FutureGrid Tutorial @
CloudCom 2010

Indianapolis, Thursday Dec 2, 2010, 4:30-5:00pm
laszewski@gmail.com

Gregor von Laszewski, Greg Pike, Archit Kulshrestha,
Andrew Younge, Fugang Wang,
and the rest of the FG Team

Community Grids Lab
Pervasive Technology Institute
Indiana University
Bloomington, IN 47408
laszewski@gmail.com

http://www.futuregrid.org

This document was developed with support from the National Science Foundation (NSF) under Grant No. 0910812.
Acknowledgement

- Slides are developed by the team. We like to acknowledge all FG team members for their help in preparing these slides.
- This document was developed with support from the National Science Foundation (NSF) under Grant No. 0910812.
Overview

- Introduction to FutureGrid
- Support
- Phase I FutureGrid Services
  - HPC on FutureGrid
  - Eucalyptus on FutureGrid
  - Nimbus on FutureGrid

(Gregor 15 min)
(Gregor 5 minutes)
(Pike 30min)
(Archit 29min)
(Archit 1 min)
Outline (cont. if time permits)

● Phase II FutureGrid Services
  o Image Management
    ■ Repository
    ■ Generation & Management
  o Dynamic Provisioning
  o Portal

  (Gregor)
  (Andrew)
  (Gregor)
  (Gregor)
OVERVIEW OF FG
FutureGrid will provide an experimental testbed with a wide variety of computing services to its users.

The testbed provides to its users:

- A rich development and testing platform for middleware and application users allowing comparisons in functionality and performance.
- A variety of environments, many be instantiated dynamically, on demand. Available resources include, VMs, cloud, grid systems …
- The ability to reproduce experiments at a later time (an experiment is the basic unit of work on the FutureGrid).
- A rich education and teaching platform for advanced cyberinfrastructure
- The ability to collaborate with the US industry on research projects.
**HW Resources at:** Indiana University, SDSC, UC/ANL, TACC, University of Florida, Purdue,

**Software Partners:** USC ISI, University of Tennessee Knoxville, University of Virginia, Technische Universität Dresden

However, users of FG do not have to be from these partner organizations. Furthermore, we hope that new organizations in academia and industry can partner with the project in the future.
FutureGrid has dedicated network (except to TACC) and a network fault and delay generator.

Can isolate experiments on request; IU runs Network for NLR/Internet2

(Many) additional partner machines will run FutureGrid software and be supported (but allocated in specialized ways)

(*) IU machines share same storage; (**) Shared memory and GPU Cluster in year 2
### Storage and Interconnect Hardware

<table>
<thead>
<tr>
<th>System Type</th>
<th>Capacity (TB)</th>
<th>File System</th>
<th>Site</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDN 9550 (Data Capacitor)</td>
<td>339</td>
<td>Lustre</td>
<td>IU</td>
<td>Existing System</td>
</tr>
<tr>
<td>DDN 6620</td>
<td>120</td>
<td>GPFS</td>
<td>UC</td>
<td>New System</td>
</tr>
<tr>
<td>SunFire x4170</td>
<td>72</td>
<td>Lustre/PVFS</td>
<td>SDSC</td>
<td>New System</td>
</tr>
<tr>
<td>Dell MD3000</td>
<td>30</td>
<td>NFS</td>
<td>TACC</td>
<td>New System</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Machine</th>
<th>Name</th>
<th>Internal Network</th>
</tr>
</thead>
<tbody>
<tr>
<td>IU Cray</td>
<td>xray</td>
<td>Cray 2D Torus SeaStar</td>
</tr>
<tr>
<td>IU iDataPlex</td>
<td>india</td>
<td>DDR IB, QLogic switch with Mellanox ConnectX adapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blade Network Technologies &amp; Force10 Ethernet switches</td>
</tr>
<tr>
<td>SDSC iDataPlex</td>
<td>sierra</td>
<td>DDR IB, Cisco switch with Mellanox ConnectX adapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Juniper Ethernet switches</td>
</tr>
<tr>
<td>UC iDataPlex</td>
<td>hotel</td>
<td>DDR IB, QLogic switch with Mellanox ConnectX adapters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Blade Network Technologies &amp; Juniper switches</td>
</tr>
<tr>
<td>UF iDataPlex</td>
<td>foxtrot</td>
<td>Gigabit Ethernet only (Blade Network Technologies; Force10 switches)</td>
</tr>
<tr>
<td>TACC Dell</td>
<td>alamo</td>
<td>QDR IB, Mellanox switches and adapters Dell Ethernet</td>
</tr>
</tbody>
</table>
Network Impairment Device

- Spirent XGEM Network Impairments Simulator for jitter, errors, delay, etc
- Full Bidirectional 10G w/64 byte packets
- up to 15 seconds introduced delay (in 16ns increments)
- 0-100% introduced packet loss in .0001% increments
- Packet manipulation in first 2000 bytes
- up to 16k frame size
- TCL for scripting, HTML for manual configuration
- Need exciting proposals to use!!
Support

- Web Site
- Portal (under development)
- Manual
- Expert team (see the manual)
  - each project will get an expert assigned
  - helps with questions, interfacing to other experts
  - helps contributing to the manual
  - staffs forums and points to answers in the manual
- help@futuregrid.org
- Knowledge Base
- Job Opening
FutureGrid Phase I Services

HPC
Eucalyptus
Nimbus
HPC on FutureGrid

Gregory G. Pike

(30 min)
FutureGrid Systems Manager

ggpike@gmail.com
A brief overview

- FutureGrid as a testbed
  - Varied resources with varied capabilities
  - Support for grid, cloud, HPC, next?
  - Continually evolving
  - Sometimes breaks in strange and unusual ways

- FutureGrid as an experiment
  - We’re learning as well
  - Adapting the environment to meet user needs
Getting Started

- Getting an account
  - Generating an SSH key pair
- Logging in
- Setting up your environment
- Writing a job script
- Looking at the job queue
- Why won’t my job run?
- Getting your job to run sooner

- http://www.futuregrid.org/
LotR principle
- If you have an account on one resource, you have an account on all resources
- It’s possible that your account may not be active on a particular resource
- Send email to help@futuregrid.org if you can’t connect to a resource
- Check the outage form to make sure the resource is not in maintenance
- http://www.futuregrid.org/status
Getting an account

- Apply through the web form
  - Make sure your email address and telephone number are correct
  - No passwords, only SSH keys used for login
  - Include the public portion of your SSH key!

- New account management is coming soon
  - Account creation may take an inordinate amount of time
  - If it’s been longer than a week, send email
Generating an SSH key pair

- For Mac or Linux users
  - `ssh-keygen -t rsa`
  - Copy `~/.ssh/id_rsa.pub` to the web form
  - For new keys, email `~/.ssh/id_rsa.pub` to help@futuregrid.org

- For Windows users, this is more difficult
  - Download `putty.exe` and `puttygen.exe`
  - Puttygen is used to generate an SSH key pair
    - Run `puttygen` and click “Generate”
    - The public portion of your key is in the box labeled “SSH key for pasting into OpenSSH authorized_keys file”
You must be logging in from a machine that has your SSH key.

Use the following command:

```
ssh username@india.futuregrid.org
```

Substitute your FutureGrid account for `username`.
Setting up your environment

- Modules is used to manage your $PATH and other environment variables

- A few common module commands
  - `module avail` – lists all available modules
  - `module list` – lists all loaded modules
  - `module load` – adds a module to your environment
  - `module unload` – removes a module from your environment
  - `module clear` – removes all modules from your environment
A job script has PBS directives followed by the commands to run your job

```
#!/bin/bash
#PBS -N testjob
#PBS -l nodes=1:ppn=8
#PBS -q batch
#PBS -M username@example.com
##PBS -o testjob.out
#PBS -j oe
#
sleep 60
hostname
echo $PBS_NODEFILE
cat $PBS_NODEFILE
sleep 60
```
Writing a job script

- Use the `qsub` command to submit your job
  - `qsub testjob.pbs`
- Use the `qstat` command to check your job

```
> qsub testjob.pbs
25265.i136
> qstat
Job id    Name         User  Time Use S Queue
---------- ------------ ----- -------- - ----- ------
25264.i136 sub27988.sub inca 00:00:00 C batch
25265.i136 testjob   gpike 0    R batch
[139]i136::gpike>
```
Both `qstat` and `showq` can be used to show what’s running on the system.

- The `showq` command gives nicer output.
- The `pbsnodes` command will list all nodes and details about each node.
- The `checknode` command will give extensive details about a particular node.
Why won’t my jobs run?

- Two common reasons:
  - The cluster is full and your job is waiting for other jobs to finish
  - You asked for something that doesn’t exist
    - More CPUs or nodes than exist
  - The job manager is optimistic!
    - If you ask for more resources than we have, the job manager will sometimes hold your job until we buy more hardware
Why won’t my jobs run?

- Use the `checkjob` command to see why your job won’t run

```
[26]s1::gpike> checkjob 319285
  job 319285

  Name: testjob
  State: Idle
  Creds: user:gpike group:users class:batch qos:od
  WallTime: 00:00:00 of 4:00:00
  SubmitTime: Wed Dec 1 20:01:42
  (Time Queued Total: 00:03:47 Eligible: 00:03:26)

  Total Requested Tasks: 320

  Req[0] TaskCount: 320 Partition: ALL

  Partition List: ALL,s82,SHARED,msm
  Flags: RESTARTABLE
  Attr: checkpoint
  StartPriority: 3
  NOTE: job cannot run (insufficient available procs: 312 available)
```

```
[27]s1::gpike>
```
Why won’t my jobs run?

- If you submitted a job that can’t run, use qdel to delete the job, fix your script, and resubmit the job
  - `qdel 319285`
- If you think your job should run, leave it in the queue and send email
- It’s also possible that maintenance is coming up soon
Making your job run sooner

- In general, specify the minimal set of resources you need
  - Use minimum number of nodes
  - Use the job queue with the shortest max walltime
    - `qstat -Q -f`
  - Specify the minimum amount of time you need for the job
    - `qsub -l walltime=hh:mm:ss`
Eucalyptus on FutureGrid

Archit Kulshrestha ~30 min
architk@gmail.com
Eucalyptus is an open-source software platform that implements IaaS-style cloud computing using the existing Linux-based infrastructure. IaaS Cloud Services providing atomic allocation for:

- Set of VMs
- Set of Storage resources
- Networking
Open Source Eucalyptus

- **Eucalyptus Features**
  - Amazon AWS Interface Compatibility
  - Web-based interface for cloud configuration and credential management.
  - Flexible Clustering and Availability Zones.
  - Network Management, Security Groups, Traffic Isolation
    - Elastic IPs, Group based firewalls etc.
  - Cloud Semantics and Self-Service Capability
    - Image registration and image attribute manipulation
  - Bucket-Based Storage Abstraction (S3-Compatible)
  - Block-Based Storage Abstraction (EBS-Compatible)
  - Xen and KVM Hypervisor Support

Source: http://www.eucalyptus.com
Eucalyptus Testbed

- Eucalyptus is available to FutureGrid Users on the India and Sierra clusters.

- Users can make use of a maximum of 50 nodes on India and 21 on Sierra. Each node supports up to 8 small VMs. Different Availability zones provide VMs with different compute and memory capacities.

<table>
<thead>
<tr>
<th>AVAILABILITYZONE</th>
<th>india 149.165.146.135</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABILITYZONE</td>
<td></td>
</tr>
<tr>
<td>vm types</td>
<td>free / max  cpu  ram  disk</td>
</tr>
<tr>
<td>m1.small</td>
<td>0400 / 0400  1  512  5</td>
</tr>
<tr>
<td>c1.medium</td>
<td>0400 / 0400  1  1024  7</td>
</tr>
<tr>
<td>m1.large</td>
<td>0200 / 0200  2  6000  10</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>0100 / 0100  2  12000 10</td>
</tr>
<tr>
<td>c1.xlarge</td>
<td>0050 / 0050  8  20000 10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>AVAILABILITYZONE</th>
<th>sierra 198.202.120.90</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVAILABILITYZONE</td>
<td></td>
</tr>
<tr>
<td>vm types</td>
<td>free / max  cpu  ram  disk</td>
</tr>
<tr>
<td>m1.small</td>
<td>0160 / 0160  1  512  5</td>
</tr>
<tr>
<td>c1.medium</td>
<td>0160 / 0160  1  1024  7</td>
</tr>
<tr>
<td>m1.large</td>
<td>0080 / 0080  2  6000  10</td>
</tr>
<tr>
<td>m1.xlarge</td>
<td>0040 / 0040  2  12000 10</td>
</tr>
<tr>
<td>c1.xlarge</td>
<td>0020 / 0020  8  30000 10l</td>
</tr>
</tbody>
</table>
Account Creation

- In order to be able to use Eucalyptus and obtain keys, users will need to request accounts at the Eucalyptus Web Interfaces at https://eucalyptus.india.futuregrid.org:8443/ and https://eucalyptus.sierra.futuregrid.org:8443/
  - In future there will be only one link
- On the Login page click on Apply for account
- On the next page that pops up fill out the Mandatory and optional sections of the form.
- Once complete click on signup and the Eucalyptus administrator will be notified of the account request.
- You will get an email once the account has been approved.
- Click on the link provided in the email to confirm and complete the account creation process.
Obtaining Credentials

- Download your credentials as a zip file from the web interface for use with euca2ools.
- Save this file and extract it for local use or copy it to India/Sierra.
- On the command prompt change to the euca2-{username}-x509 folder which was just created.
  - `cd euca2-{username}-x509`
- Source the eucarc file using the command `source ./eucarc`
Install/Load Euca2ools

- Euca2ools are the command line clients used to interact with Eucalyptus.
- If using your own platform Install euca2ools bundle from http://open.eucalyptus.com/downloads
  - Instructions for various Linux platforms are available on the download page.
- On FutureGrid log on to India/Sierra and load the Euca2ools module.

$ module add euca2ools
  euca2ools version 1.2 loaded
Euca2ools

- **Testing your setup**
  - Use `euca-describe-availability-zones` to test the setup.

- **List the existing images using `euca-describe-images`**

```
euca-describe-availability-zones
AVAILABILITYZONE india
  149.165.146.135

$ euca-describe-images
IMAGE emi-0B951139 centos53/centos.5-3.x86-64.img.manifest.xml admin available public x86_64 machine
IMAGE emi-409D0D73 rhel55/rhel55.img.manifest.xml admin available public x86_64 machine
...```
Key management

- Create a keypair and add the public key to eucalyptus.
  
  `euca-add-keypair userkey > userkey.pem`

- Fix the permissions on the generated private key.

  `chmod 0600 userkey.pem`

  `$ euca-describe-keypairs
Now we are ready to start a VM using one of the pre-existing images. We need the emi-id of the image that we wish to start. This was listed in the output of euca-describe-images command that we saw earlier.

- We use the euca-run-instances command to start the VM.

```
euca-run-instances -k userkey -n 1 emi-0B951139 -t c1.medium
RESERVATION r-4E730969 archit archit-default
INSTANCE i-4FC40839 emi-0B951139 0.0.0.0 0.0.0.0 pending userkey
2010-07-20T20:35:47.015Z eki-78EF12D2 eri-5BB61255
```
Monitoring

- `euca-describe-instances` shows the status of the VMs.

```
$ euca-describe-instances
RESERVATION r-4E730969 archit default
INSTANCE i-4FC40839 emi-0B951139 149.165.146.153 10.0.2.194
pending userkey 0 m1.small 2010-07-20T20:35:47.015Z india eki-78EF12D2 eri-5BB61255
```

- Shortly after...

```
$ euca-describe-instances
RESERVATION r-4E730969 archit default
INSTANCE i-4FC40839 emi-0B951139 149.165.146.153 10.0.2.194
running userkey 0 m1.small 2010-07-20T20:35:47.015Z india eki-78EF12D2 eri-5BB61255
```
First we must create rules to allow access to the VM over ssh.

```
euca-authorize -P tcp -p 22 -s 0.0.0.0/0 default
```

The ssh private key that was generated earlier can now be used to login to the VM.

```
ssh -i userkey.pem root@149.165.146.153
```
Image Deployment (1/3)

- We will use the example Fedora 10 image to test uploading images.
  - Download the gzipped tar ball

```
wget http://open.eucalyptus.com/sites/all/modules/pubdlcnt/pubdlcnt.php?
file=http://www.eucalyptussoftware.com/downloads/eucalyptus-
images/euca-fedora-10-x86_64.tar.gz&amp;nid=1210
```

- Uncompress and Untar the archive

```
tar zxf euca-fedora-10-x86_64.tar.gz
```
Next we bundle the image with a kernel and a ramdisk using the euca-bundle-image command.

- We will use the xen kernel already registered.
  - euca-describe-images returns the kernel and ramdisk IDs that we need.

```bash
$ euca-bundle-image -i euca-fedora-10-x86_64/fedora.10.x86-64.img --kernel eki-78EF12D2 --ramdisk eri-5BB61255
```

- Use the generated manifest file to upload the image to Walrus

```bash
$ euca-upload-bundle -b fedora-image-bucket -m /tmp/fedora.10.x86-64.img.manifest.xml
```
Register the image with Eucalyptus

euca-register fedora-image-bucket/fedora.10.x86-64.img.manifest.xml

This returns the image ID which can also be seen using euca-describe-images

$ euca-describe-images
IMAGE emi-FFC3154F fedora-image-bucket/fedora.10.x86-64.img.manifest.xml archit available public x86_64 machine eri-5BB61255 eki-78EF12D2
IMAGE emi-0B951139 centos53/centos.5-3.x86-64.img.manifest.xml admin available public x86_64 machine ...
Nimbus on FutureGrid
Nimbus

- **Hotel** (University of Chicago) 41 nodes, 328 cores
- **Foxtrot** (University of Florida) 26 nodes, 208 cores
- **Sierra** (San Diego Supercomputer Center) 18 nodes, 144 cores
- Online Tutorial: [http://www.futuregrid.org/tutorials/nm1](http://www.futuregrid.org/tutorials/nm1)
- FutureGrid users are automatically provided Nimbus credentials.
  - Login to Hotel to find the zip file with your nimbus credentials.
  - If missing write to help@futuregrid.org
- Go to the Nimbus tutorial tomorrow.... Room 216, 11:00AM
FutureGrid Phase II Services

Image Management
Dynamic Provisioning
Image Generation and Management on FutureGrid
Motivation

- The goal is to create and maintain platforms in custom FG VMs that can be retrieved, deployed, and provisioned on demand.
- Imagine the following scenario for FutureGrid:
  - `fg-image-generate -o ubuntu -v lucid -s openmpi-bin,openmpi-dev,gcc,fftw2,emacs -n ubuntu-mpi-dev`
  - `fg-image-store -i ajyounge-338373292.manifest.xml -n ubuntu-mpi-dev`
  - `fg-image-deploy -e india.futuregrid.org -i /tmp/ajyounge-338373292.manifest.xml`
  - `fg-rain -provision -n 32 ubuntu-mpi-dev`
Image Management

- A unified Image Management system to create and maintain VM and bare-metal images.
- Integrate images through a repository to instantiate services on demand with RAIN.
- Essentially enables the rapid development and deployment of Platform services on FutureGrid infrastructure.

http://futuregrid.org
Users who want to create a new FG image specify the following:
- OS type
- OS version
- Architecture
- Kernel
- Software Packages

Image is generated, then deployed to specified target.

Deployed image gets continuously scanned, verified, and updated.

Images are now available for use on the target deployed system.
Deployment View

[Diagram showing a network flow from users to web portal, image management, base OS, and cloud provider]
Implementation

● Image Generator
  o Still in development, but alpha available now.
  o Built in Python.
  o Debootstrap for debian & ubuntu, YUM for RHEL5, CentOS, & Fedora.
  o Simple CLI now, but later incorporate a web service to support the FG Portal.
  o Deployment to Eucalyptus & Bare

● Image Management
  o Currently operating an experimental BCFG2 server.
  o Image Generator auto-creates new user groups for software stacks.
  o Supporting RedHat and Ubuntu repo mirrors.
  o Scalability experiments of BCFG2 to be tested, but previous work shows scalability to thousands of VMs without problems.

http://futuregrid.org
Image Repository on FutureGrid

Gregor
Image Repository

Image Repository Client
- Portal
- CLI

Provisioning Subsystem
RAIN
Performance Subsystem

Image Repository Service Interface

FG Security

Image Repository Functionality
- Image usage information and user activity log
- Image attribute configuration
- Image management (upload, register, retrieve, remove, ...)
- Image metadata management (get, update, ...)
- User quota configuration

Metadata Store
Image Repository Storage and Access
Image Store

3rd Party Image Store
Dynamic Provisioning & RAIN on FutureGrid

Gregor (4 slides)

Include slides or link to slides here.
- dynamically partition a set of resources
- dynamically allocate the resources to users
- dynamically define the environment that the resource use
- dynamically assign them based on user request
- deallocate the resources so they can be dynamically allocated again
Use cases of dynamic provisioning

- **Static provisioning:**
  - Resources in a cluster may be statically reassigned based on the anticipated user requirements, part of an HPC or cloud service. It is still dynamic, but control is with the administrator. (Note some call this also dynamic provisioning.)

- **Automatic Dynamic provisioning:**
  - Replace the administrator with intelligent scheduler.

- **Queue-based dynamic provisioning:**
  - Provisioning of images is time consuming, group jobs using a similar environment and reuse the image. User just sees queue.

- **Deployment:**
  - Dynamic provisioning features are provided by a
Give me a virtual cluster with 30 nodes based on Xen
Give me 15 KVM nodes each in Chicago and Texas linked to Azure and Grid5000
Give me a Eucalyptus environment with 10 nodes
Give 32 MPI nodes running on first Linux and then Windows
Give me a Hadoop environment with 160 nodes
Give me a 1000 BLAST instances linked to Grid5000

Run my application on Hadoop, Dryad, Amazon and Azure … and compare the performance
● In FG dynamic provisioning goes beyond the services offered by common scheduling tools that provide such features.
  ○ Dynamic provisioning in FutureGrid means more than just providing an image
  ○ adapts the image at runtime and provides besides IaaS, PaaS, also SaaS
  ○ We call this “raining” an environment

● Rain = Runtime Adaptable INsertion Configurator
  ○ Users want to "rain" an HPC, a Cloud environment, or a virtual network onto our resources with little effort.
  ○ Command line tools supporting this task.
  ○ Integrated into Portal

● Example "rain" a Hadoop environment defined by an user on a cluster.
  ○ fq-hadoop -n 8 -app myHadoopApp.jar ...
**FG RAIN Command**

- fg-rain –h hostfile –iaas nimbus –image img
- fg-rain –h hostfile –paas hadoop …
- fg-rain –h hostfile –paas dryad …
- fg-rain –h hostfile –gaas gLite …

- fg-rain –h hostfile –image img

**Additional Authorization is required to use fg-rain without virtualization.**
Rain in FutureGrid

Dynamic Prov.

Cloud Frameworks
- Nimbus
- Eucalyptus

Map/Reduce Frameworks
- Hadoop
- Dryad

Parallel Programming Frameworks
- MPI
- OpenMP

Grids
- Globus
- EGEE

many many more

FG Perf. Monitor

Moab

XCAT

Future Grid
Portal

Gregor

Include slides or link to slides here.
What is happening on the system?
  o System administrator
  o User
  o Project Management & Funding agency

Remember FG is not just an HPC queue!
  o Which software is used?
  o Which images are used?
  o Which FG services are used (Nimbus, Eucalyptus, …?)
  o Is the performance we expect reached?
  o What happens on the network
### Machine Partition Information

<table>
<thead>
<tr>
<th>Resource</th>
<th>Eucalyptus</th>
<th>HPC</th>
<th>Mgmt</th>
<th>Misc</th>
<th>Nimbus</th>
</tr>
</thead>
<tbody>
<tr>
<td>iu-india</td>
<td>39.1%</td>
<td>55.5%</td>
<td>0.8%</td>
<td>4.7%</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>(400 cores)</td>
<td>(568 cores)</td>
<td>(8 cores)</td>
<td>(48 cores)</td>
<td></td>
</tr>
<tr>
<td>iu-xray</td>
<td>0</td>
<td>100%</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(672 cores)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>uc-hotel</td>
<td>0</td>
<td>50%</td>
<td>0</td>
<td>0</td>
<td>50%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(336 cores)</td>
<td></td>
<td></td>
<td>(336 cores)</td>
</tr>
<tr>
<td>ucsd-sierra</td>
<td>21.4%</td>
<td>46.4%</td>
<td>6%</td>
<td>2.4%</td>
<td>23.8%</td>
</tr>
<tr>
<td></td>
<td>(144 cores)</td>
<td>(312 cores)</td>
<td>(40 cores)</td>
<td>(16 cores)</td>
<td>(160 cores)</td>
</tr>
<tr>
<td>ufi-foxtrot</td>
<td>0</td>
<td>0</td>
<td>3.1%</td>
<td>0</td>
<td>96.9%</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(8 cores)</td>
<td></td>
<td>(248 cores)</td>
</tr>
</tbody>
</table>

### Number of cores allocated per type of partition

- **Eucalyptus**
- **HPC**
- **Mgmt**
- **Misc**
- **Nimbus**
Nimbus

This graph shows the number of currently running VMs within the Nimbus deployment on each machine.

Running VMs

x-axis = Timestamp, y-axis = Count

23Sep 24Sep 25Sep 26Sep 27Sep
Summary

- Introduced FG Resource overview
- Services for Phase I
  - HPC
  - Eucalyptus
  - Nimbus
- Outlook: Services for Phase II
  - Dynamic Provisioning
  - Image Management