Building Secure and Transparent Inter-Cloud Infrastructure for Scientific Applications

Yoshio Tanaka
Information Technology Research Institute, National Institute of AIST, Japan
We could gain a lot of insights through Grid experiments!!

• Building Grid
  – Communities
    • APGrid PMA (2004~), IGTF(2005~)

• Using Grid
  – Programming
    • Ninf-G: GridRPC-based programming middleware (1994~)
  – Applications
    • Hybrid QM/MD Simulation (2004~)
    • Grid FMO (2005~)

• Experiments
  – Run Grid applications on large-scale Grid infrastructure
    • PRAGMA + OSG (2007)
Multi-Scale Simulation using GridRPC+MPI

- Simulation of Hydrogen Diffusion Process of Gamma-alumina
  - Executed on the Pan-pacific Grid Testbed (~ 1000 CPUs)
  - More than 5000 QM simulations during 50 hours
  - Implemented based on GridRPC + MPI Programming model.
Grid-enabled SIMOX Simulation on Japan-US Grid at SC2005

- Experiment Time: 18.97 days
- Simulation steps: 270 (~ 54 fs)
- Total: 150K CPU x hour
Empowered by Grid technology, large scale fragment molecular orbital (FMO) calculations were conducted making use of computational resources on the Grid. The resources were collected interoperably from PRAGMA and OSG domains, via ordinary queue systems on each machine. Without special arrangements, a series of the calculations were carried out for about 3 months using maximum of 240 CPUs.

- Full electron calculation of bio molecules by using FMO method
- Long-term calculations supported by fault-tolerance
- Utilize vast computational resources on the Grid
- Flexible resource management to exploit ad-hoc available resources
Grid experiments were exciting and successful, but...

- Using heterogeneous and distributed resources was not easy.
- Heterogeneity did exist in various places
  - OS, Library version
  - in more details of the system configuration
    - Configuration of queuing systems
      - e.g. max wall clock time
    - disk quota limit
    - Firewall configuration
- We need to install and test applications at each site one by one.
  - We should not expect end users to do the same!
Goals of and approaches by PRAGMA

• Enable Specialized Applications to run easily on distributed resources
  – Build once, run everywhere!!

• Investigate Virtualization as a practical mechanism
  – Supporting Multiple VM Infrastructures (Xen, KVM, OpenNebula, Rocks, WebOS, EC2)

• Share VM images in PRAGMA VM repository so that we can boot our application VMs at any site by any PRAGMA colleagues.
  – Discussed in PRAGMA 20 workshop @ HK, March 3rd and 4th, 2011, 1 week before big earthquake in Japan...
2011 Tohoku Earthquake changed our R&D environments
ASTER data: NASA → ERSDAC → AIST
PALSAR data: JAXA → ERSDAC → AIST
(processing, WMS, portal site, and data providing by AIST)
Data Flow and Services from March 11 till April 20

- **ASTER data:** NASA → ERSDAC → (AIST) →
- **PALSAR data:** JAXA → ERSDAC → (AIST) →
  (processing and WMS by Orkney, portal site by Google)
Data Flow and Services from April 21

- **ASTER data**: NASA $\rightarrow$ ERSDAC $\rightarrow$ (AIST) $\rightarrow$
- **PALSAR data**: JAXA $\rightarrow$ ERSDAC $\rightarrow$ (AIST) $\rightarrow$ (processing by NCHC, SDSC, and OCCI, WMS by NCHC, portal site by Google)
Insights

• Fortunately, we already had VM images for satellite data processing.
  – We have prepared for using cloud.
• Need to make it routine use!
• PRAGMA members had disasters/accidents.
  – Japan earthquake
  – Thailand flooding
  – California power outage
• PRAGMA members has common interests/needs to build a sustainable infrastructure which could be used to support each other in case of emergency.
  – We accelerated the development/deployment of PRAGMA Cloud.
PRAGMA Grid v.s. PRAGMA Cloud

• PRAGMA Grid
  – Each site
    • Installs common software stack (e.g. Globus, Condor)
      – It’s not easy to synchronize version up…
    • If the application needs additional libraries and configuration, etc., they must be installed / configured.
  – Application drivers
    • Install, test, and run applications at each site.

• PRAGMA Cloud
  – Each site
    • Deploy cloud hosting environments.
  – Application drivers
    • Build VM image for their applications.
    • Boot the VM at each site.
<table>
<thead>
<tr>
<th>Institution</th>
<th>Region</th>
<th>Host Name</th>
<th>nodes</th>
<th>CPUs</th>
<th>Memory (GB)</th>
<th>Disk Space (TB)</th>
<th>VM Platform</th>
<th>VM Manager</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIST</td>
<td>Japan</td>
<td>pragma-v</td>
<td>32</td>
<td>64</td>
<td>768</td>
<td>600</td>
<td>KVM</td>
<td>OpenNebula</td>
</tr>
<tr>
<td>ASTI</td>
<td>Philippines</td>
<td>one</td>
<td>10</td>
<td>80</td>
<td>160</td>
<td>5</td>
<td>Xen</td>
<td>OpenNebula</td>
</tr>
<tr>
<td>CNIC</td>
<td>China</td>
<td>cluster</td>
<td>1</td>
<td>2</td>
<td>128</td>
<td>2</td>
<td>Xen</td>
<td>Rocks 5.4.3</td>
</tr>
<tr>
<td>HKU</td>
<td>Hong Kong</td>
<td>hpcrocks</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>Xen</td>
<td>Rocks 5.4.3</td>
</tr>
<tr>
<td>IU</td>
<td>USA</td>
<td>pragma</td>
<td>8</td>
<td>16</td>
<td>128</td>
<td>12</td>
<td>Xen</td>
<td>Rocks 5.4</td>
</tr>
<tr>
<td>JLU</td>
<td>China</td>
<td>gfarm</td>
<td>1</td>
<td>4</td>
<td>96</td>
<td>1</td>
<td>Xen</td>
<td>Rocks 5.4.3</td>
</tr>
<tr>
<td>LZU</td>
<td>China</td>
<td>vm</td>
<td>1</td>
<td>8</td>
<td>16</td>
<td>1</td>
<td>KVM</td>
<td>Rocks 5.4</td>
</tr>
<tr>
<td>MIMOS</td>
<td>Malaysia</td>
<td>vm</td>
<td>6</td>
<td>12</td>
<td>192</td>
<td>1</td>
<td>KVM</td>
<td>Eucalyptus</td>
</tr>
<tr>
<td>NCHC</td>
<td>Taiwan</td>
<td>Snowfox</td>
<td>8</td>
<td>64</td>
<td>128</td>
<td>2</td>
<td>KVM</td>
<td>OpenNebula</td>
</tr>
<tr>
<td>OSAKAU</td>
<td>Japan</td>
<td>cider</td>
<td>3</td>
<td>24</td>
<td>34</td>
<td>3</td>
<td>Xen</td>
<td>Rocks 5.4</td>
</tr>
<tr>
<td>SDSC</td>
<td>USA</td>
<td>fiji</td>
<td>18</td>
<td>25</td>
<td>380</td>
<td>10</td>
<td>Xen</td>
<td>Rocks 5.4</td>
</tr>
<tr>
<td>SDSC</td>
<td>USA</td>
<td>rockstar</td>
<td>32</td>
<td>64</td>
<td>512</td>
<td>16</td>
<td>Xen</td>
<td>Rocks 5.4.3</td>
</tr>
<tr>
<td>SDSC</td>
<td>USA</td>
<td>pragma-kvm</td>
<td>1</td>
<td>4</td>
<td>32</td>
<td>1</td>
<td>KVM</td>
<td>Rocks 5.4</td>
</tr>
<tr>
<td>UoHyd</td>
<td>India</td>
<td>venus</td>
<td>4</td>
<td>4</td>
<td>16</td>
<td>4</td>
<td>Xen</td>
<td>Eucalyptus</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td></td>
<td>12</td>
<td>8</td>
<td>14</td>
<td>128</td>
<td></td>
<td>374</td>
</tr>
</tbody>
</table>
Deploy Three Different Software Stacks on the PRAGMA Cloud

• QuiQuake
  – Simulator of ground motion map when earthquake occurs
  – Invoked when big earthquake occurs

• HotSpot
  – Find high temperature area from Satellite
  – Run daily basis (when ASTER data arrives from NASA)

• WMS server
  – Provides satellite images via WMS protocol
  – Run daily basis, but the number of requests is not stable.

All these applications run as Condor workers
Gfarm is used for sharing VM images

PRAGMA VM REPOSITORY

Gfarm Client

Gfarm file server

Gfarm meta-server

Gfarm file server

Gfarm file server

Gfarm file server

Gfarm Client

Geogrid + Bloss

Fmotif

HotSpot

QuickQuake

vmdb.txt
Put all together

Store VM images in Gfarm systems
Run vm-deploy scripts at PRAGMA Sites
Copy VM images on Demand from gFarm
Modify/start VM instances at PRAGMA sites
Manage jobs with Condor

SDSC (USA)
Rocks Xen

NCHC (Taiwan)
OpenNebula KVM

LZU (China)
Rocks KVM

AIST (Japan)
OpenNebula Xen

IU (USA)
Rocks Xen

Osaka (Japan)
Rocks Xen

= VM deploy Script
= Grid Farm Client
= Grid Farm Server
What are the Essential Steps

1. AIST/GEO Grid creates their VM image
2. Image made available in “centralized” storage (currently Gfarm is used)
3. PRAGMA sites copy GEO Grid images to local clouds
   1. Assign IP addresses
   2. What happens if image is in KVM and site is Xen?
4. Modified images are booted
5. GEO Grid infrastructure now ready to use
Basic Operation

• VM image authored locally, uploaded into VM-image repository (Gfarm from U. Tsukuba)

• At local sites:
  – Image copied from repository
  – Local copy modified (automatic) to run on specific infrastructure
  – Local copy booted

• For running in EC2, adapted methods automated in Rocks to modify, bundle, and upload after local copy to UCSD.
Cloud Sites Integrated in GEO Grid Execution Pool
Next step: Network virtualization

• Some motivations
  – Need to change configuration of firewall and condor master when a new VM launches
  – Network isolation for security
  – Provide transparent view for users.
OpenFlow

A centralized programmable remote controller dictates the forwarding behavior to multiple OpenFlow switches. This architecture separating forwarding plane from control plane allows flexible management to network operator.

Utilized software tools for this demo:

- **Trema** ([http://trema.github.com/trema/](http://trema.github.com/trema/))
  A framework for developing OpenFlow controllers

- **Open vSwitch** ([http://openvswitch.org/](http://openvswitch.org/))
  A software based implementation of OpenFlow switch

---

Slide by courtesy of Prof. Kohei Ichikawa, NAIST
Openflow network environment

Virtual network slice A

Virtual network slice B

Openflow Controller
Trema
(Sliceable routing switch)

GRE

Osaka Univ.

AIST

UCSD

Openflow network

GRE

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM

VM
Comparison between the past and the present approach

The past

Condor Master
- Register the new launched VM's IP to the condor configuration
- Assign a new global IP for each VM launching
- Configure firewall policy to pass the communication for Gfarm, condor and so on

Condor Master
- Register the new launched VM's IP to the condor configuration
- Assign a new global IP for each VM launching
- Configure firewall policy to pass the communication for Gfarm, condor and so on

Site A admin
Site B admin
Site C admin

each VM

The present

Condor Master
- Register the IP to the condor pool automatically
- Dedicated isolated virtual L2 network
- Request IP via DHCP

Condor Master
- Register the IP to the condor pool automatically
- Dedicated isolated virtual L2 network
- Request IP via DHCP

Site A admin
Site B admin
Site C admin

Slide by courtesy of Prof. Kohei Ichikawa, NAIST
Summary

• We learned a lot through Grid experiments.

• Migrating from Grid to Cloud
  – Virtualization technologies is useful for making distributed infrastructure easy to use.

• Still have many research issues.
  – Data
  – Network virtualization
  – Resource managements
  – Security
  – Making it routine-use
Acknowledgements

• The most of the work was done in collaboration with PRAGMA members, especially UCSD, NCHC, Osaka U.
• GEO Grid Disaster Response work was supported by domestic and international organizations.

• NARL|NCHC
• UCSD|SDSC
• Open Cloud Consortium
• ERSDAC
• Orkney
• NTT-data-CCS
• CTC
• Université Lille 1 (Science and technology)
• ITT