ExoGENI/GIMI Tutorial
Part 1: ExoGENI

Ilia Baldine ibaldin@renci.org

Anirban Mandal, Yufeng Xin

Tutorial sections

- Configure environment
- ExoGENI and Orca Overview
- Creating slices with Omni and GENI AM API
- Flukes Overview
- Creating slices with Flukes
- Tutorial page:
  - http://groups.geni.net/geni/wiki/ORCAExoGENITutorial
  - Please open in your browser
  - Please open the presentation

Configuring Environment

- All user properties are under $HOME/.flukes.properties – it is a text file
  - Edit $HOME/.flukes.properties
    - Open in an editor and replace *EVERY* occurrence of XX with your index (including leading zero)
    - Update the orca.xmlrpc.url property
  - Inspecting keystore file (make note of key alias (‘tutorialXX’)
    - $ cd Tutorials/GIMI/gimiXX
    - $ keytool –list –keystore ssh/gimi01.jks
    - NOTE: your key name is gimiXX, and your key and keystore password is g5C7r#XX

Rack binding

- Today’s tutorial will use 3 separate AMs

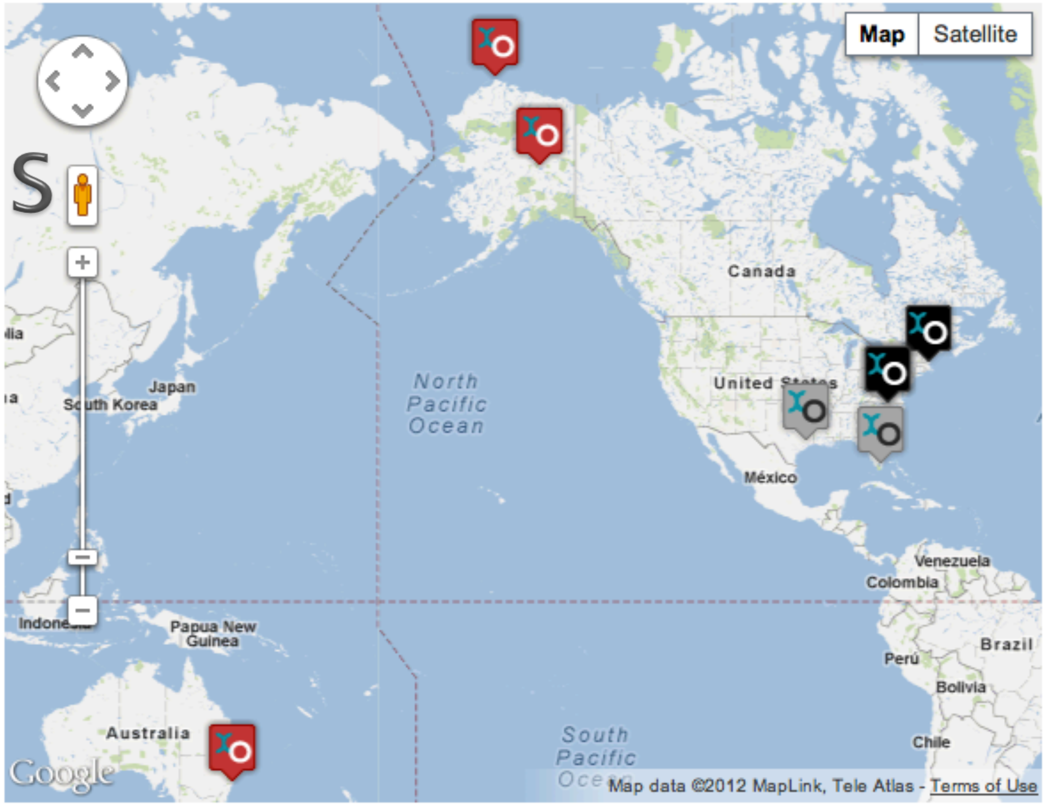
- Users 01-05 -> RCI
  - https://rci-hn.exogeni.net:11443/orca/xmlrpc
- Users 06-10 -> BBN
  - https://bbn-hn.exogeni.net:11443/orca/xmlrpc
- Users 11-20 -> NICTA
  - https://nicta-hn.exogeni.net:11443/orca/xmlrpc

Section: ExoGENI and ORCA Overview

ExoGENI Testbed

- 14 GPO-funded racks
  - Partnership between RENCI, Duke and IBM
  - IBM x3650 M3/M4 servers
    - 48G RAM
    - Dual-socket 8-core Intel X5650 2.66Ghz CPU
    - 10G dual-port Chelseo adapter
  - BNT 8264 10G/40G OpenFlow switch
  - DS3512 6TB sliverable storage
- Each rack is a small networked cloud
  - OpenStack- and xCAT based
  - EC2 nomenclature for VM node sizes (m1.small, m1.large etc)
  - Baremetal node provisioning
  - Interconnected by combination of dynamic and static L2 circuits through regionals and national backbones
- http://wiki.exogeni.net

ExoGENI Status

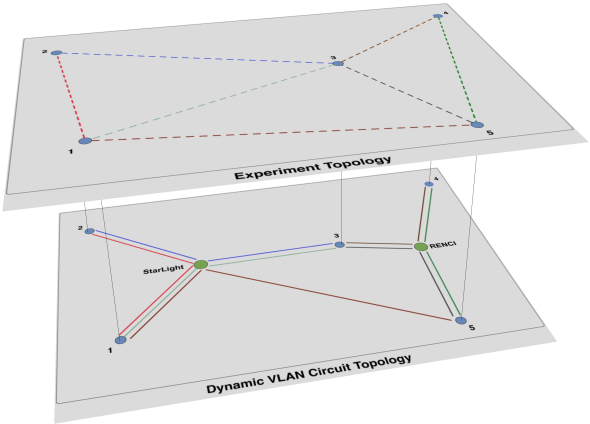


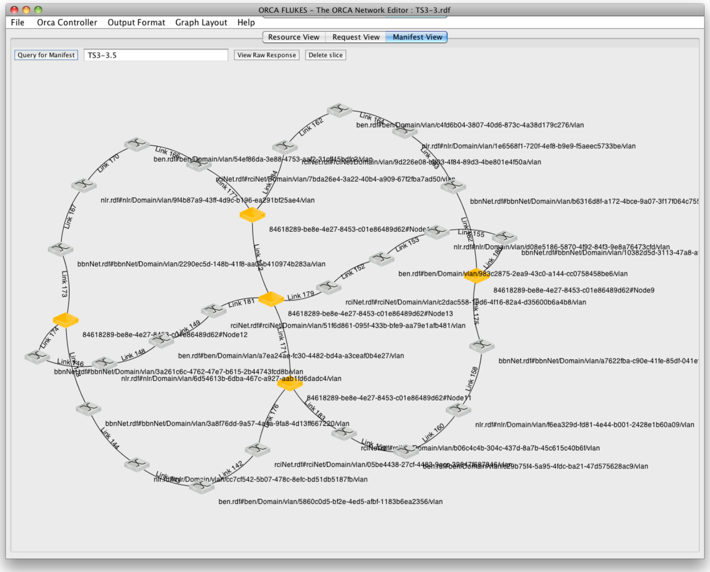
- 2 new racks deployed
  - RENCI and GPO
- 2 older existing racks
  - Duke and UNC
- 2 more racks coming
  - FIU and UH
- Partner racks
  - NICTA
  - U of Alaska Fairbanks

- Connected via BEN (http://ben.renci.org), LEARN and NLR FrameNet, (eventually I2)

ExoGENI slice isolation

- Strong isolation is the goal
- Compute instances are KVM based and get a dedicated number of cores (ExoGENI does not over-provision cores)
  - Caveat: currently all instances get 1 core (different RAM and disk).
- VLANs are the basis of connectivity
  - VLANs can be best effort or bandwidth-provisioned (within and between racks)
  - Caveat: current hardware in the racks allows best-effort VLANs only – will be remedied by Fall 2012 with support from the vendor




ORCA FLUKES – The ORCA Network Editor : TS3-3.rdf
File
Orca Controller
Output Format
Graph Layout
Help
Resource View
Request View
Manifest View
Query for Manifest
TS3-3.5
View Raw Response
Delete slice
Link 162
Link 164
ben.rdf#ben/Domain/vlan/c4fd6b04-3807-40d6-873c-4a38d179c276/vlan
nlr.rdf#nlr/Domain/vlan/1e6568f1-720f-4ef8-b9e9-f5aeec5733be/vlan
Link 170
Link 166
ben.rdf#ben/Domain/vlan/54ef86da-3e88-4753-aaf2-31cff45bdfc2/vlan
rciNet.rdf#rciNet/Domain/vlan/9d226e08-b803-4f84-89d3-4be801e4f50a/vlan
rciNet.rdf#rciNet/Domain/vlan/7bda26e4-3a22-40b4-a909-67f2fba7ad50/vlan
Link 177
Link 184
nlr.rdf#nlr/Domain/vlan/9f4b87a9-43ff-4d9c-b196-ea291bf25ae4/vlan
bbnNet.rdf#bbnNet/Domain/vlan/b6316d8f-a172-4bce-9a07-3f17f064c75
Link 167
84618289-be8e-4e27-8453-c01e86489d62#Node
Link 155
Link 153
nlr.rdf#nlr/Domain/vlan/d08e5186-5870-4f92-84f3-9e8a76473cfd/vlan
bbnNet.rdf#bbnNet/Domain/vlan/10382d5d-3113-47a8-a
bbnNet.rdf#bbnNet/Domain/vlan/2290ec5d-148b-41f8-aa04b410974b283a/vlan
Link 152
ben.rdf#ben/Domain/vlan/983c2875-2ea9-43c0-a144-cc0758458be6/vlan
Link 173
84618289-be8e-4e27-8453-c01e86489d62#Node9
Link 179
Link 181
rciNet.rdf#rciNet/Domain/vlan/c2dac558-19d6-4f16-82a4-d35600b6a4b8/vlan
84618289-be8e-4e27-8453-c01e86489d62#Node13
Link 149
Link 175
rciNet.rdf#rciNet/Domain/vlan/51f6d861-095f-433b-bfe9-aa79e1afb481/vlan
Link 174
84618289-be8e-4e27-8453-c01e86489d62#Node12
Link 148
Link 171
ben.rdf#ben/Domain/vlan/a7ea24ae-fc30-4482-bd4a-a3ceaf0b4e27/vlan
Link 146
bbnNet.rdf#bbnNet/Domain/vlan/a7622fba-c90e-41fe-85df-041e
bbnNet.rdf#bbnNet/Domain/vlan/3a261c6c-4762-47e7-b615-2b44743fcd8b/vlan
nlr.rdf#nlr/Domain/vlan/6d54613b-6dba-467c-a927-aab1fd6dadc4/vlan
Link 158
84618289-be8e-4e27-8453-c01e86489d62#Node11
Link 176
Link 183
bbnNet.rdf#bbnNet/Domain/vlan/3a8f76dd-9a57-4a4a-9fa8-4d13ff667220/vlan
nlr.rdf#nlr/Domain/vlan/f6ea329d-fd81-4e44-b001-2428e1b60a09/vlan
Link 144
Link 160
rciNet.rdf#rciNet/Domain/vlan/b06c4c4b-304c-437d-8a7b-45c615c40b6f/vlan
rciNet.rdf#rciNet/Domain/vlan/05be4438-27cf-4483-9ecc-39647f687846/vlan
ben.rdf#ben/Domain/vlan/e29b75f4-5a95-4fdc-ba21-47d575628ac9/vlan
Link 142
nlr.rdf#nlr/Domain/vlan/cc7cf542-5b07-478c-8efc-bd51db5187fb/vlan
ben.rdf#ben/Domain/vlan/5860c0d5-bf2e-4ed5-afbf-1183b6ea2356/vlan

ORCA Overview

- Originally developed by Jeff Chase and his students at Duke
- Funded as Control Framework Candidate for GENI
  - Jointly developed by RENCI and Duke for GENI since 2008.
- A federation of networked clouds with a variety of interfaces
  - Native ORCA
  - GENI AM API
- Unique feature of ExoGENI: experimenter can
  - Operate on individual racks as <u>independent aggregates</u>
  - Operate on <u>entire testbed</u> and link racks together using ExoSM

ORCA deployment in ExoGENI

- Each rack runs its own Orca actor called the ‘SM’ that presents as GENI AM and exposes
  - ORCA native API
  - GENI AM API
- Rack-local SM can only create slices with **resources within that rack (virtual machines, baremetal nodes and vlans)**
- ‘ExoSM’ has global visibility
  - Has access to a fraction of resources **resources in all racks**
  - Has access to **network backbone resources for stitching topologies between racks**
- ExoSM
  - https://geni.renci.org:11443/orca/xmlrpc
- Rack SMs:
  - RENCI Rack: https://rci-hn.exogeni.net:11443/orca/xmlrpc
  - BBN Rack: https://bbn-hn.exogeni.net:11443/orca/xmlrpc
  - NICTA Rack: https://nicta-hn.exogeni.net:11443/orca/xmlrpc

How are resources split?

- Resources in each rack are split between rack SM and ExoSM.
- Currently the split is 50/50 for VMs and internal VLANs between rack SM and ExoSM
- Baremetal nodes are usable only by ExoSM
- Stitching links from regional and national providers are delegated to ExoSM only.
- If your experiment is rack-local, you can always choose a specific rack SM to run your experiment
- If your experiment involves resources from multiple racks you must provision via ExoSM

How do I use it?

- Request credentials through the GPO
- Build your own VM Image [optional]
  - https://geni-orca.renci.org/trac/wiki/neuca-images
- Define topology and submit request
  - Use either Omni or Flukes
- Maximum slice lifetime is 2 weeks

Section: Creating slices with Omni

Creating slices with Omni and GENI AM API

- Use the RSpec file linked to this tutorial webpage:
  - http://groups.geni.net/geni/wiki/ORCAExoGENITutorial

Sample simple RSpec

```
<?xml version="1.0" encoding="UTF-8"?>
<rspec type="request"
       xsi:schemaLocation="http://www.geni.net/resources/rspec/3
                           http://www.geni.net/resources/rspec/3/request.xsd
                          http://www.protogeni.net/resources/rspec/ext/shared-vlan/1
                          http://www.protogeni.net/resources/rspec/ext/shared-vlan/1/request.xsd"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:s="http://www.protogeni.net/resources/rspec/ext/shared-vlan/1"
       xmlns=http://www.geni.net/resources/rspec/3>
<node client_id="geni1" component_manager_id="urn:publicid:IDN+bbnvmsite+authority+cm">
<sliver_type name="m1.small">
<disk_image name="http://geni-images.renci.org/images/standard/debian/debian-squeeze-amd64-neuca-2g.zfilesystem.sparse.v0.2.xml"
       version="397c431cb9249e1f361484b08674bc3381455bb9" />
</sliver_type>
<interface client_id="geni1:if0">
<ip address="172.16.2.1" netmask="255.255.255.0" />
</interface>
</node>
<node client_id="geni2" component_manager_id="urn:publicid:IDN+bbnvmsite+authority+cm">
<sliver_type name="m1.small">
<disk_image name="http://geni-images.renci.org/images/standard/debian/debian-squeeze-amd64-neuca-2g.zfilesystem.sparse.v0.2.xml"
       version="397c431cb9249e1f361484b08674bc3381455bb9" />
</sliver_type>
<interface client_id="geni2:if0" >
  <ip address="172.16.2.2" netmask="255.255.255.0" />
 </interface>
</node>
<link client_id="local">
  <interface_ref client_id="geni1:if0" />
  <interface_ref client_id="geni2:if0" />
</link>
</rspec>
```

Issuing OMNI commands (1/2)

- Look at ‘getVersion()’ output
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc getversion
- Download request RSpec from tutorial webpage
- Create a slice with GPO SA
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc createslice orcav2–test3
- CreateSliver with given RSpec
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc –n createsliver orcav2–test3 two–node.rspec

Issuing OMNI commands (2/2)

- Query sliver status
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc sliverstatus orcav2–test3
- List resources within the slice (IP addresses)
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc listresources orcav2–test3
- Login to nodes
  - ssh –i ssh/gimiXX_key root@<ip address>
- Delete sliver
  - omni.py –c omni_config –a https://geni.renci.org:11443/orca/xmlrpc deletesliver orcav2–test3

RSpec conventions

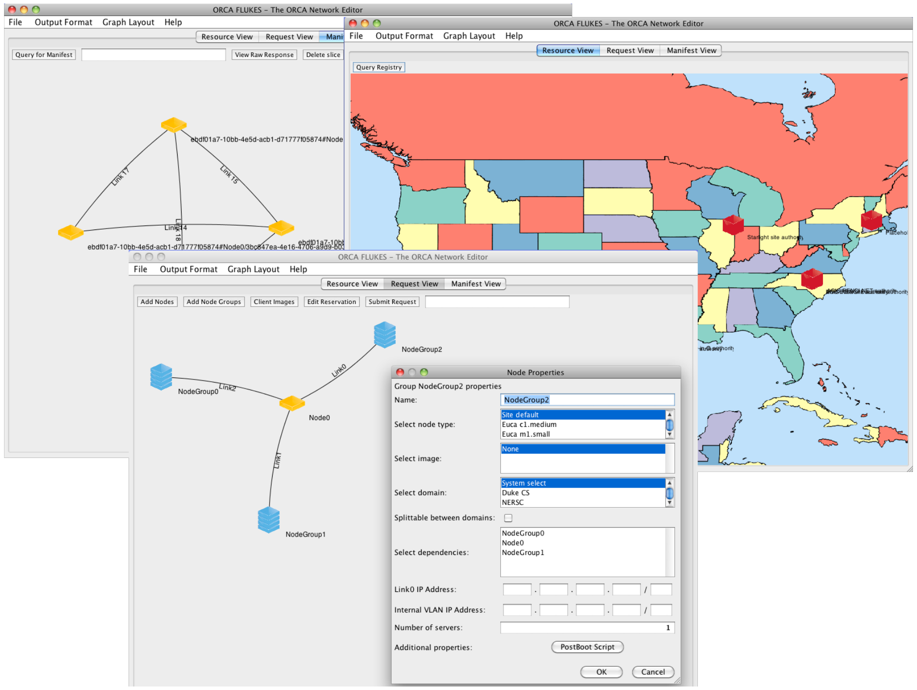
- *Install* and *execute* service tags are respected
  - tar.gz, tar.Z, tar.bz, deb, rpm and zip recognized
- Domain binding is possible by specifying component id or component manager id

```
<node component_id="urn:publicid:IDN+uncvmsite+node+vm"
      component_manager_id="urn:publicid:IDN+uncvmsite+authority+cm"
      client_id="pc175"
      exclusive="true">
  <sliver_type name="raw-pc" />
</node>
```

- See https://geni-orca.renci.org/trac/wiki/orca-and-rspec for updated information

Section: Flukes Overview

- Graphical tool for creating and managing slice topologies in ORCA
  - JAVA (JNLP)



Group NodeGroup2 properties

Name: NodeGroup2

Select node type: Site default, Euca c1.medium, Euca m1.small

Select image: None

Select domain: System select, Duke CS, NERSC

Splittable between domains: □

Select dependencies: NodeGroup0, Node0, NodeGroup1

Link0 IP Address: . . . /

Internal VLAN IP Address: . . . /

Number of servers: 1

Additional properties:

Section Overview

- Configuring Flukes prior to launch
- Launching Flukes
  - GUI Overview
  - Nodes, NodeGroups and Link parameters
  - Node-level vs. reservation level options
- Building slice request topologies
- Launching slice requests
- Inspecting slice manifests
- Logging into nodes in the slice
- NEuca-py tools

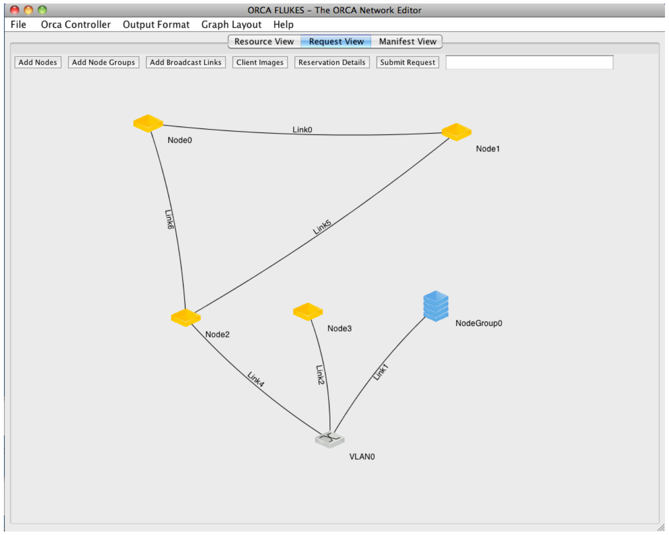
Launch Flukes!

- Double-click Flukes icon on your desktop
- Permanent stable version link
  - http://geni-images.renci.org/webstart/flukes.jnlp
- Can I use Flukes outside of ExoGENI?
  - No. Flukes uses semantic web mechanisms (RDF and OWL) to describe resources that is only compatible with ORCA and ExoGENI.

GUI Overview

- Tabs
  - Resources, Request, Manifest
- Menus
  - Current properties
  - Overwriting properties ($HOME/.flukes.properties)
- Mouse modes
- Buttons
- Adding nodes, nodegroups and links

Node and Link parameters



- Create a single node
- Right-click on the Node
  - Look at properties
  - Edit properties
    - Node type (size)
    - VM image
    - Domain (binding)
    - PostBoot script
- Create another node, link the two together
- Right-click and open properties again
  - Specify IP address on the link
  - Node functional dependencies
- Right-click on links
  - Inspect and edit link properties
  - Note only bandwidth is currently respected (and not everywhere due to hardware limitations)
- Broadcast links
- Specifying vlan tags
  - Only 'special' shared tags can be specified

NodeGroup parameters

- Create a single unattached node group
- Right-click to inspect and edit properties
  - Group sizes
  - Splittable groups

Nodes and NodeGroups

- A Node is an individual compute element
  - Typically a VM or a hardware node
  - IP address(es) on links, size, image, site binding, post boot script
  - Can I control management IP address assignment? NO!
- A NodeGroup is a group of identically configured nodes
  - A lot like a node except
    - PostBoot script is templated using Velocity template engine
      - https://geni-orca.renci.org/trac/wiki/flukes
    - IP address assignment is semi-automatic (starting with a user-specified address)
    - Node groups can be splittable between sites

What do I get when I …?

- Create a standalone node?
  - You get a single compute element at one of the sites with a single network interface to the management network through which you can SSH into the node
  - Management interface is always eth0
- Connect two nodes together?
  - You get two compute elements each with two network interfaces – one for management access and one for the link between two nodes.
  - User-controlled interfaces start with eth1
  - You can control IP address assignment on the interfaces linking the two nodes (suggested range: 172.16.0.0/16)

What do I get when I …?

- Create a standalone NodeGroup?
  - You get some number of nodes (specified in the group size) each with a single interface to the management network (eth0)
  - Nodes typically will be within the same rack
  - If node group is marked splittable nodes may be split across sites
- Connect a node group to a node or another node group?
  - All nodes within the group and the adjacent node (or all nodes in both groups) have interfaces on a common VLAN. They also have management interfaces (eth0)
  - IP address is specified similarly to private VLAN
  - Beware of address clashes! (i.e. here is a piece of rope, feel free to shoot yourself in the foot)

Can I tell which interface in the node will be eth1, eth2 etc?

- No, nor should you need to. Interfaces are identified by links they belong to.
- Note that different OSs name interfaces differently

Node-level vs. Reservation-level options

- Reservation-level options overwrite node-level options
  - VM image
  - Domain binding
- OpenFlow slice parameters can currently only be specified at reservation level

Domain binding

- Unbound requests are automatically bound to domains with available resources.
  - This depends on the visibility of the SM.
  - ExoSM can bind to any rack
  - Rack SM will always bind to its rack
- Bound requests are honored if resources are available
  - To create an inter-rack request, bind some of the nodes in it to one rack, and others to another
  - Can only be done via ExoSM!

Section: Creating slices with Flukes

Launching a slice

- Retrieve the RDF request file:
  - cd ~/Tutorials/GIMI/gimiXX
  - wget http://emmy9.casa.umass.edu/RDFs/gimiXX.rdf

- Click ‘File | Load Request’ to load the request
- Inspect the topology, node properties and post boot scripts

Launch a slice

- These will be unbound slices
  - We will let ORCA select sites with available resources
- Specify slice duration (click ‘Edit Reservation’)
- Fill in slice name (must be unique)
- Click ‘Submit Request’
  - Type in the alias of the key in the keystore (‘tutorialXX’)
  - Type in the password (‘tut0rialXX’)
  - ‘OK’
- Inspect the output window
  - Mainly a debugging tool. Will go away in the future.

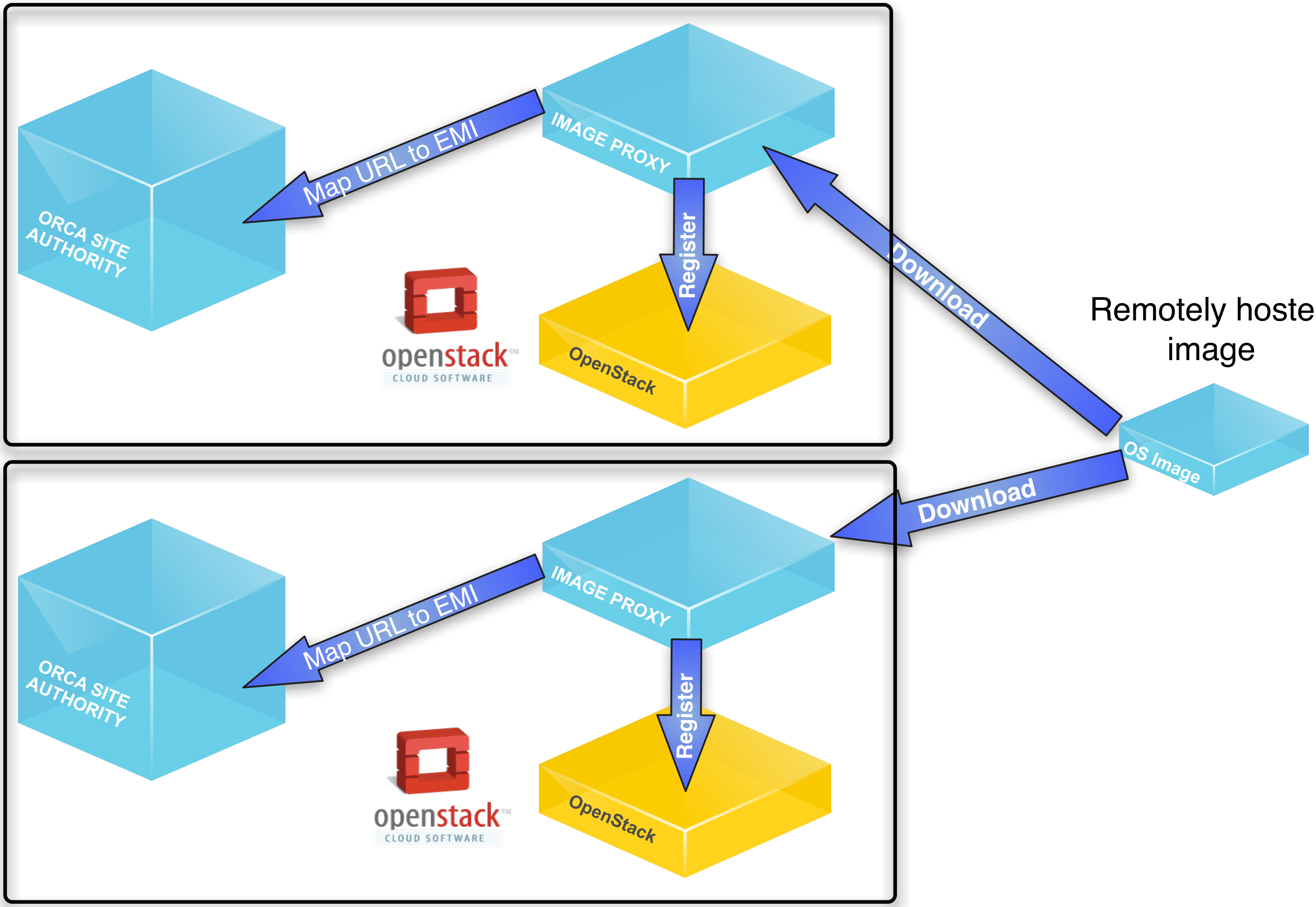
Inspect slice manifest

- Cut and paste slice name into the ‘Manifest View’ tab
- Click ‘Query for Manifest’
    - Inspect raw output if interested
    - Inspect the state of slice elements by right clicking on each element (usually ‘Ticketed’)
    - If you see ‘Failed’ you have a problem
- Poll by clicking ‘Query for Manifest’
    - Topology should materialize
    - All states should report ‘Active’
- Play around with layouts to get something pleasing

VM images

- Creating your own image
- Specifying your own image for ORCA
- Delays
  - Images are downloaded and registered with the site at the time of slice creation
    - If you repeatedly use the same image and the site already has it, this step is skipped
    - Images may be cached-out causing longer delays (to download and re-register)
- Are there examples of known good images?
  - Yes, visit https://geni-orca.renci.org/trac/wiki/neuca-images

VM Images



Remotely hosted image

Example image metafile

```
<images>
  <image>
     <type>ZFILESYSTEM</type>
     <signature>b54ed5a42cd99475c3d5d7c7a9839b69cf2076d5</signature>
     <url>http://geni-images.renci.org/images/workflows/pegasus/images/
 pegasus-4.0-v0.3.sparse.img.tgz</url>
  </image>
  <image>
     <type>KERNEL</type>
     <signature>f8a64d3bc429e8fb46c94ff3b11a932a27c142bc</signature>
     <url>http://geni-images.renci.org/images/workflows/debian-squeeze-
 kernel/vmlinuz-2.6.28-11-generic</url>
  </image>
  <image>
    <type>RAMDISK</type>
    <signature>6225968f43299aa40f6b1491360f3ce080bd16c4</signature>
    <url>http://geni-images.renci.org/images/workflows/debian-squeeze-kernel/
 initrd.img-2.6.28-11-generic</url>
  </image>
</images>
```

How does ORCA refer to an image

- URL of a metafile (can be same or different webserver as the image)
- SHA1 checksum of the metafile (to ensure it has not been modified)
- Workflow
  - Create filesystem, kernel ramdisk
  - Place on webserver
  - Take SHA1 signatures of each file
  - Generate metafile
  - Take SHA1 signature of metafile and its URL and add it to .flukes.properties or put it in Rspec
  - Try on a small slice (one node) to make sure it boots
- Can I put my image on your server?
  - Sorry, no.
- Will my image always remain cached at the racks?
  - No, depending on the use, your image may be cached out.

Doman binding

- Leaving domain as 'System Select' leaves ORCA to pick the domain that has available resources
- You can explicitly bind to specific domains
- If nodes or groups in slice request belong to different domains, appropriate inter-domain links will be provisioned on demand

Inspect slice manifest

- Cut and paste slice name into the ‘Manifest View’ tab
- Click ‘Query for Manifest’
  - Inspect raw output if interested
  - Inspect the state of slice elements by right clicking on each element (usually ‘Ticketed’)
  - If you see ‘Failed’ you have a problem
- Poll by clicking ‘Query for Manifest’
  - Topology should materialize
  - All states should report ‘Active’
- Play around with layouts to get something pleasing

Logging into nodes

- Right click on node
- Select ‘Login to node’
  - In terminal window type in SSH key password (‘gec13’)
- Inspect uptime

```
$ uptime
```

- Inspect the output of your boot script
- Inspect interfaces

```
$ ifconfig
```

- Try to ping node neighbors

NEuca-py tools

- NEuca tools are loaded in the image
  - They configure network interfaces at boot time
  - They execute the post boot script
  - An image with NEuca tools will do neither of those things
    - You can still configure interfaces manually
- Allow you to inspect the VM configuration
- Run ‘neuca’ to get the list of neuca tools

  $ neuca
- Run ‘neuca-user-data’
  - Note your boot script
- If you create your own VM image you are strongly encouraged to install NEuca tools on it
  - Visit https://geni-orca.renci.org/trac/wiki/NEuca-guest-configuration for instructions

Building your own image

- Build one from scratch using instructions from OpenStack or Eucalyptus
  - Virtio support is required
  - NEuca-py tools should be installed
- Build one by adding packages to one of the existing images
- Use post-boot scripts to install packages after the VM boots
  - E.g. wget a tar file or a deb or an RPM and install it
  - Beware some tools (e.g. apt-get) need a controlling TTY, which is not available when executed from post-boot script

Thank you for attending the tutorial!

- More Orca information
  - http://geni-orca.renci.org
- More ExoGENI information
  - http://wiki.exogeni.net