# FermiCloud Status Report
# Fall 2010

Keith Chadwick
Grid & Cloud Computing Department Head
Fermilab
chadwick@fnal.gov

Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

# People

Steve Timm (project leader)

Gabriele Garzoglio

Ted Hesselroth

Faarooq C. Lowe

Parag Mhashilkar

Neha Sharma

Doug Strain

Dan Yocum

# Types of Cloud - 1

## Infrastructure as a Service (IaaS):

– The delivery of computer infrastructure as a service.

FermiCloud

## Software as a Service (SaaS):

– The delivery of software "solution stack" on demand as a service.

## Platform as a Service (PaaS):

– The delivery of a computer infrastructure and solution stack as a service.

# Types of Cloud - 2

## Public Cloud:

- Amazon EC2.

## Private Cloud:

- FermiCloud.
- Magellan.

## Hybrid Cloud:

- Cloud composed of Private and Public resources.

## Cloud Bursting:

- Dynamically adding other public or private cloud resources to a cloud to add "instantaneous" capacity.

# Multiple Projects

## FermiGrid Services:

- Highly available statically provisioned virtualized services.
- SLF5+Xen.

## General Physics Compute Facility (GPCF):

- Mostly statically provisioned virtualized systems for scientific analysis.
- Oracle VM.

## FermiCloud:

- Scientific Infrastructure as a Service.
- SLF5+Xen, SLF5+KVM.

## Virtual Services Group:

- Virtualization of Fermilab business systems using VMware.
- Microsoft Windows.

# Need for FermiCloud

Large need for development, integration, and testing machines.

- Many only need to be active during testing cycles, not all the time.

Previous developer machines were old legacy hardware:

- Keeping the old legacy hardware alive and healthy required a significant amount of system administrator support.
- The old legacy systems were extremely power hungry.

There weren't enough of them, and they didn't have enough resources:

- RAM, disk, CPU, etc.

FermiCloud was already in planning when the two power incidents at Fermilab killed half of the developer machines and forced us to turn the other half off.

Not in scope of FermiCloud phase 1:

- Virtualizing worker nodes,
- Accepting Virtual Machines as grid jobs.
- These will addressed in later phases of the project.

# Stakeholders and Early Adopters

## Joint Dark Energy Mission (WFIRST):

- Distributed messaging system, testing fault tolerance, ideal application for cloud

## Grid Department Developers:

- Authentication/Authorization
- Storage evaluation
- Monitoring/MCAS
- GlideinWMS

## dCache Developers

## LQCD testbed

# Initial Project Plan

## Technology Evaluation:

- Hypervisors, both Open Source and Commercial.
- Cloud Control Software.
- Provisioning and contextualization.
- Scheduling.
- File Systems.

## Requirements Gathering:

- HW and SW technical.
- Security.
- Stakeholder needs.

## Customization and Deployment.

# Common Cloud Concepts

Overall User Interface for requesting a VM (cloud controller).

One or more Cluster Controllers that control a group of nodes.

A Node Controller on each node that can run virtual machines.

A repository of virtual image files.

Lack of useful documentation.

# FermiCloud Hardware

Individual System:

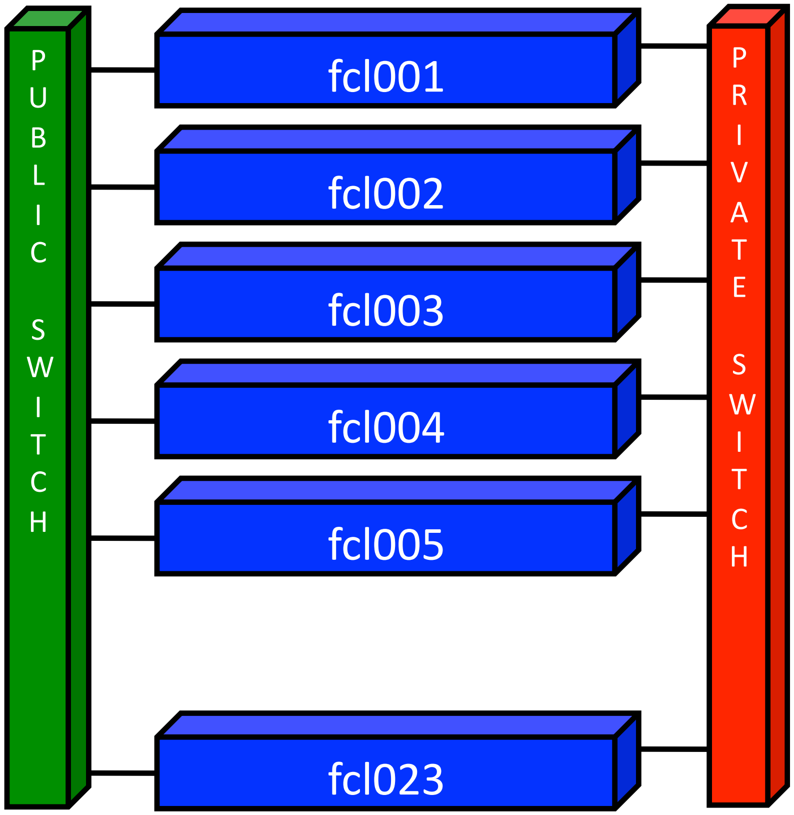
- 2 x Quad Core Intel Xeon E5640 CPU
- 24GB RAM
- Storage:
  - 2 x 300 GB SAS 15K rpm system disk.
  - 6 x 2TB SATA disk.
  - LSI 1078 RAID controller.
- Connect-X QDR Infiniband

Currently 23 systems total.

Will be expanding to 36 systems later this year.

# Network Topology

## Physical:



PUBLIC SWITCH

fcl001

fcl002

fcl003

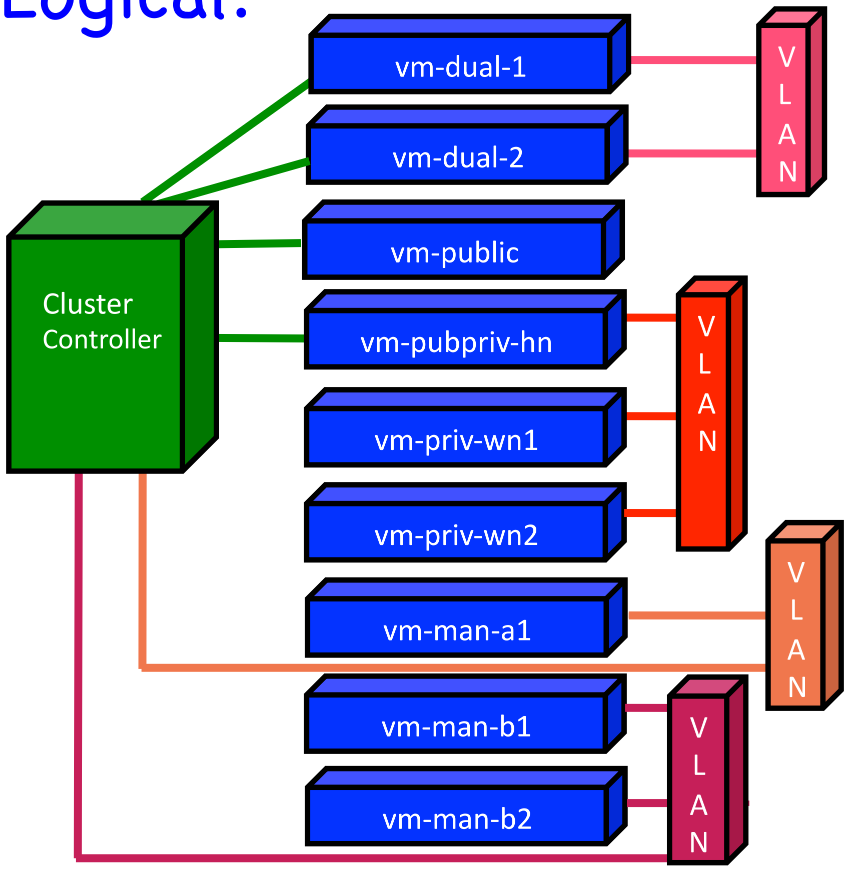
fcl004

fcl005

fcl023

PRIVATE SWITCH

## Logical:



Cluster Controller

vm-dual-1

vm-dual-2

vm-public

vm-pubpriv-hn

vm-priv-wn1

vm-priv-wn2

vm-man-a1

vm-man-b1

vm-man-b2

VLAN

VLAN

VLAN

VLAN

# Hypervisor Evaluation - 1

## Commercial Hypervisors:

- VMware cost-prohibitive for 50-processor cloud, although used on the business systems of FNAL.
- Oracle VM (commercialized Xen) used in some scientific applications, less costly. Many of its features gradually coming to open source, XCP, etc.

## Open Source Hypervisors:

- Xen and KVM.

## Measured disk throughput, intra-node network speed, inter-node network speed for KVM and Xen under SL5.4.

- Paravirtualized Xen shows near-bare-iron I/O disk rates, near wire speed for intra-node networking
- KVM shows about 50% disk speed of bare iron, but faster network throughput with virtualized IO drivers.
- Expect that the performance penalty will decrease with time.

# Hypervisor Evaluation - 2

KVM as shipped with SL<=5.5 has a few reliability issues.

RHEL6 doesn't have native Xen support, promises improved KVM. Not tested yet.

RHEL5 updates 4 and 5 (as recompiled by Scientific Linux) slowly breaking Xen kernel, introducing bugs.

- Random time skews for 32 bit VMs on 64 bit hypervisor.

Conclusion -> KVM is likely to be the bulk of VM's in FermiCloud but we will keep capacity to run Xen as well especially for DB and IO.

Having one OS image that can run on both — even better.

# Fermilab Requirements

Cloud machines, especially developer machines expect:

- Public addressable IP
- Static IP (for grid applications)
- Strong authentication – Kerberos login

Run all Fermi-baselined operating systems

- Windows, Scientific Linux [4 & 5], CERNVM.

Keep systems up to current patch levels.

Reasonable power draw and cost.

Big, fast, and cheap storage on each node for storage system testing.

# Stakeholder Requirements

Provision cluster of associated machines.

Scheduling and allocation features.

Harvest idle virtual machines and deploy worker node virtual machines overnight.

Capacity to pause and resume virtual machine and save state.

High bandwidth interconnect for MPI testing.

# Evaluated Cloud Frameworks

The FermiCloud project has evaluated the following three Cloud frameworks:

- Eucalyptus
- Nimbus
- OpenNebula

# Eucalyptus Evaluation

## Strengths:

- Good RPM packaging
- Support for Xen and KVM
- Includes Walrus (S3 emulation) to store virtual images and EBS emulation to store block devices
- Scalable architecture built on web services
- Uses 3rd-party add-ons for Amazon EC2, including HybridFox GUI
- Supports a number of different network topologies
- This was the first one that worked for us.
- Security groups allow for separate private nets for different users

## Weaknesses:

- No notion of scheduling (but could use Condor as front-end scheduler).
- Possible but not easy to save state of VM's
- Difficult how to figure out how to format VM's
- Multi-node coordinated cluster launch is difficult.
- Now a commercial company, most of goodies are, or will be, in enterprise version.

# Nimbus Evaluation

## Strengths:

- VM instantiation available via WSRF (Grid) interface and EC2.
- Multi-cluster launch is easy
- Can launch VM's via pilot job in PBS batch system
- Well-developed scheduling and allocation system
- Open-source system catering to science clouds and looking for extensions.

## Weaknesses:

- Image distribution means installing your own GridFTP server, not documented at all.
  - Nimbus 2.5 and greater replace this with S3 emulation
- Privilege separation model needs work particularly in libvirt communications.
- Dependence on SimpleCA certificate authority

# OpenNebula Evaluation

Strengths:

- Large developer and user base.
- Rich API.
- Good scheduling features.
- Least system administrator time required to install it.
- Very flexible capabilities to create many different kinds of virtual machines, different network topologies.
- Scalable ways to provision lots of virtual machines rapidly.

Weaknesses:

- Express documentation geared to a couple specific use cases, it was hard to generalize to get something working.
- Default security is wide open, pluggable mechanisms are available but takes time to make it work.
- EC2 API is limited, (no SOAP interface) doesn't work with Condor-G.

# Default Authentication/Authorization of Common Open Source Clouds

| CLOUD SYSTEM | Upload Image | Launch VM CLI | Launch VM API | Login |
|---|---|---|---|---|
| Eucalyptus | X509 | X509 | X509, EC2_ACCESS_KEY | ssh-keypair |
| Nimbus | X509 | X509 | X509, EC2_ACCESS_KEY | ssh-keypair |
| OpenNebula | user/pass | user/pass | user/pass, EC2_ACCESS_KEY | ssh-keypair |

# What we have now

Eucalyptus:

– 3 nodes Xen, 3 nodes KVM

OpenNebula:

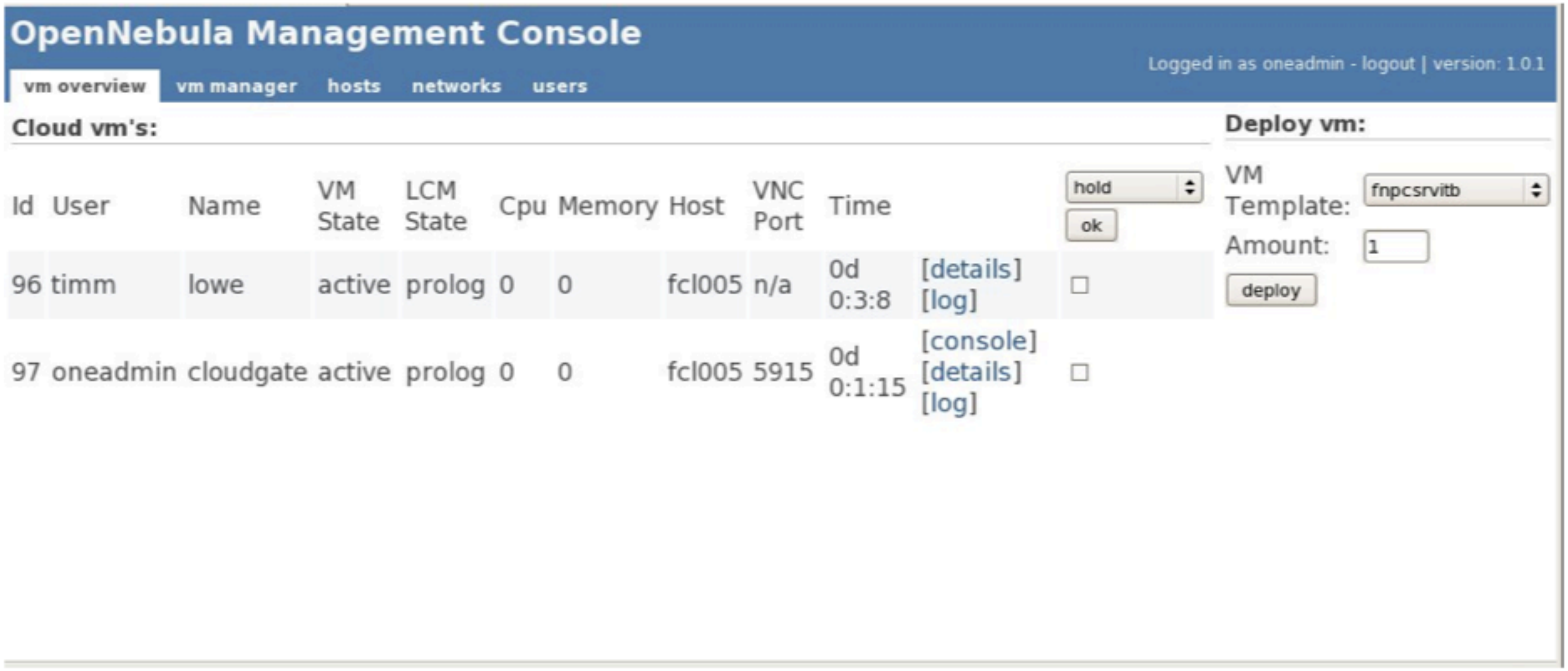
– 6 nodes

First stakeholder, JDEM, has begun work.

Will eventually deploy on 23 systems = 368 logical cores (with hyperthreading enabled).

– Expanding to 36 systems = 576 cores shortly.

# OpenNebula Management Console


OpenNebula Management Console
Logged in as oneadmin - logout | version: 1.0.1
vm overview
vm manager
hosts
networks
users

**Cloud vm's:**

| Id | User | Name | VM State | LCM State | Cpu | Memory | Host | VNC Port | Time | | hold<br>ok |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 96 | timm | lowe | active | prolog | 0 | 0 | fcl005 | n/a | 0d 0:3:8 | [details]<br>[log] | □ |
| 97 | oneadmin | cloudgate | active | prolog | 0 | 0 | fcl005 | 5915 | 0d 0:1:15 | [console]<br>[details]<br>[log] | □ |

**Deploy vm:**

VM Template: fnpcsrvitb

Amount: 1

deploy

# Contextualization

For all cloud software we have developed a script to fetch node-specific files (kerberos keytab, host x509 cert and keys) via ssl-encrypted wget.

OpenNebula allows to attach an ISO image of files when VM is launched.

Eucalyptus has Instance Metadata, node-specific information that can be fetched by wget once the node is up, including user-defined information.

We have modified OpenNebula's network script to start network earlier.

– In OpenNebula 2.0 they have done that too.

# Security Policy

Goal: Take a virtual machine and make it a node that is legal to run on the Fermilab public network.

Fermilab Enclaves:

- Open Science Enclave - Can use Grid credentials to launch a job, protected by strong controls. Worker node machines, grid integration machines, are here.
- General Computing Enclave - For regular login, can only use strong-auth credentials such as Kerberos for login. Most of cloud machines will be here.
- Network Jail - Untrusted place where unregistered laptops get sent when they first come on site. Can only access update sites.

All cloud provisioning techniques use ssh key-pair for initial contact by default.

- Our site policy does not allow this, we inject Kerberos credentials instead.

# Provisioning and Patching

We want a “kickstart-me-now” feature in which a user can request a PXE boot and a clean kickstart install of Scientific Linux with a user-specified KS file.

Leverage Fermilab's (network jail) scanning and node registration service and treat the new VM like an incoming hostile laptop, don't let it on the net until it is scanned, registered and patched.

Wake dormant (shelved) machines up from time to time and give them their patches.

- Use IPtables on a restricted hypervisor to restrict network access to only patch update sites.

Both Eucalyptus and Nimbus give only the system administrators the power to upload kernels and ramdisk.

- Gives control over which OS you will support and keeping the kernels updated.

# Storage Evaluation

Looking to identify new many-to-many storage technology.

- Currently use Bluearc NAS device.

Trying Lustre, Hadoop and other solutions on both "bare iron" and under KVM.

8-core machine probably unnecessary to serve 10TB of disk:

- Hope to allocate 2-4 cores to serve storage.
- Have the rest of the cores be allocated compute virtual machines.

Testing with actual neutrino experiment "root" application, benchmarking different solutions.

Lustre:

- Clients work at wire speed in Xen and KVM.
- Servers work under KVM but there is a performance hit:
  - The numbers from our Lustre evaluation w/ 3 OST and 1 MDT were:
    - Bare Metal: 350 MB/s read; 250 MB/s write
    - KVM: 350 MB/s read; 80 MB/s write
  - We are investigating what can be done to eliminate or at least minimize the factor of 3 overhead for write.
  - Changing memory and number of CPUs dedicated to the virtual server did not significantly change the bandwidth to storage.

Investigations of Hadoop and other solutions still to come.

# Infiniband and MPI

Lattice QCD cluster wants capacity to make mini-MPI cluster of 4 virtual nodes for users to test applications on.

For this we want actual Infiniband drivers and hardware present and visible in the VM's.

3rd generation "Infiniscale" cards can be passed through to 1 VM

4th generation "Connect-X" cards claim that they can be shared with several VM on same machine.

– Investigation continues...

Can use Infiniband for private IP network applications if faster private net is necessary

Infiniband can also be important for connection to storage and SAN.

# FermiCloud Phase 2

FermiCloud Phase 2 has already been approved!

Target is small low-cpu-load production servers:

- Grid gatekeepers, forwarding nodes, small databases, monitoring, etc.

Many other "small" server purchases will be directed to FermiCloud for resources:

- Contribute $$$ to FermiCloud
- Receive corresponding allocation of Virtual Machines and resources.

Live migration becomes important for this phase.

First of these services are up:

- mostly Integration Test Bed stuff right now.

# Ongoing Research

Amazon EC2 REST (Query) API is very useful but is it secure?

- We are currently investigating this area of work.

Goal is to require Kerberos or GSI authentication to launch a VM as well as log into it.

- It's claimed this can be done but we haven't made it work yet!

For GSI authentication on EC2 SOAP API, want to use all IGTF-trusted certificates.

- As delivered the default is SimpleCA, this does not meet the Fermilab computer policy requirements.
- We think it can be done but will take some work.

# Conclusions

The FermiCloud project has evaluated three cloud technologies thus far:

- Nimbus, Eucalyptus, and OpenNebula.

All do what they claim to do.

- We are now letting early adopters try them out and see what works.

We have currently deployed a mix of Eucalyptus and OpenNebula,

- Our plan is to use a weighted decision matrix do determine the best candidate.
- We haven't picked a final winner yet, final report is still in draft pending ongoing evaluation.

All three cloud technologies are still in early phases of development,

- All are evolving rapidly and are adding new features.
- We will try to stay as generic as possible so we don't have vendor lock-in.

# Fin

## Any Questions?