FermiCloud

Steven C. Timm
Fermilab Grid Facilities Dept.
Workshop on Adapting Applications And Computing Services to Virtualization and Multi-core
CERN
June 22, 2010

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
  - Took large amount of sysadmin support and power
  - There weren't enough of them, and they didn't have enough RAM.
- FermiCloud was already in planning when 2 power crunches 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 all worker nodes or accepting Virtual Machines as grid jobs.

Stakeholders and Early Adopters

- Joint Dark Energy Mission
  - Distributed messaging system, testing fault tolerance, ideal application for cloud
- Grid Department Developers
  - Authentication/Authorization
  - Storage evaluation
  - Monitoring/MCAS
  - GlideinWMS
- dCache Developers
- LQCD testbed

Project Plan

- Technology Evaluation
  - Hypervisors—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

Hypervisor Evaluation

- 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
- 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.
- 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, Sci. Linux 4, 5), plus 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

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

Common Cloud concepts

- Overall User Interface for requesting a VM (cloud controller)
- One or more Cluster Controllers which control a group of nodes
- A Node Controller on each node which can run virtual machines
- A repository of virtual image files
- Lack of useful documentation

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 addons for Amazon EC2, including HybridFox GUI
  - Supports a number of different network topologies
  - This was the first one that worked for us.
- 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.
  - 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 sysadmin time required to install it.
- Weaknesses
  - Express documentation geared to a couple specific use cases, it was hard to generalize to get something working

What we have now

- Eucalyptus: 1 controller node and 2 new execution nodes, all Xen for now. KVM coming next week.
- First stakeholder, JDEM, has begun work.
- Contextualization (Kerberos keytabs, etc) is manual for now.
- Exploring managed network topology and instance metadata
- Nimbus and OpenNebula working too
- Will eventually deploy on 23 nodes = 368 logical cores

Contextualization

- 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.
- Don't want to store any machine-specific secrets (Kerberos keytab, host and http certs) in the machine repository, rather put them on as the machine comes up. Will have to customize their wrappers or write our own.
- All cloud provisioning techniques use ssh key-pair for initial contact. This is illegal at Fermilab—have to modify routines that inject the ssh keypair to inject or create the right kerberos magic.

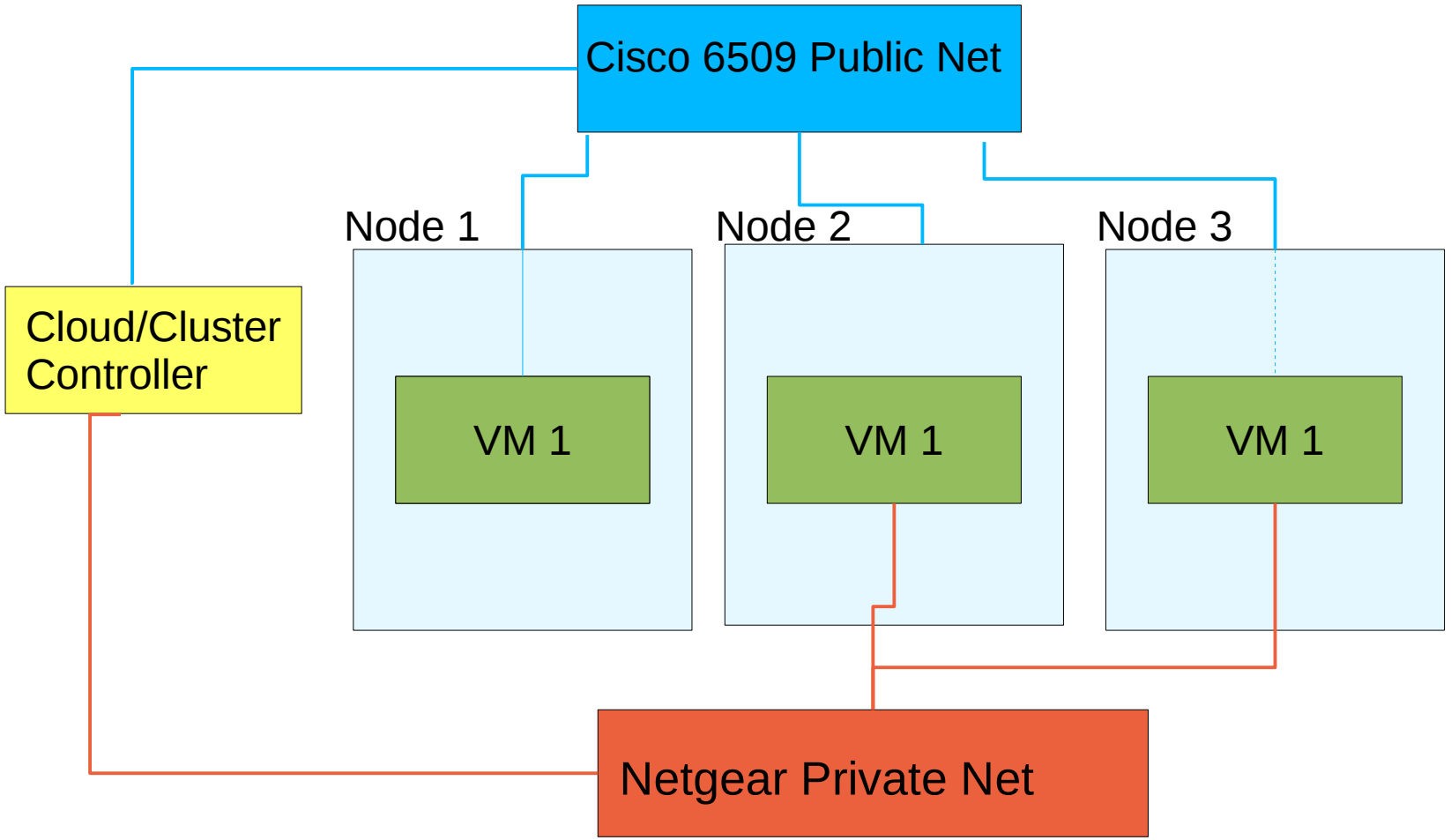
Provisioning and patching

- We want a “kickstart-me-now” feature in which a user can request a PXE boot and a clean kickstart install of Sci. Linux with a user-specified KS file.
- Leverage Fermilab's 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 machines up from time to time and give them their patches
- Both Eucalyptus and Nimbus give only the system admins 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 and Hadoop both on bare iron and under KVM
- 8-core machine probably unnecessary to serve 10TB of disk, hope to allocate 2 cores to serve storage and let the rest of them be compute virtual machines.
- Testing with actual neutrino experiment “root” application, benchmarking different solutions.

Network topology



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.

Phase 2

- FermiCloud Phase 2 already approved!
- Target is small low-cpu-load production servers
- Grid gatekeepers, forwarding nodes, small databases, monitoring, etc.
- Many other server purchases will be directed to FermiCloud for resources.
- Live migration becomes important for this phase.
- Expect to go live with first of these services in October.

Conclusions

- FermiCloud has evaluated three cloud technologies thus far, Nimbus, Eucalyptus, and OpenNebula
- All do what they claim to do—now we are in the next phase of letting early adopters try them out and see what works.
- Expect to make final decision later this summer on which one we push forward to modify and customize.
- All three are still in early phases of development, will try to stay as generic as possible so we don't have vendor lock-in.