FermiCloud

K. Chadwick, T. Hesselroth, F. Lowe, S. Timm, D. R. Yocum
Grid And Cloud Computing Department
Fermilab
ISGC2011

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

Fermilab

Cloud Computing Introduction

- 3 basic types of Cloud Computing Services:
  - Infrastructure-as-a-service (Magellan, Amazon Web Services)
  - Platform-as-a-service (Windows Azure, Google App Engine)
  - Software-as-a-service (salesforce.com, Kronos)
- 4 types of Cloud:
  - Public cloud – Web API allows all authorized users to launch virtual machines remotely on your cloud. (Amazon)
  - Private cloud – Only users from your facility can use your cloud (FermiCloud)
  - Community cloud – Only users from your community can use your cloud (Magellan)
  - Hybrid cloud – Infrastructure built from mix of public and private.
- Object-oriented storage (Hadoop, etc.) closely linked to cloud paradigm.
- In the cloud paradigm, resources are provisioned "on demand" and decommissioned when the user no longer needs the resources.

Common Cloud Concepts

- Overall User Interface for requesting a VM (Cloud Controller + API),
- One or more Cloud Controllers which control a group of nodes,
- A Node Controller on each node which can activate virtual machines,
- A repository of virtual image files,
- "Ecosystem" – the group of developers and users who make 3rd-party tools for cloud computing,
- Hypervisor – the part of operating system which manages virtual machines.

Related Fermilab Enterprise Virtualization Projects

- FermiGrid Services:
  - Highly Available statically provisioned virtual services,
  - SLF5+Xen, SLF5+kvm.
- General Physics Compute Facility (GPCF):
  - Deployment of experiment-specific virtual machines for Intensity Frontier experiments,
  - Oracle VM (Commercialized Xen).
- Virtual Services Group:
  - Virtualization of Fermilab core computing/business systems using VMware,
  - Windows,
  - RHEL/SLF in future.

What is FermiCloud?

- Infrastructure-as-a-service facility:
  - Developers, integrators, and testers get access to virtual machines without system administrator intervention,
  - Virtual machines are created by users and destroyed by users when no longer needed. (Idle VM detection coming in phase 2),
  - Testbed to let us try out new storage applications for grid and cloud.
- A Private cloud – on-site access only for registered Fermilab users.
- A Project to evaluate the technology, make the requirements, and deploy the facility.
- Unique use case for cloud – on the public production network, integrated with rest of infrastructure.

Drivers for FermiCloud

- Previous developer machines in the FAPL/Gridworks cluster were 8+ years old with limited memory and CPU, and were slowly dying, then two unplanned power outages in February 2010 essentially killed off the remainder.
- Developers/integrators need for machines delivered on fast turnarounds for short periods of time.
- Improved utilization of power, cooling, and employee time for managing small servers and integration machines.
- CERN IT + HEPiX Virtualisation Taskforce program to have uniformly-deployable virtual machines.
- Virtualization under extensive use by SNS, FEF, FGS, and CMS T1.
- 16+core systems lend themselves to hosting multiple logical servers on the same physical hardware.

FermiCloud Project Staff

- Steve Timm - project lead,
- Dan Yocum, Faarooq Lowe - hypervisor and cloud control software installation and evaluation, early user support,
- Keith Chadwick - management and security policy,
- Gabriele Garzoglio, Doug Strain - storage evaluation
- Ted Hesselroth - authentication and authorization development,
- Many other Grid dept. staff and stakeholders who come regularly to meetings and tried early versions of cloud.

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/test-stands,
  - Monitoring/MCAS (Metrics Correlation and Analysis Service),
  - GlideinWMS.
- Fermilab Scientist Survey,
- Extenci project (wide area Lustre),
- REX department - production GlideinWMS forwarding node server.

FermiCloud Project Phase 1

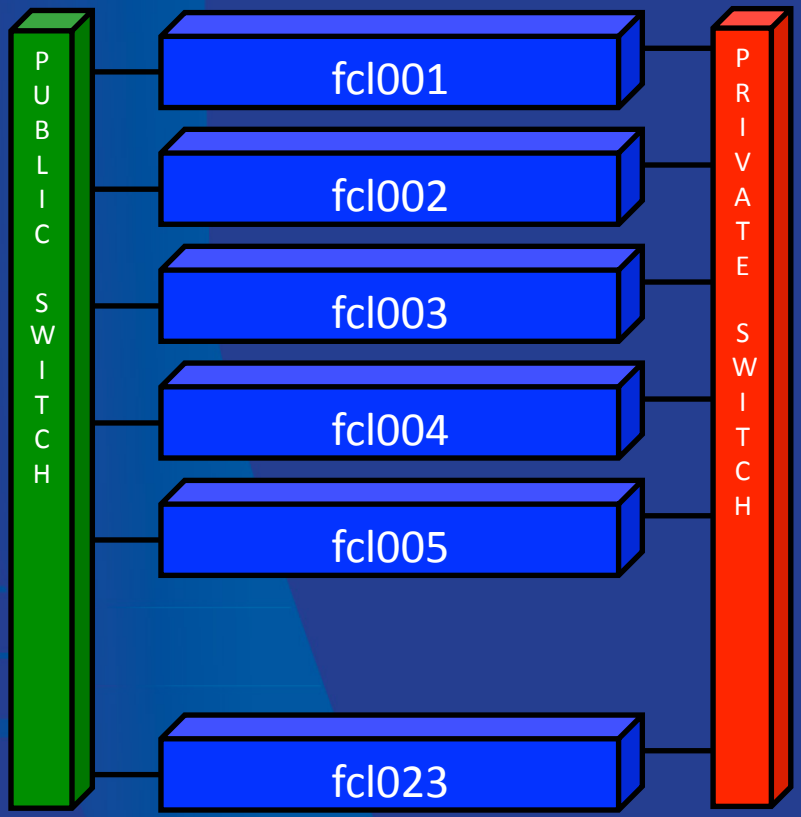
- Acquisition of FermiCloud hardware (done),
- Development of requirements based on stakeholder inputs (done),
- Review of how well open source cloud computing frameworks (Eucalyptus, OpenNebula, Nimbus) match the requirements (done),
- Storage evaluation (in process, see G. Garzoglio talk this conference):
  - Lustre, Hadoop, BlueArc, OrangeFS.
- Being used by Grid, CET, and REX developers and integrators.

FermiCloud Hardware

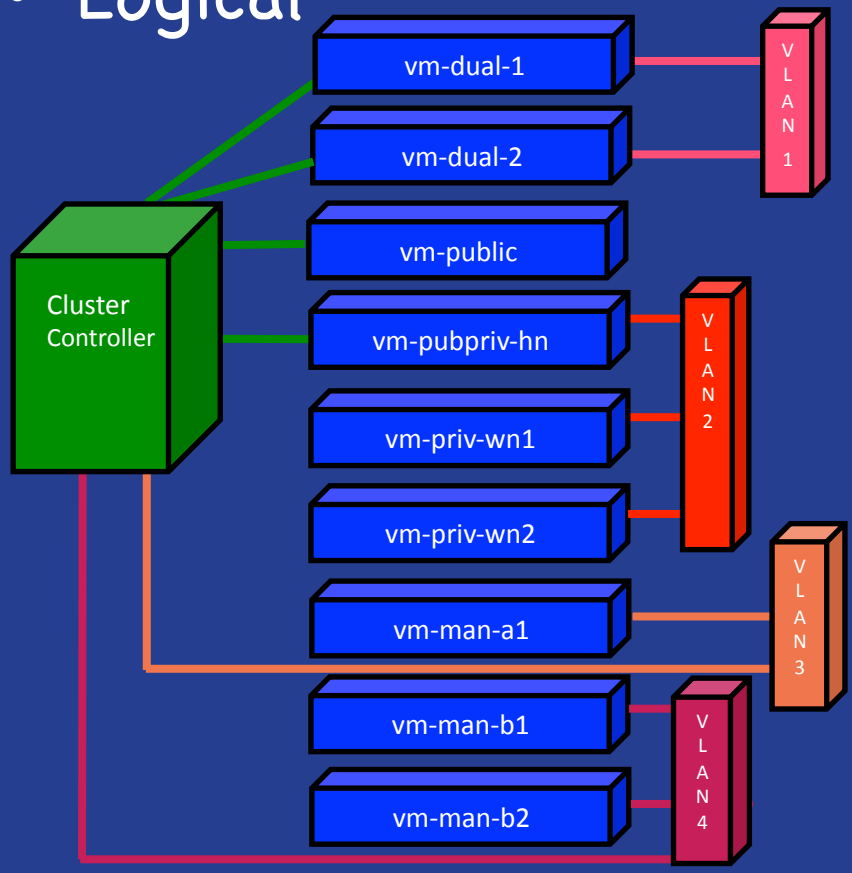
- 2x Quad Core Intel Xeon E5640 CPU
- 2 SAS 15K RPM system disk 300GB
- 6x 2TB SATA disk
- LSI 1078 RAID controller
- Infiniband card
- 24GB RAM
- 23 machines total
- Arrived June 2010
- +25TB BlueArc NAS disk

FermiCloud Network Topology

- Physical



- Logical



Requirements

- OS: SLF, Fedora, RHEL, Windows,
- Hypervisor: Xen, KVM,
- Flexible machines: Multiple networks, Infiniband, multiple disks,
- Provisioning: Clusters of VM's, leverage cfengine and puppet, add secrets such as krb5.keytab at launch time,
- Object store: No machine dependent secrets stored on machine image,
- Compatible with coming WLCG/Hepix standards for VM exchange and endorsing,
- Interoperability: EC2 SOAP and ReST, Condor-G, CERNVM, CVMFS, cloudburst from FermiCloud to EC2 or DOE Magellan,
- Functionality: Pause and save virtual machines, live migration, stable running, reboot without loss of VM's,
- Network topology: See previous network topology slide,
- Accounting: Who is using VM's, how much CPU, memory, disk in each VM.

Requirements - Security

- New VM's subjected to network vulnerability and virus scan before allowed on Fermi Network, leverage laptop network jail if possible.
- VM's must use standard site-wide patching mechanisms.
- Periodically wake up dormant virtual machines to be sure they get their patches.
- Must have either Kerberos or x509 credential to launch a virtual machine and to log into it once it's launched.
- Cloud daemons must communicate via secure protocols.
- If x509 used, must be possible to replace SimpleCA with certificates issued by IGTF accredited CA.

Hypervisor Evaluation - Xen

- At Fermilab since 2004,
- Consists of hypervisor, paravirtualized kernel, user tools,
- Supports both paravirtualization and full hardware virtualization,
- Open Source (Citrix/EMC distribute commercial version also available),
- FermiGrid uses paravirtualized Xen almost exclusively
  - On all production grid gatekeepers, auth servers, batch system masters, and databases,
- Part of Scientific Linux since SL 5.2,
- Red Hat drops support for Xen hypervisor in RHEL6 but RHEL6 can still be a Xen guest,
- If necessary, we could get Xen hypervisor rpm's from xen.org as we did before,
- Some time instability seen in 32-bit guest OS from SLF5.4+,
- Paravirtualized performance very good, almost indistinguishable from bare metal.

Hypervisor Evaluation - kvm

- Bought a few years ago by RedHat, fully virtualized, works on newer hardware,
- Initially just gave 100 Mbit/s ethernet, IDE disk,
- Now "virtio" drivers give much better performance,
- Support is in stock RHEL kernel, no alternate kernel needed,
- Possible to overbook memory on a VM host,
  - We regularly run two 16 Gbyte VMs on a system that only has 24 Gbytes of physical memory without swapping!
- Possible to see real memory and cpu usage from "top" on a VM host,
- Some performance issues:
  - Particularly on complex I/O tasks like MySQL, Root, Lustre server, etc.
  - Expect that this will improve with subsequent releases.

Hypervisor Evaluation, Continued

- Commercial hypervisors:
  - VMware is cost-prohibitive for 50-slot cloud,
  - Commercialized Xen products also available:
    - Oracle VM, commercial HVM Xen-based solution, used by FEF on GPCF,
    - Citrix XenServer, and its open-source cousin XCP,
  - Commercial hypervisors certainly have their place but features are gradually moving to their open-source cousins,
  - In a cloud environment, extra bells and whistles of commercial hypervisor usually aren't needed.

- KVM vs. Xen:
  - Past experience has shown that it is difficult to work against RedHat when they pick a technology winner,
  - We will deploy most of FermiCloud on KVM, but will keep the capacity to run Xen for I/O intensive applications.

Eucalyptus Evaluation

- Philosophy:
  - Produce a open-source emulation of Amazon EC2 cloud,
  - Cloud and cluster controllers for overall control, Node controller on each node that hosts VM's.
- Strengths:
  - Most complete implementation of Amazon EC2 API's, Emulates Amazon's S3 and EBS storage API's as well,
  - Cleanly packaged software (RPMS), Easy to deploy a small installation,
  - Web GUI support via HybridFox 3rd party browser addon.
- Weaknesses:
  - Protocols are scalable in theory but not the way Eucalyptus implemented them,
  - Most network traffic and disk traffic goes through cluster controller - single bottleneck and single point of failure,
  - When cluster controller reboots all VM's are lost,
  - Not flexible in the kind of VM's you can create,
  - Uses x509 authentication on SOAP API but with self-signed SimpleCA certs and passwordless keys,
  - Developers promise scalability improvements but only in enterprise version ($$$),
  - Developers refuse to make any changes that break compatibility with EC2,
  - Takes manual operation to save state of running VM,
  - No notion of scheduling at all.

Nimbus Evaluation

- Philosophy:
  - Grows out of Globus Virtual Workspace project,
  - Includes a Globus WSRF interface to take grid certificates,
  - Project dedicated to enabling science users to use "science clouds" both at university and lab facilities and on EC2.
- Strengths:
  - Has Globus WSRF frontend that handles grid certificates,
  - Has notions of user and group quotas,
  - Has notion of machine reservations,
  - Can launch virtual machines via pilot jobs into a batch system,
  - Has context broker for easy coordination of cluster launches,
  - Developers are local and eager to collaborate.
- Weaknesses:
  - Documentation of early versions was exasperating, dozens of little gotchas. Most have been fixed in version 2.6 but some examples still don't all work right,
  - Have to open up lots of permissions on libvirt sockets and in sudoers to get things to work right,
  - Default installation dependent on SimpleCA certificate authority and passwordless private keys, provides way to swap them out.

OpenNebula Evaluation

- Philosophy:
  - OpenNebula is part of EU Reservoir project,
  - Started as a virtual infrastructure manager and added cloud API's afterwards.
- Strengths:
  - Most flexibility in making the virtual machines we want,
  - Large developer and user base,
  - Proven performance at HEP-lab scale at CERN,
  - Good scheduling features,
  - Least sysadmin time required to install it,
  - Fewest single points of failure and network bottlenecks,
  - Most robust operations, daemons run well, recover after reboot.
- Weaknesses:
  - Default security is wide-open,
  - Has "pluggable authentication module." You bring the plug,
  - Limited Amazon ReST API functionality, no Amazon SOAP API (but this is promised in future releases).

Current FermiCloud Deployments

- 8 nodes deployed in OpenNebula,
- 7 nodes deployed in Nimbus,
- 3 nodes deployed in Eucalyptus,
- 4 nodes dedicated to storage investigations.

- Persistent virtual machines running include:
  - GUMS (OSG Grid User Mapping Service) servers,
  - SAZ (Fermilab Site AuthoriZation Services) servers,
  - MySQL servers,
  - MCAS (Metrics Correlation and Analysis Service) servers,
  - dCache servers,
  - JDEM/WFIRST machines.

- We supply a sample virtual machine OS install and a template to start a virtual machine.

FermiCloud Phase 2 - Authentication/Authorization

- Fermilab developers have written and activated an x509 authentication plugin for OpenNebula,
- Works with CLI and EC2 ReST (Query) API,
- Currently beta testing with patched version of pre-release OpenNebula 2.2,
- Have back-contributed our patches to the OpenNebula source tree,
- Will deploy in production when OpenNebula 2.2 is officially released, any day now.

FermiCloud Phase 2 - Monitoring and Backfill

- Goals:
  - Make sure the machines that are supposed to stay up all the time stay up,
  - Make sure the appropriate cloud daemons are running and load of the head nodes is reasonable,
  - Detect idle virtual machines and pause them based on policy,
  - Fill in with worker node VM's and take jobs from the grid.
- Nimbus 2.7 already claims to have idle VM detection.
- Will evaluate this functionality, explore other generic methods of idle VM detection as well.

FermiCloud Phase 2

- Complete the Storage evaluation,
  - See talk by G. Garzoglio et al.
- Develop (if necessary) and deploy an automated provisioning and patching infrastructure,
- Significant amount of work on security policy,
- Explore infiniband and MPI,
- Image repository,
- Live migration,
- OSG collaboration.

Further Work (Phase 3)

- Evaluation of OpenStack:
  - The OpenStack "out of the box" storage model, does not appear to play well with the posix I/O utilized by typical HEP applications,
  - The recent announcement of commercialized OpenStack is a concern.
- Evaluation of Scientific Linux 6:
  - Verify compatibility with current open source cloud frameworks,
  - Measure I/O performance of kvm VMs.
- Start the planning for FermiCloud-HA (high availability):
  - FermiCloud systems are physically located in one building,
  - Possibly moving ~half of the systems to another building,
  - Looking at various storage options to support FermiCloud-HA.

Conclusions

- FermiCloud has completed our technology evaluation and requirements gathering phase.
- "Out of the box", no single open source cloud framework meets all of our requirements:
  - The OpenNebula framework does the best at meeting our requirements,
  - We are focusing on OpenNebula and expect to address the remainder of our requirements by a combination of collaboration with the OpenNebula developers and the Fermilab developers,
  - We will keep a eye on the other open source frameworks.
- We have deployed a pilot service which is gaining increased use by developers and integrators.
- We have a robust program of work defined to transform our pilot service into a production service.
- We welcome interest from new users and stakeholders.

Fin

- Any Questions?