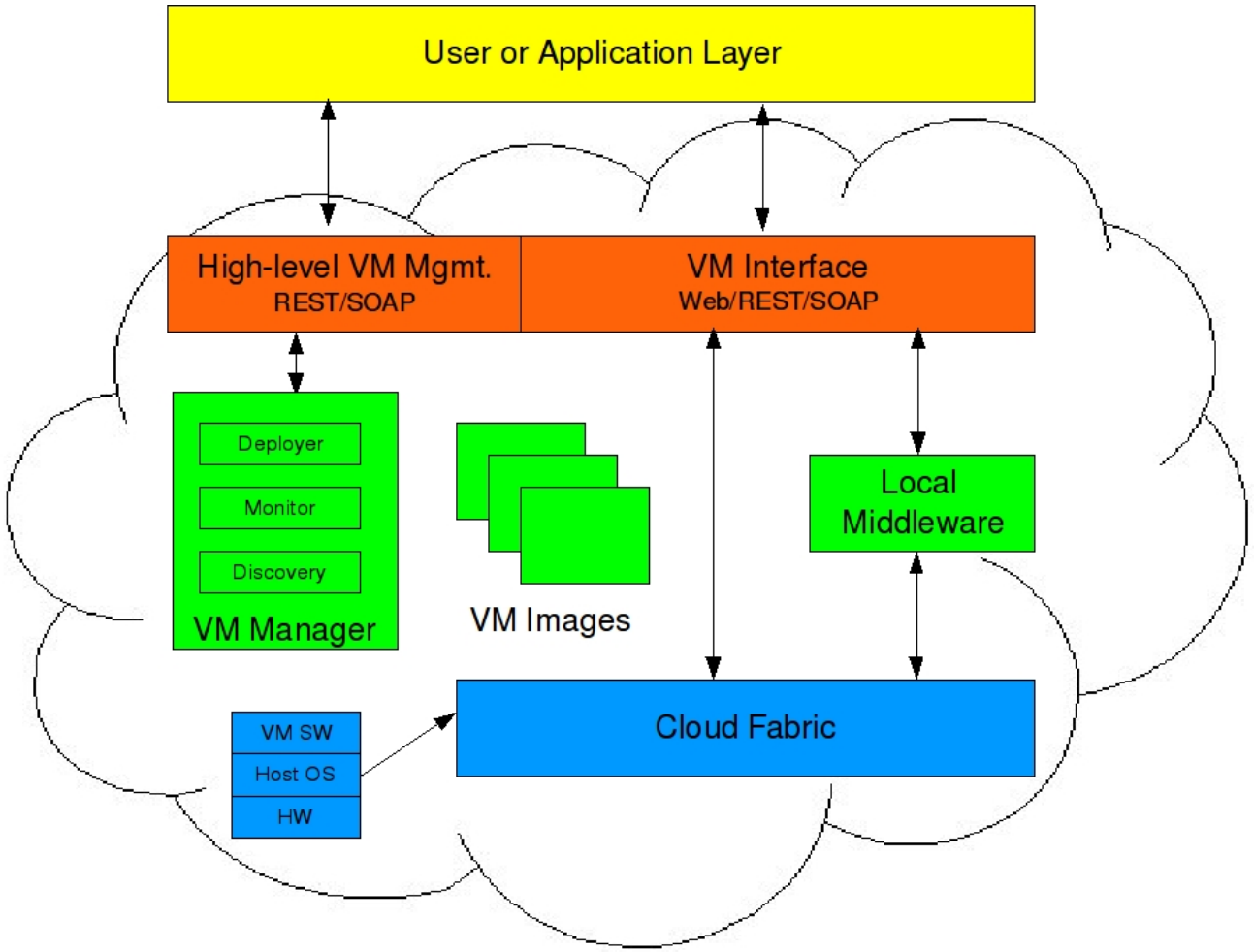
Cloud Interface

- Cloud systems have a definite and narrow user interface
  - Limited capabilities to the user, simplicity is gained
  - The purpose of creating the Cloud
- Composition between VM management interface and the instance interfaces
  - VM Management allows for creating image instances and deploying/removing them from the cloud.
  - The image instance interface for the user is usually HTTP based (HTML or Web Service)
  - In Amazon's EC2, each image instance is accessible through console

Cloud implementation

- Cloud Fabric
  - Composed of clusters or heterogeneous hardware
  - Homogeneous VM solution (Xen, VMWare, VirtualBox)
  - Hypervisor must be able to be controlled remotely
- Virtual Machine Manager
  - Handles the user's request regarding image instances
  - Monitor the state of execution of image instances
  - Can manage automatic scaling of resources
- Local middleware
  - Creates additional software abstractions
  - Seamless distributed file share (ex: Amazon's S3)
  - Graphical interface translation of resources (ex: 3Tera's AppLogic)

Cloud design



Laboratório Nacional de Computação Científica

Clouds for simplicity

- Broad interface to the application level
- Reduced user interface
  - Removes functionality, but many users may not need it
  - Easier to build an application against it, easier to mantain inner architecture
- Small details like libraries, versioning are effort-consuming
- Software homogenization via VM images
  - Unique VM image database provides single point of application deployment
  - Libraries and similar resolved once, version mismatch problems eliminated

Clouds for simplicity

- Scaling issues with grid middleware
- Different types of Scaling
  - Automatic scaling of processing power by spawning more compute-intensive VMs
  - Automatic scaling for more users using load balancing at the high-level services
  - Solutions like Amazon S3 help data scaling, but proper metadata and discovery are needed

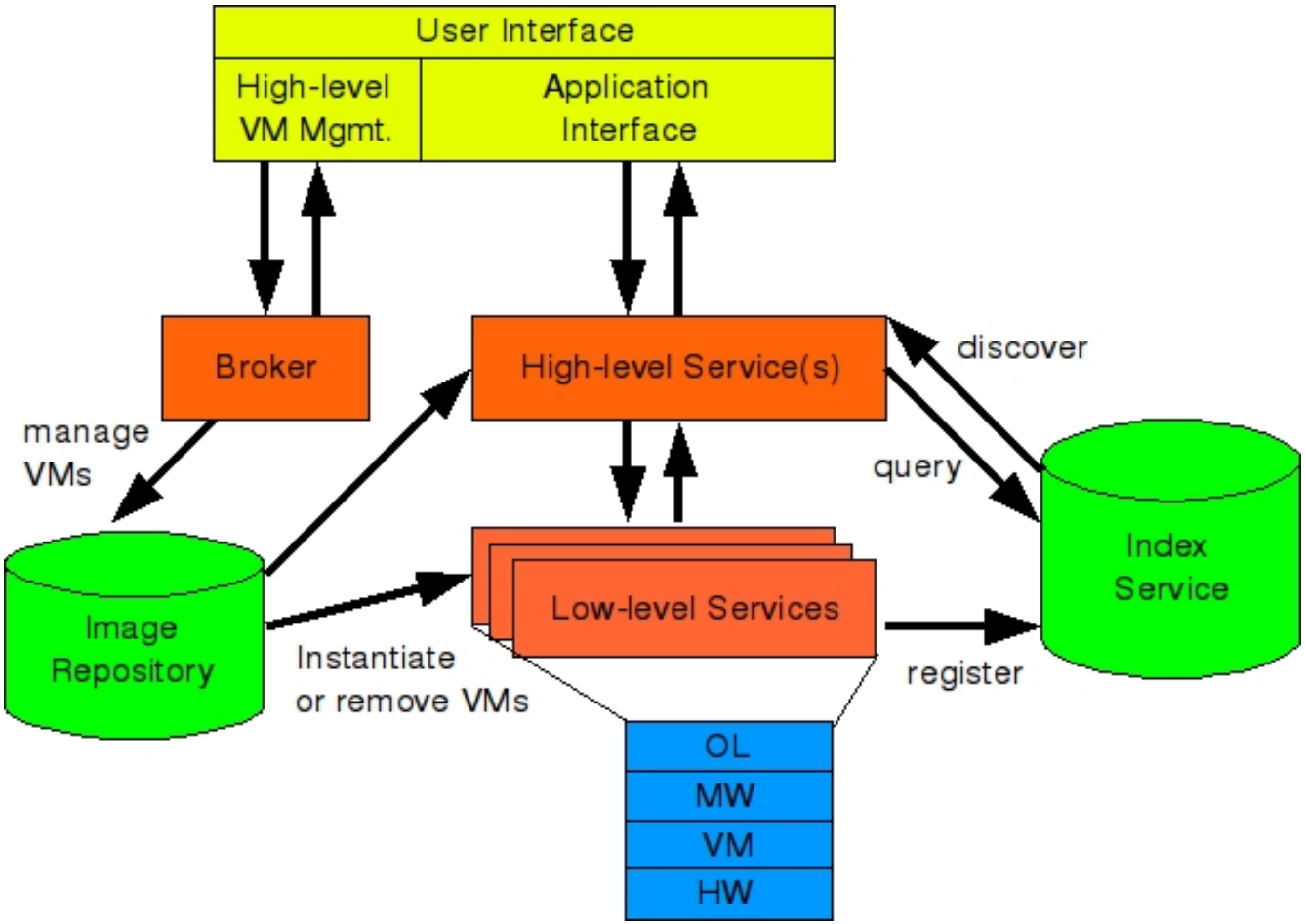
Clouds for enforcement of design

- Access to low level functionality and services
- Tighter control of access
  - No access to middleware functions except VM management
  - Deny access to the lower level services by using private networks
- Ontology usage is not ensured to access data
- Common ontology inside the Cloud
  - Can easily mantain a controlled set of data abstractions
  - A cloud can be characterized by an ontology, purpose of design
- Resources consumed on a given machine
- VM Manager controls resource usage in any VM
  - Can define limit levels of memory and disk utilization (CPU using virtual cores)

Clouds and the Master/Worker paradigm

- The paradigm:
  - One Master Service and multiple Worker Services
  - Client interacts with Master only with a high-level interface, the Workers perform the low-level operations
  - Grid middleware provides discovery of Workers by means of an Index Service (MDS in Globus)
- Under Cloud computing:
  - Workers need only be configured once
  - Seamless scaling of application by instantiating additional Workers
  - Master must use the Index Service to find Workers. Master may use the additional workers on the fly
  - Clients may know about the Workers if they wish for job monitoring

Clouds and the Master/Worker paradigm



Summary

- Cloud Computing can be used in composition with grid middleware to provide an additional layer of abstraction to address some of limitations in the grid approach
- Virtualization is key to homogenize computer resources and data
- Cloud computing may become the leading technology for data scaling, but need to consider loss of effienciency due to hypervisor
- Our future work will continue to explore applications of Cloud Computing in the context of grid computing

Thank you!

Giacomo Mc Evoy
giacomo@lncc.br