EMI Middleware in Cloud Environments

Shahbaz Memon (JUELICH), Eric Yen (ASGC), Morris Riedel (JUELICH), Mischa Salle (NIKHEF), Oscar Koeroo (NIKHEF)

EGI Technical Forum 2011, Lyon
Outline

• Objectives
• Association Models
• Outlook
Objectives

• Enable dynamic and on-demand provisioning of EMI services
• Identify EMI positioning with virtualization and cloud computing technologies used in the current DCI ecosystem
• EMI service interoperation with clouds - StratusLab
• EMI appliance based mechanism to achieve grid service on-demand scenarios

EMI is not cloud
DCI Collaboration Map as a Reference Model
Association Models

• **Model 1: Service interoperation**
  – Cloud services use EMI components to efficiently perform cloud infrastructure management functions
  – More Priority

• **Model 2: Virtual Grid Service**
  – EMI services are „packaged and configured„ ready to be deployed in virtual machines
Model-1: Service Interoperation (SI)

- We can leverage the strength of EMI in the existing virtual infrastructure management implementations
  - Production software components
  - Standards based AuthN/AuthZ mechanisms
  - Support of virtual organizations
  - Service discovery
  - Unified infrastructure messaging model
  - and much more ..
Scenario: VM run in a StratusLab cloud
**Virtual Infrastructure Management**

1. **X509-Proxy**
2. **VOMS-Server**
3. **XACML**
4. **Argus**

**Cloud Clients**

- **Administrator**
- **CLI**
- **Scientist**

**Hardware Resources**

- **Data resource**
- **HTC Resources**

**Virtual Infrastructure Management**

- **VMM** (e.g. OpenNebula, Hypervisor (e.g. XEN))
- **Image Repository** (e.g. Marketplace)
- **REST**
- **VM Image** (e.g. STAR, Amber Appliance)
- **Gateway** (e.g. OpenNebula Authentication Proxy)
- **Proprietary & OCCI & EC2**

**Cloud Clients**

- **Administrator**
- **CLI**
- **Scientist**
Sequence of actions: Starting an VM instance

1. Grid user fetches VOMS-Proxy from a VOMS-Server

2. Grid user contacts \texttt{(stratus-run-instance)} OpenNebula Authentication Proxy (OAP) using the VOMS-Proxy

3. OAP makes a XACML callout to the Argus services to know whether the user is authorized to perform this action \texttt{(stratus-run-instance)}

4. Once OAP recieves a positive response, it will forward user request to the VMM Service

5. VMM then provisions the requested VM image onto the physical resources, and returns the VMID and status to the user
User access in a federated cloud

Infrastructure Management

Life Science

Fusion

Virtual Organizations

Science Clouds

GR-Net

Juelich

ASGC

Applications
Services
Platform
VMM

Applications
Services
Platform
VMM

Applications
Services
Platform
VMM

Applications
Services
Platform
VMM

StratusLab

20/09/2011
## Model 1: Service Interoperation

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>EMI Areas</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Users with grid credentials can access the OAP (integration with StratusLab rather than re-implementation)</td>
<td>Security</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>OAP must have a fool proof central authorization system to enforce and manage service and VO level policies</td>
<td>Security</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>OAP contacts central service to authorize users intending to upload and register VM metadata to the StratusLab’s appliance repository and market place.</td>
<td>Security</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>User being a VO member can easily interact with multiple private clouds part of that VO</td>
<td>Security</td>
<td>Medium</td>
</tr>
<tr>
<td>5</td>
<td>Private cloud deployment must be able to publish all the service details in a DCI level service registry (e.g. EMI Registry)</td>
<td>Infrastructure</td>
<td>Medium</td>
</tr>
<tr>
<td>6</td>
<td>Cloud deployment should use a messaging infrastructure; EMI messaging guidelines should be considered (extend if required)</td>
<td>Infrastructure</td>
<td>Low</td>
</tr>
<tr>
<td>7</td>
<td>Persistent-disk-store must integrate with the EMI storage namespace services (DPM, d-Cache) to eliminate data naming conflicts in federated environments</td>
<td>Data</td>
<td>Low</td>
</tr>
<tr>
<td>8</td>
<td>OAP must implement a grid authentication plugin using EMI common authentication library</td>
<td>Security</td>
<td>Low</td>
</tr>
</tbody>
</table>
Model-2: Virtual Grid Services (VGS)

- Grid admins can setup a grid site over cloud resources in an automated manner
- Grid site needs to dynamically adapt the adhoc nature of virtual services,
  - Monitoring, Service discovery, Security, Accounting, and Messaging,..
Contextualization Strategies

• VM appliance is contextualized through set of contextualization parameters provided by a user (Push)

• VM appliance contacts the repository to fetch the contextualization parameters (Pull)
Contextualization: Push Model

1. Reference of VM image

VM Metadata

Grid Admin

2. Start image: Image id and Context parameters

VMM

Example:
--context='ENABLE_UNICORE=true;site-name=VDEMO-SITE-1;emi-registry-url=url; Argus-url=url'

EMI VM

3. Setup using user and default context parameters

Context Agent
Contextualization: Pull Model

1. Publish context parameters
2. Reference of VM image
3. Start image: Image id
4. Image provisioning on a physical node
5. Context agent fetches context parameters
## Model 2: Virtual Grid Services (VGS)

<table>
<thead>
<tr>
<th>No.</th>
<th>Description</th>
<th>EMI Areas</th>
<th>Priority</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Create pre-configured EMI based virtual appliances which are preferred by StratusLab and EGI</td>
<td>Compute, Data, Information, Security</td>
<td>High</td>
</tr>
<tr>
<td>2</td>
<td>Develop contextualization agents to automate a VGS setup and configuration</td>
<td>Compute</td>
<td>High</td>
</tr>
<tr>
<td>3</td>
<td>VGS must adequately react to the VM lifecycle functions (Start-Running-Stop and Snapshotting)</td>
<td>Compute</td>
<td>Medium</td>
</tr>
<tr>
<td>4</td>
<td>Provision/de-provision of virtual EMI-SEs and the backend raw storage</td>
<td>Data</td>
<td>Low</td>
</tr>
<tr>
<td>5</td>
<td>Support of virtual EMI-CE with 4</td>
<td>Data, Compute</td>
<td>Low</td>
</tr>
<tr>
<td>6</td>
<td>VGS must adhere to the EMI messaging guidelines</td>
<td>Infrastructure</td>
<td>Low</td>
</tr>
<tr>
<td>7</td>
<td>VGS must be able to publish Nagios probes to a monitoring service already used by a grid site</td>
<td>Infrastructure</td>
<td>Low</td>
</tr>
<tr>
<td>8</td>
<td>EMI infrastructure services in a DCI must ensure a seamless integration of virtual and non-virtual services</td>
<td>Infrastructure</td>
<td>Low</td>
</tr>
<tr>
<td>9</td>
<td>VGS must publish resource accounting information in a format adopted by EMI resource accounting teams (e.g. OGF UR)</td>
<td>Infrastructure</td>
<td>Low</td>
</tr>
</tbody>
</table>
Scenario: Job execution in a cloud
Job Execution on Virtual OGSA-BES

1. Create VM instance via StratusLab proprietary API or EC2 (not OCCI yet) using the StratusLab CLI client
   - Specify parameters like URI (VM Image Appliance Reference) or Appliance ID, disk space, compute image type (small, medium, large)
   - Response from VMM Server is a Vmid (not appliance ID) and IP address of the VM and its hostname
   - Vmid (is kind of an Grid job execution ID, BES activity ID)

2. VMM Server is looking up the Image Appliance based on the URI

3. Schedule and execute the Image Appliance specified by the URI on the Hypervisor

4. Specified URI Appliance is up and running within the Hypervisor
   - Takes 2-3 minutes until the VM Image really runs

5. VM Image Appliance is running on a HTC resource: Hypervisor installed on each of the HTC Resource cores

6. OGSA-BES is instantiated inside the running VM image appliance and is accessible by end-users with clients
   - Living duration of this service depends, might be days, weeks (not as static as forever as used to be in previous EGI infrastructures)

7. No automatism yet about the correct endpoint URI and of OGSA-BES to be transferred to the EMI Registry Client

8. OGSA-BES endpoint information is put inside the non-virtualized EMI Registry and is exposed, e.g. https://hostvirtualized.com:8080/BES

9. End-user using its scientific client tool (with integrated EMI Registry Client) in order to obtain the OGSA-BES endpoint for job submission

10. Scientist is using an OGSA-BES client in his specific client tool and the obtained URI to contact the virtualized OGSA-BES endpoint submitting a JSDL

11. Specified application In JSDL is running on the VM instance (same where the OGSA-BES service is installed on)
EMI – StratusLab in DCI
Outlook

• More scrutinize and prioritize the SI and VGS usecases in collaboration with StratusLab and EMI functional areas

• Evolve technical objectives, implementation plan, and timelines
Questions?
Acknowledgements

- Vangelis Floros, Charles Loomis (StratusLab)
- Michel Drescher (EGI)
Thank you!

EMI is partially funded by the European Commission under Grant Agreement RI-261611