Grid on Demand

Willem Toorop and Alain van Hoof

{wtoorop,ahoof}@os3.nl

June 30, 2010
Can Grid computing be offered as a Cloud service and what type of applications benefit from this?
The Grid and The Cloud: Sharing Resources

Sharing Resources to minimizing idle resources

<table>
<thead>
<tr>
<th>Resource</th>
<th>Capacity</th>
<th>Goods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Flops</td>
<td>Compute</td>
</tr>
<tr>
<td>Hard-disk memory</td>
<td>Bytes</td>
<td>Storage</td>
</tr>
<tr>
<td>Network</td>
<td>Bytes/sec</td>
<td>Communication Bandwidth</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Grid</th>
<th>Cloud (IaaS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution</td>
<td>By Collaboration</td>
</tr>
<tr>
<td></td>
<td>By Cloud Provider</td>
</tr>
<tr>
<td>Resource Type</td>
<td>Heterogeneous</td>
</tr>
<tr>
<td></td>
<td>Homogeneous</td>
</tr>
<tr>
<td>Abstraction</td>
<td>Middleware (open standards)</td>
</tr>
<tr>
<td></td>
<td>Virtualization</td>
</tr>
<tr>
<td>Sharing Guarantee</td>
<td>Security infrastructure</td>
</tr>
<tr>
<td></td>
<td>Financial infra</td>
</tr>
</tbody>
</table>
The Grid and The Cloud: Dividing the Resources

Willem Toorop and Alain van Hoof (OS3)
Using the elasticity and scalability of the Cloud and providing a Grid interface, current e-Science (computationally intensive science) applications can use a **Grid on Demand** when there is a high demand for resources for a short period of time.

**Example**

UrbanFlood (http://urbanflood.eu) wants to calculate flooding predictions when actual flooding happens, Grid on Demand can provide this.
Grid Interface: Globus Toolkit 4.2 (open standard, widely used)
Queue Manager (Distribution of jobs to nodes): Torque 2.6.2
Amazon EC2 Cloud Provider (de-facto standard with open API)
What makes the Grid, the Grid

Typical Grid usage example spanning multiple organizations
Proxy certificates for delegation

The Grid is a collaboration between multiple organizations.

- Grid users need to be authorized to use remote resources
- Those resources should be authorized to perform actions on behalf of the Grid user

Proxy certificates enable delegation of authority
Grid user

create proxy

Scheduler

create cert. request

compute X

sign this

Okay

Compute

compute X
Virtual Organizations are based on delegation by proxy certificates.
Proxy certificates allow for dynamic resource allocation,

But the hosts still need a CA signed certificate.

This is a problem when hostname’s are assigned dynamically!
Proxy certificates allow for dynamic resource allocation,

But the hosts still need a CA signed certificate.

This is a problem when hostname’s are assigned dynamically!

Fixed CA in image?
Proxy certificates allow for dynamic resource allocation,

But the hosts still need a CA signed certificate.

This is a problem when hostname’s are assigned dynamically!

Fixed CA in image?

Dynamic CA on every Launch?
**Easy Image management with Amazon Elastic Block Store**

Life cycle of Amazon Machine Images using Elastic Block Storage

\[ A = \text{Amazon Machine Image}, \ S = \text{Snapshot}, \ I = \text{Instance}, \ V = \text{Volume}, \ H = \text{Hostname} \]
EBS for configuration stages

\[ (A, S) \]

\[ V_S \leftarrow S \]

**Running** \( (I_A, V_S, H) \)

configure

**Running** \( (I_A, V_S, H) \)

provide Amazon Access Credentials

configure

\[ (A', S') \]

\[ V_S \leftarrow S' \]

**Stopped** \( (I_A, V_S) \)

start

**Stopped** \( (I_A, V_S) \)

shutdown for reuse

**Stopped** \( (I_A, V_S) \)

shutdown for sharing

**Stopped** \( (I_A', V_S) \)

shutdown for reuse

**Stopped** \( (I_A'', V_S) \)

shutdown for sharing

**Stopped** \( (I_A''', V_S) \)

**create AMI** \( A' \)

\[ S' \leftarrow V_S \]

**create AMI** \( A'' \)

\[ S'' \leftarrow V_S \]

**create AMI** \( A''' \)

\[ S''' \leftarrow V_S \]

**launch** instance \( I_A \)

\[ V_S \leftarrow S \]

**launch** instance \( I_A' \)

\[ V_S \leftarrow S' \]

**launch** instance \( I_A'' \)

\[ V_S \leftarrow S'' \]

**launch** instance \( I_A''' \)

\[ V_S \leftarrow S''' \]
Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.
Web Interface

Launching Grid on Demand

Willem Toorop and Alain van Hoof (OS3)

To access your instance using any SSH client

1. Open the SSH client of your choice (e.g., PuTTY, terminal).
2. Locate your private key file, default.pem
3. Use `chmod` to make sure your key file isn't publicly viewable, ssh won't work otherwise:
   `chmod 400 default.pem`

Example

Enter the following command line:

```
ssh -i default.pem root@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com
```

Status: running 021450451553

© 2008 - 2009, Amazon Web Services LLC or its affiliates. All rights reserved.

Feedback Support Privacy Policy Terms of Use
willem@willem-laptop:~$ ssh -i default.pem root@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com
Please login in as griduser to immediately use the Grid on Demand:

    ssh -i default.pem -L 8080:localhost:80 \
    griduser@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com

or login as ubuntu for maintenance:

    ssh -i default.pem -L 8080:localhost:80 \
    ubuntu@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com

Then Grid on Demand can be configured and monitored via:

    http://localhost:8080/

willem@willem-laptop:~$
Web Interface
Launching Grid on Demand

Willem Toorop and Alain van Hoof (OS3)

Grid on Demand

June 30, 2010 17 / 39

griduser@ec2-79-125-48-213:~ 84x24

willem@willem-laptop:~$ ssh -i default.pem root@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com
Please login in as griduser to intedimly use the Grid on Demand:

    ssh -i default.pem -L 8080:localhost:80 \
        griduser@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com

or login as ubuntu for maintenance:

    ssh -i default.pem -L 8080:localhost:80 \
        ubuntu@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com

Then Grid on Demand can be configured and monitored via:

    http://localhost:8080/


willem@willem-laptop:~$ ssh -i default.pem -L 8080:localhost:80 \
>    griduser@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com
Last login: Mon Jun 28 20:16:08 2010 from happusxs4all.nl
griduser@ec2-79-125-48-213:~$
To be able to spawn new compute nodes on demand, Amazon Access Credentials are needed. They can be found [here](#).
griduser@ec2-79-125-48-213:~ 84x24

or login as ubuntu for maintenance:

    ssh -i default.pem -L 8080:localhost:80 \\n    ubuntu@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com

Then Grid on Demand can be configured and monitored via:

    http://localhost:8080/

willem@willem-laptop:~$ ssh -i default.pem -L 8080:localhost:80 \\n>    griduser@ec2-79-125-48-213.eu-west-1.compute.amazonaws.com
Last login: Mon Jun 28 20:16:08 2010 from happus.xs4all.nl
griduser@ec2-79-125-48-213:~$ grid-proxy-info
subject  : /O=Grid/OU=GlobusTest/OU=GridOnDemand/CN=Grid User/CN=1418860555
issuer   : /O=Grid/OU=GlobusTest/OU=GridOnDemand/CN=Grid User
identity : /O=Grid/OU=GlobusTest/OU=GridOnDemand/CN=Grid User
type     : RFC 3820 compliant impersonation proxy
strength  : 512 bits
path     : /tmp/x509up_u1003
timeleft : 11:57:02
griduser@ec2-79-125-48-213:~$ globus-job-run localhost /usr/bin/whoami
torqueuser
griduser@ec2-79-125-48-213:~$
Access Credentials

There are three types of access credentials used to authenticate your requests to AWS services: (a) access keys, (b) X.509 certificates, and (c) key pairs. Each access credential type is explained below.

Access Keys

Use access keys to make secure REST or Query protocol requests to any AWS service API. We create one for you when your account is created — see your access key below.

Your Access Keys

<table>
<thead>
<tr>
<th>Created</th>
<th>Access Key ID</th>
<th>Secret Access Key</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>January 31, 2010</td>
<td>AKIAJPB6XI2Z5J6RLA</td>
<td>Show</td>
<td>Active (Make Inactive)</td>
</tr>
<tr>
<td>February 15, 2010</td>
<td>AKIAJJQSOYC3ZWISH3Q</td>
<td>Show</td>
<td>Active (Make Inactive)</td>
</tr>
</tbody>
</table>

For your protection, you should never share your secret access keys with anyone. In addition, industry best practices recommends frequent key rotation.

Learn more about Access Keys

Sign-In Credentials

To sign in to AWS web sites and applications, AWS requires your Amazon e-mail address and password. Additionally, AWS supports the AWS Multi-Factor Authentication option. Each sign-in credential is explained below.

Amazon E-mail Address and Password

To sign in to secure pages on the AWS website, the AWS Management Console, the AWS Discussion Forums and AWS Premium Support site, you need to provide your Amazon e-mail address and password.
willem@diglett:~$ grid-proxy-init
Your identity: /O=dutchgrid/O=users/O=uva/OU=wins/CN=Willem Toorop
Creating proxy ............................... Done
Your proxy is valid until: Tue Jun 29 11:50:33 2010
willem@diglett:~$
willem@diglett:~$ grid-proxy-init
Your identity: /O=dutchgrid/O=users/O=uva/OU=wins/CN=Willem Toorop
Creating proxy ........................................ Done
Your proxy is valid until: Tue Jun 29 11:55:08 2010
willem@diglett:~$ tar xzvf gridondemand-ca-certs.tgz
.globus/certificates/
.globus/certificates/globus-host-ssl.conf.5c703074
.globus/certificates/grid-security.conf.5c703074
.globus/certificates/globus-user-ssl.conf.5c703074
.globus/certificates/5c703074.signing_policy
.globus/certificates/5c703074.0
willem@diglett:~$ for i in {1..10} 
> do
>   globus-job-submit ec2-79-125-48-213.eu-west-1.compute.amazonaws.com \
>   /bin/sleep `expr $i \* 5` &
> sleep $i
> done
Web Interface

Elasticity of the cluster

Willem Toorop and Alain van Hoof (OS3)
Web Interface

Elasticity of the cluster

Willem Toorop and Alain van Hoof (OS3)
Thank you for using Grid on Demand

You have spent: $1.14 = €0.92

This instance is now shutting down for reuse

When you have made adaptations you may make a new AMI from instance: i-64d34c13
To test the possibilities and performance of Grid on Demand a Biomedical Application and a workflow manager are used.
"Real use" and Performance test

To test the possibilities and performance of Grid on Demand a Biomedical Application and a workflow manager are used.

Biomedical Application: WAVE

Parallel (MonteCarlo simulation) application
“Real use” and Performance test

To test the possibilities and performance of Grid on Demand a Biomedical Application and a workflow manager are used.

Biomedical Application: WAVE
Parallel (MonteCarlo simulation) application

Workflow Manager
- A (graphical) tool to assist complex e-Science application creation
- Creates a series of job submissions
- WS-VLAM created by UvA SNE Group
- Connects to Globus Grid Interface
Connecting the Grid on Demand to a Workflow Manager

- Head Node
  - OS: Ubuntu 10.04
  - Grid: Globus 4.2.1
  - PBS: Torque 2.6.8
- Compute Nodes

WS-VLAM

WAVE Biomedical Compute Application

Willem Toorop and Alain van Hoof (OS3)
Preliminary performance results: total execution time

Grid on demand compared to a 32 node physical UvA cluster (DAS3) with Globus Grid interface

<table>
<thead>
<tr>
<th>Grid</th>
<th>Jobs</th>
<th>Total execution time</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAS3</td>
<td>28</td>
<td>00:42:58</td>
</tr>
<tr>
<td>Grid on Demand</td>
<td>28</td>
<td>00:46:22</td>
</tr>
<tr>
<td>DAS3</td>
<td>98</td>
<td>02:46:36</td>
</tr>
<tr>
<td>Grid on Demand</td>
<td>98</td>
<td>00:52:26</td>
</tr>
</tbody>
</table>
Preliminary performance results of job pending and running time: 30 jobs
Preliminary performance results of job pending and running time: 100 jobs
Conclusion

- Grid on Demand can offer Grid as a Cloud service
- Parallel applications run/scale very well on Grid on Demand
- Grid on Demand can be a solution for Urgent Computing demands
Questions