# High Throughput & Resilient Fabric Deployments on FermiCloud

Steven C. Timm
Grid & Cloud Services Department
Fermilab

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

Fermilab

# Outline

Introduction: Fermilab and High-Throughput Computing

HPC vs. HTC

Drivers for FermiCloud Project

Virtualized MPI

Virtualized File Systems

Resilience

Summary and Acknowledgements

# Fermilab and HTC

Fermi National Accelerator Laboratory:

- Lead United States particle physics laboratory
- ~60 PB of data on tape
- High Throughput Computing characterized by:
  - “Pleasingly parallel” tasks
  - High CPU instruction / Bytes IO ratio
  - But still lots of I/O.  See Pfister: “In Search of Clusters”
- Fermilab does HPC too, see http://www.usqcd.org/fnal for description of Lattice QCD

# Grid and Cloud Services

Operations:

Grid Authorization

Grid Accounting

Computing Elements

Batch Submission

All require high availability

All require multiple integration systems to test.

Also requires virtualization

And access as root.

Solutions:

Development of authorization, accounting, and batch submission software

Packaging and integration

Requires development machines not used all the time

Plus environments that are easily reset

And access as root

# HTC Virtualization Drivers

Large multi-core servers have evolved from from 2 to 64 cores per box,

- A single “rogue” user/application can impact 63 other users/applications.
- Virtualization can provide mechanisms to securely isolate users/applications.

Typical “bare metal” hardware has significantly more performance than usually needed for a single-purpose server,

- Virtualization can provide mechanisms to harvest/utilize the remaining cycles.

Complicated software stacks are difficult to distribute on grid,

- Distribution of preconfigured virtual machines together with GlideinWMS and HTCondor can aid in addressing this problem.

Large demand for transient development/testing/integration work,

- Virtual machines are ideal for this work.

Science is increasingly turning to complex, multiphase workflows.

- Virtualization coupled with cloud can provide the ability to flexibly reconfigure hardware “on demand” to meet the changing needs of science.

Legacy code:

- Data and code preservation for recently-completed experiments at Fermilab Tevatron and elsewhere.

Burst Capacity:

- Systems are full all the time, need more cycles just before conferences.

# HPC Virtualization Drivers

Dependent on high-performance shared file systems (GPFS, Lustre)

Heterogeneous hardware (Phi, GPU, etc)

Need ways to dynamically configure infrastructure

Sandbox small users to keep them from crashing big cluster

But what about the performance hit?

# FermiCloud – Initial Project Specifications

FermiCloud Project was established in 2009 with the goal of developing and establishing Scientific Cloud capabilities for the Fermilab Scientific Program,

- Building on the very successful FermiGrid program that supports the full Fermilab user community and makes significant contributions as members of the Open Science Grid Consortium.
- Reuse High Availabilty, AuthZ/AuthN, Virtualization from Grid

In a (very) broad brush, the mission of the FermiCloud project is:

- To deploy a production quality Infrastructure as a Service (IaaS) Cloud Computing capability in support of the Fermilab Scientific Program.
- To support additional IaaS, PaaS and SaaS Cloud Computing capabilities based on the FermiCloud infrastructure at Fermilab.

The FermiCloud project is a program of work that is split over several overlapping phases.

- Each phase builds on the capabilities delivered as part of the previous phases.

# Overlapping Phases

Today

**Phase 1:**
"Build and Deploy the Infrastructure"

**Phase 2:**
"Deploy Management Services, Extend the Infrastructure and Research Capabilities"

**Phase 3:**
"Establish Production Services and Evolve System Capabilities in Response to User Needs & Requests"

**Phase 4:**
"Expand the service capabilities to serve more of our user communities"

Time →

# Current FermiCloud Capabilities

The current FermiCloud hardware capabilities include:

- Public network access via the high performance Fermilab network,
  - This is a distributed, redundant network.
- Private 1 Gb/sec network,
  - This network is bridged across FCC and GCC on private fiber,
- High performance Infiniband network,
  - Currently split into two segments,
- Access to a high performance FibreChannel based SAN,
  - This SAN spans both buildings.
- Access to the high performance BlueArc based filesystems,
  - The BlueArc is located on FCC-2,
- Access to the Fermilab dCache and enStore services,
  - These services are split across FCC and GCC,
- Access to 100 Gbit Ethernet test bed in LCC (Integration nodes),
  - Intel 10 Gbit Ethernet converged network adapter X540-T1.

# Typical Use Cases

Public net virtual machine:

- On Fermilab Network open to Internet,
- Can access dCache and Bluearc Mass Storage,
- Common home directory between multiple VM’s.

Public/Private Cluster:

- One gateway VM on public/private net,
- Cluster of many VM’s on private net.
- Data acquisition simulation

Storage VM:

- VM with large non-persistent storage,
- Use for large MySQL or Postgres databases, Lustre/Hadoop/Bestman/xRootd/dCache/OrangeFS/IRODS servers.

# OpenNebula

- OpenNebula was picked as result of evaluation of Open source cloud management software.
- OpenNebula 2.0 pilot system in GCC available to users since November 2010.
- Began with 5 nodes, gradually expanded to 13 nodes.
- 4500 Virtual Machines run on pilot system in 3+ years.
- OpenNebula 3.2 production-quality system installed in FCC in June 2012 in advance of GCC total power outage—now comprises 18 nodes.
- Transition of virtual machines and users from ONe 2.0 pilot system to production system almost complete.
- In the meantime OpenNebula has done five more releases, will catch up shortly.

# FermiCloud – Infiniband & MPI

FermiCloud purchased Mellanox SysConnect II Infiniband adapters in May 2010. These cards were advertised to support SR-IOV virtualization.

We received the SR-IOV drivers from Mellanox as a result of a meeting at SC2011. These were a private patch to the normal open-source OFED drivers.

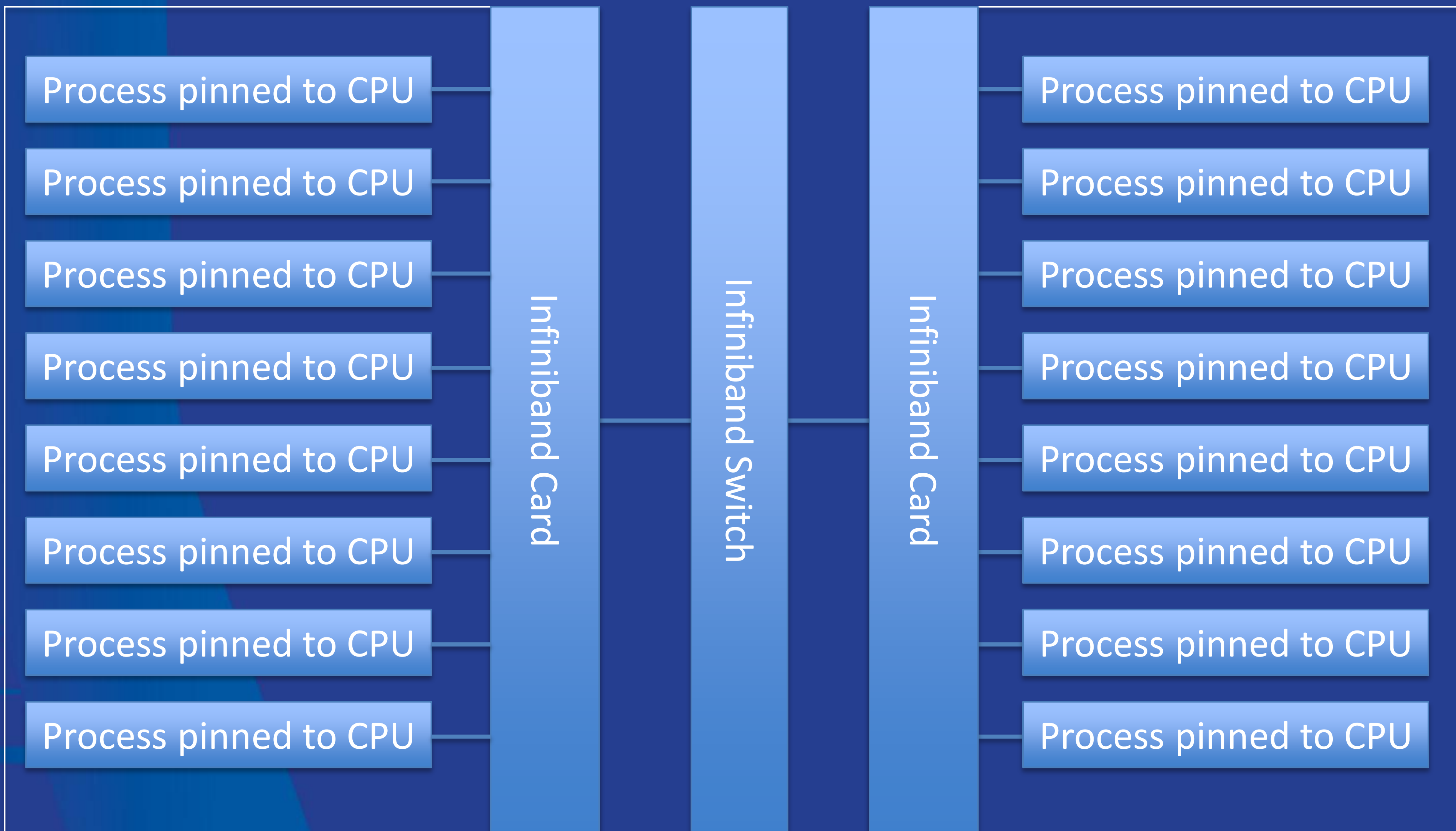
After a couple months of kernel building, MPI recompilation, firmware burning, and finding the right double-pre-alpha libvirt, etc, we were ready to measure.

Using this driver, we were able to make measurements of High Performance Linpack comparing MPI on “bare metal” to MPI on KVM virtual machines on the identical hardware.

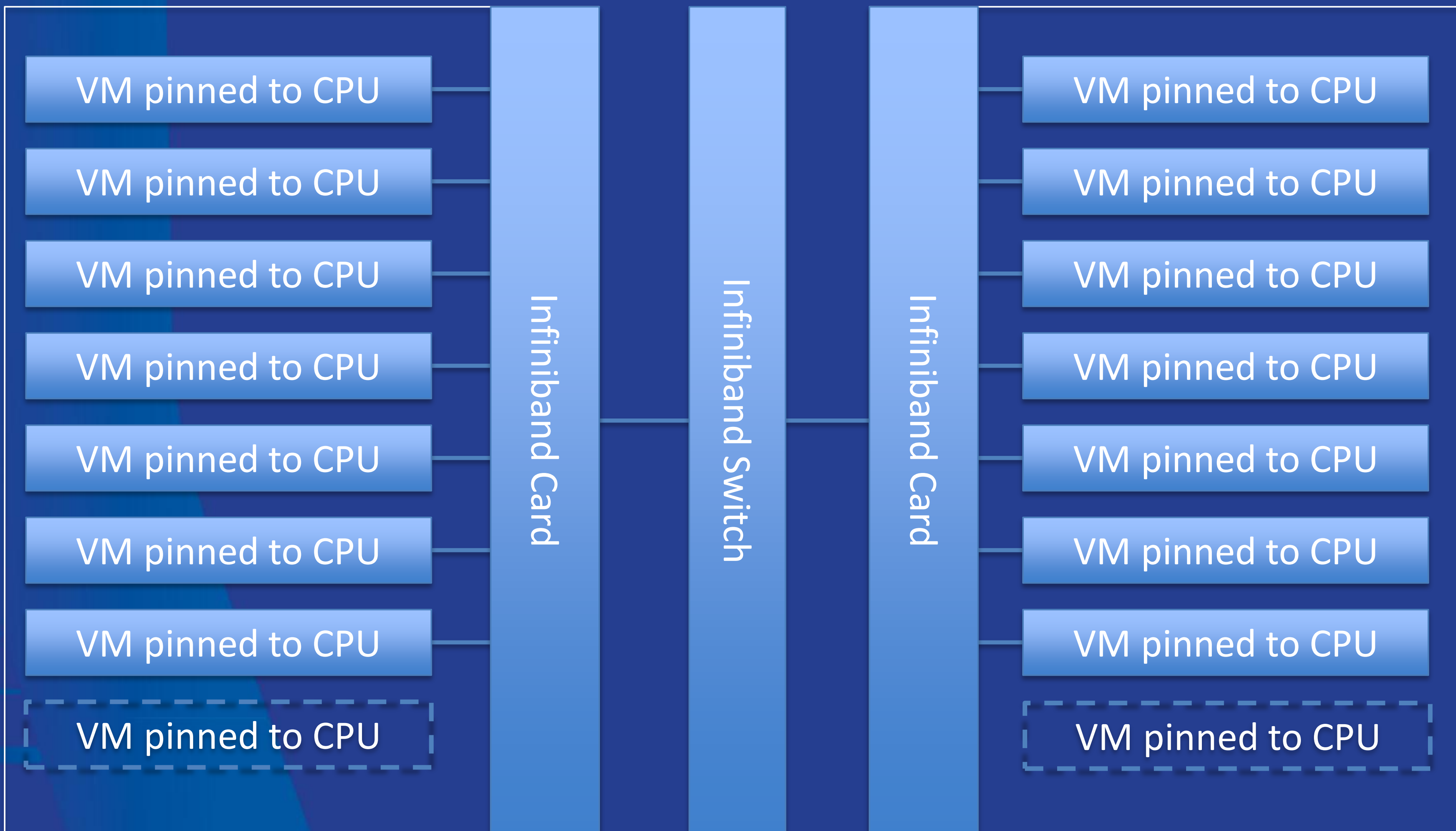
- The next three slides will detail the configurations and measurements that were completed in March 2012.
- MPI is compiled using IP over Infiniband (IPOIB)

This allows a direct measurement of the MPI “virtualization overhead”.

# FermiCloud "Bare Metal MPI"



# FermiCloud "Virtual MPI"



# MPI on FermiCloud (Note 1)

| Configuration | #Host Systems | #VM/host | #CPU | Total Physical CPU | HPL Benchmark (Gflops) | Gflops/Core |
|---|---|---|---|---|---|---|
| Bare Metal without pinning | 2 | -- | 8 | 16 | 13.9 | 0.87 |
| Bare Metal with pinning (Note 2) | 2 | -- | 8 | 16 | 24.5 | 1.53 |
| VM without pinning (Notes 2,3) | 2 | 8 | 1 vCPU | 16 | 8.2 | 0.51 |
| VM with pinning (Notes 2,3) | 2 | 8 | 1 vCPU | 16 | 17.5 | 1.09 |
| VM+SRIOV with pinning (Notes 2,4) | 2 | 7 | 2 vCPU | 14 | 23.6 | 1.69 |

Notes: (1) Work performed by Dr. Hyunwoo Kim of KISTI in collaboration with Dr. Steven Timm of Fermilab.
(2) Process/Virtual Machine "pinned" to CPU and associated NUMA memory via use of numactl.
(3) Software Bridged Virtual Network using IP over IB (seen by Virtual Machine as a virtual Ethernet).
(4) SRIOV driver presents native InfiniBand to virtual machine(s), 2nd virtual CPU is required to start SRIOV, but is only a virtual CPU, not an actual physical CPU.

# Infiniband 2 years later

Now KVM SR-IOV drivers part of standard OFED distribution.

No special kernel or driver build needed

OpenNebula can launch your VM with an Infiniband interface, with no libvirt hackery needed.

Still: be careful burning new firmware on IB Card.

# Virtualized Storage Service Investigation

Motivation:

General purpose systems from various vendors being used as file servers,

Systems can have many more cores than needed to perform the file service,

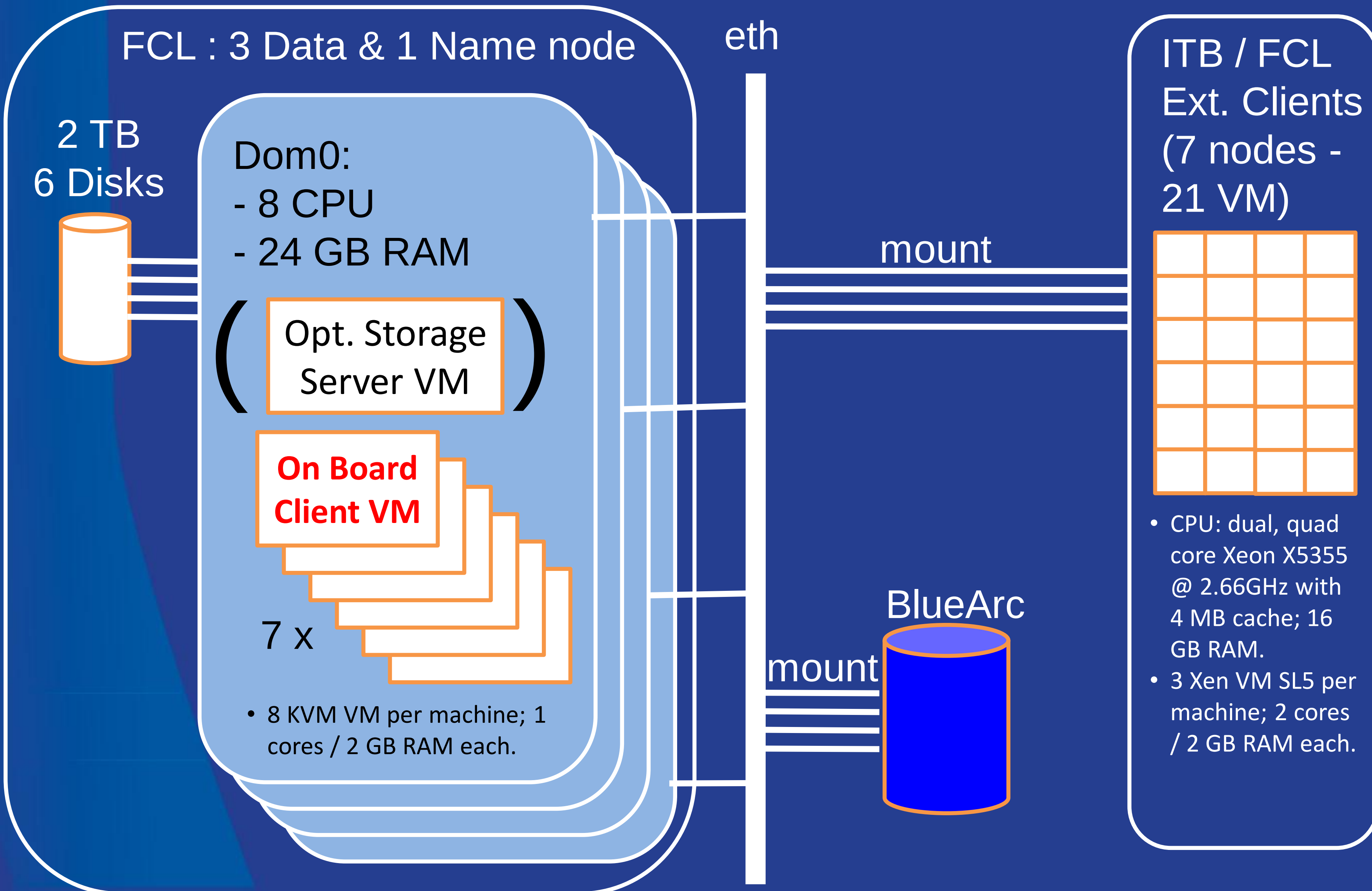
- Cores go unused => Inefficient power, space and cooling usage,
- Custom configurations => Complicates sparing issues.

Question:

Can virtualization help here?

What (if any) is the virtualization penalty?

# FermiCloud Test Bed - Virtualized Server



# Virtualized File Service Results Summary

| FileSystem | Benchmark | Read (MB/s) | “Bare Metal” Write (MB/s) | VM Write (MB/s) | Notes |
|---|---|---|---|---|---|
| Lustre | IOZone | 350 | 250 | 70 | Significant write penalty when FS on VM |
| | Root-based | 12.6 | - | - | |
| Hadoop | IOZone | 50 - 240 | 80 - 300 | 80 - 300 | Varies on number of replicas, fuse does not export a full posix fs. |
| | Root-based | 7.9 | - | - | |
| OrangeFS | IOZone | 150 - 330 | 220 - 350 | 220 - 350 | Varies on number of name nodes |
| | Root-based | 8.1 | - | - | |
| BlueArc | IOZone | 300 | 330 | n/a | Varies on system conditions |
| | Root-based | 8.4 | - | - | |

See ISGC talk for the details - http://indico3.twgrid.org/indico/getFile.py/access?contribId=32&sessionId=36&resId=0&materialId=slides&confId=44

# FermiGrid-HA2 Experience

In 2009, based on operational experience and plans for redevelopment of the FCC-1 computer room, the FermiGrid-HA2 project was established to split the set of FermiGrid services across computer rooms in two separate buildings (FCC-2 and GCC-B).

- This project was completed on 7-Jun-2011 (and tested by a building failure less than two hours later).
- FermiGrid-HA2 worked exactly as designed.

Our operational experience with FermiGrid-HA and FermiGrid-HA2 has shown the benefits of virtualization and service redundancy.

- Benefits to the user community – increased service reliability and uptime.
- Benefits to the service maintainers – flexible scheduling of maintenance and upgrade activities.

# Experience with FermiGrid = Drivers for FermiCloud

Access to pools of resources using common interfaces:

- Monitoring, quotas, allocations, accounting, etc.

Opportunistic access:

- Users can use the common interfaces to “burst” to additional resources to meet their needs

Efficient operations:

- Deploy common services centrally

High availability services:

- Flexible and resilient operations

# Some Recent Major Facility and Network Outages at Fermilab

- FCC main breaker 4x (Feb, Oct 2010)
- FCC-1 network cuts 2x (Spring 2011)
- GCC-B Load shed events (June-Aug 2011)
  - This accelerated planned move of nodes to FCC-3.
- GCC load shed events and maintenance (July 2012).
  - FCC-3 cloud was ready just in time to keep server VM’s up.
- FCC-2 outage (Oct. 2012)
  - FermiCloud wasn’t affected, our VM’s stayed up.

# FermiCloud – Fault Tolerance

As we have learned from FermiGrid, having a distributed fault tolerant infrastructure is highly desirable for production operations.

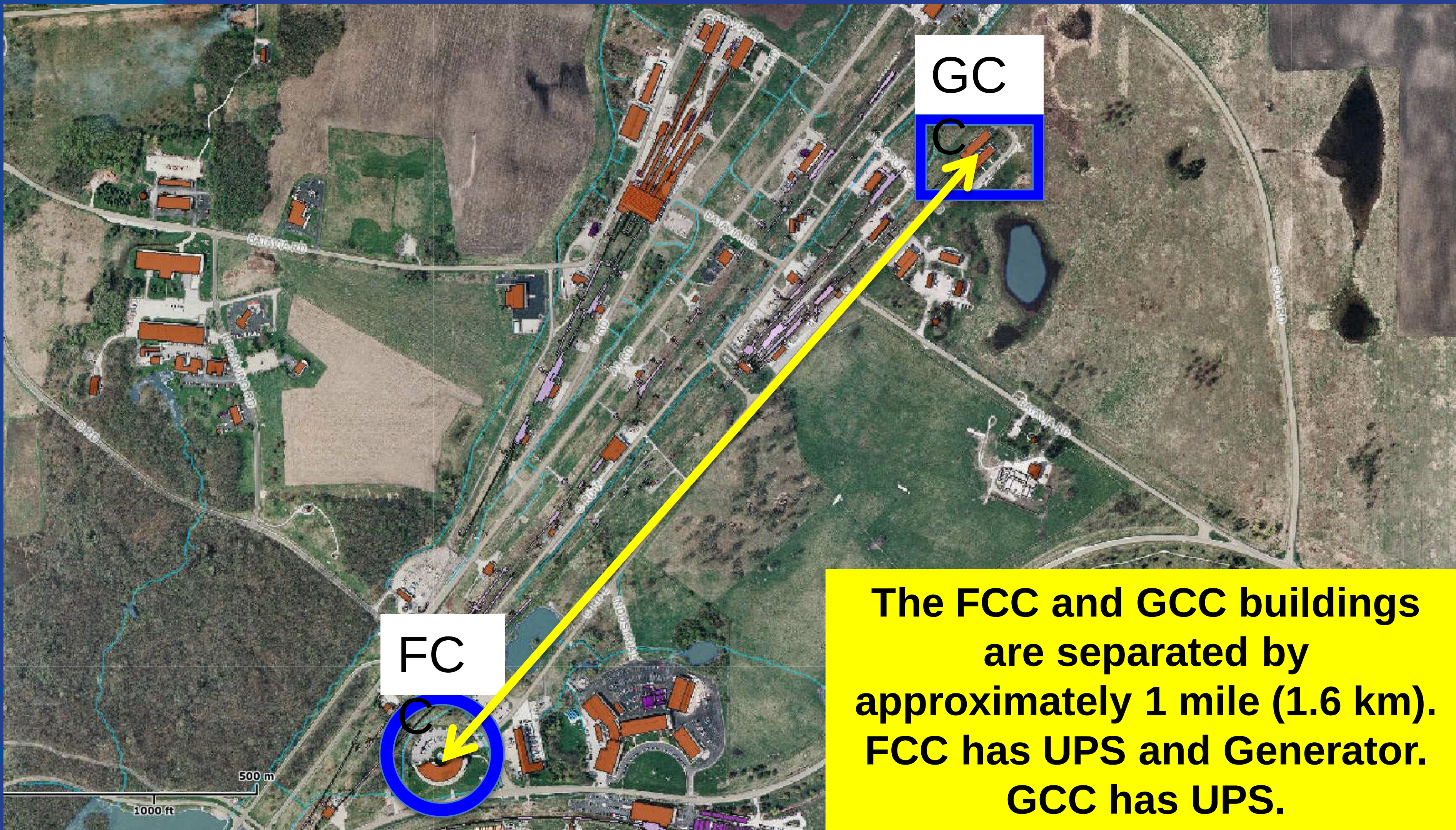
We are actively working on deploying the FermiCloud hardware resources in a fault tolerant infrastructure:

- The physical systems are split across two buildings,
- There is a fault tolerant network infrastructure in place that interconnects the two buildings,
- We have deployed SAN hardware in both buildings,
- We have a dual head-node configuration with failover
- We have a GFS2 + CLVM for our multi-user filesystem and distributed SAN.
- SAN replicated between buildings using CLVM mirroring.

GOAL:

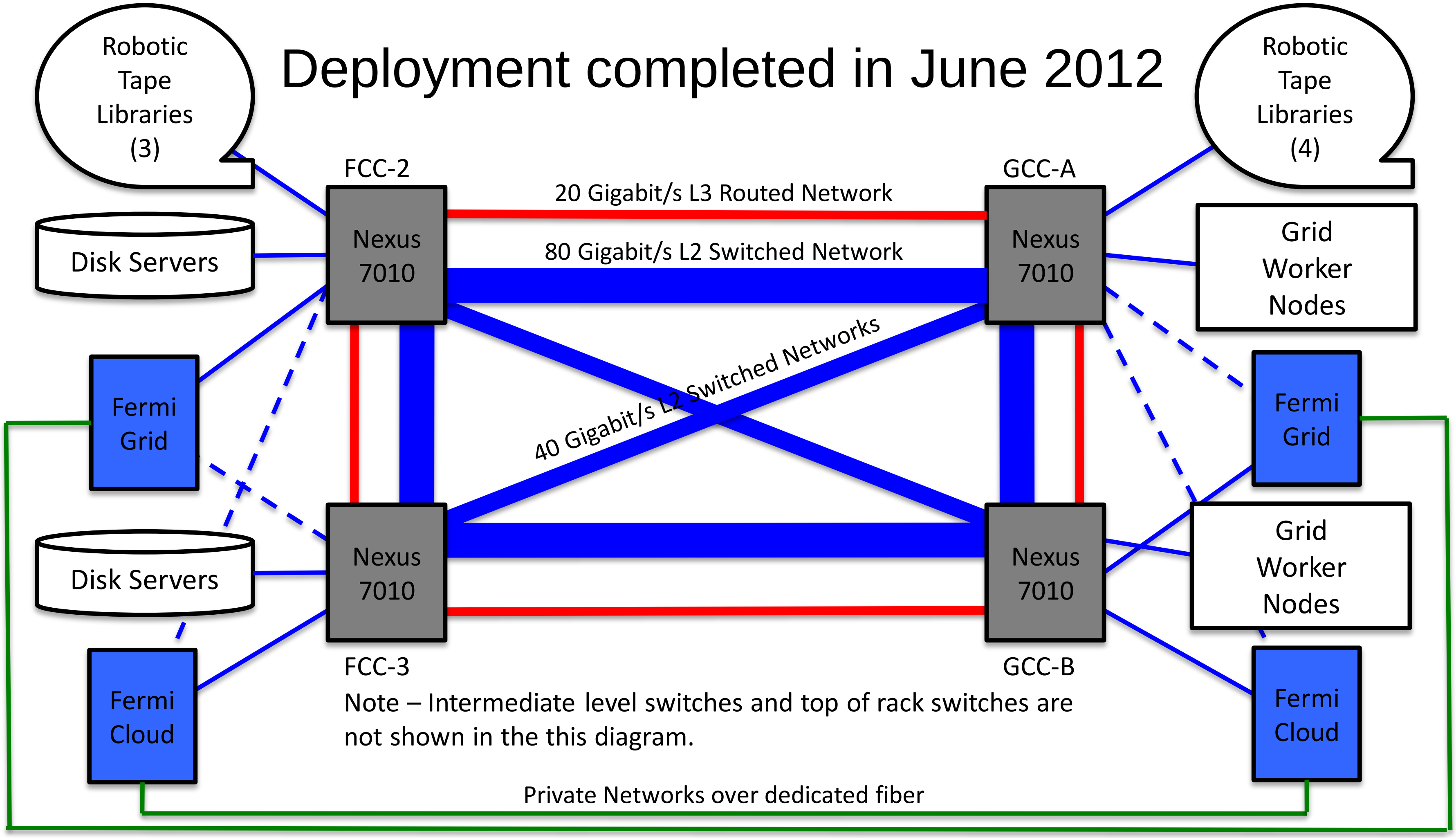
- If a building is “lost”, then automatically relaunch “24x7” VMs on surviving infrastructure, then relaunch “9x5” VMs if there is sufficient remaining capacity,
- Perform notification (via Service-Now) when exceptions are detected.

# FCC and GCC




**The FCC and GCC buildings are separated by approximately 1 mile (1.6 km). FCC has UPS and Generator. GCC has UPS.**

# Distributed Network Core Provides Redundant Connectivity



# Distributed Shared File System

Design:

Dual-port FibreChannel HBA in each node,

Two Brocade SAN switches per rack,

Brocades linked rack-to-rack with dark fiber,

60TB Nexsan Satabeast in FCC-3 and GCC-B,

Redhat Clustering + CLVM + GFS2 used for file system,

Each VM image is a file in the GFS2 file system

LVM mirroring RAID 1 across buildings.

Benefits:

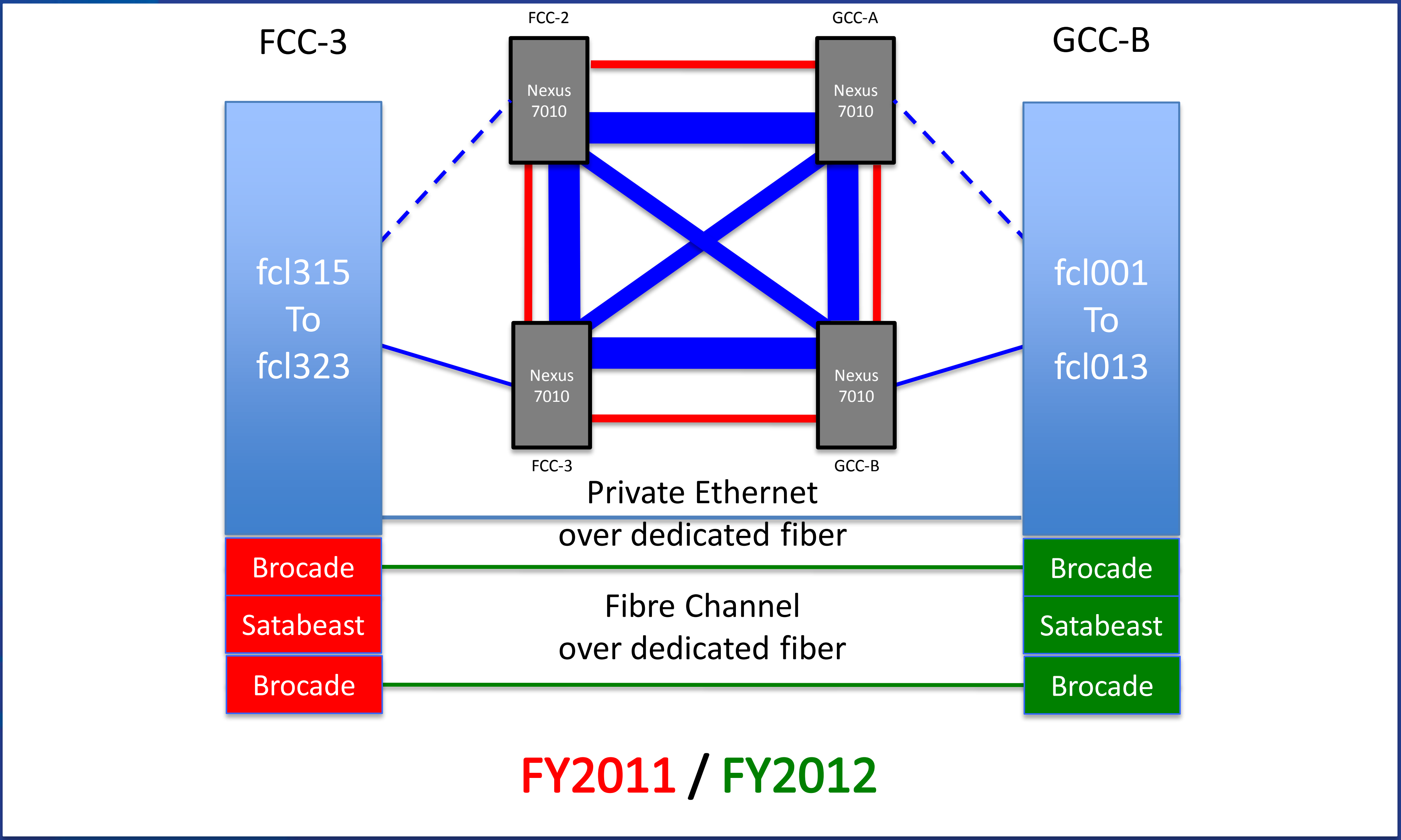
Fast Launch—almost immediate as compared to 3-4 minutes with ssh/scp,

Live Migration—Can move virtual machines from one host to another for scheduled maintenance, transparent to users,

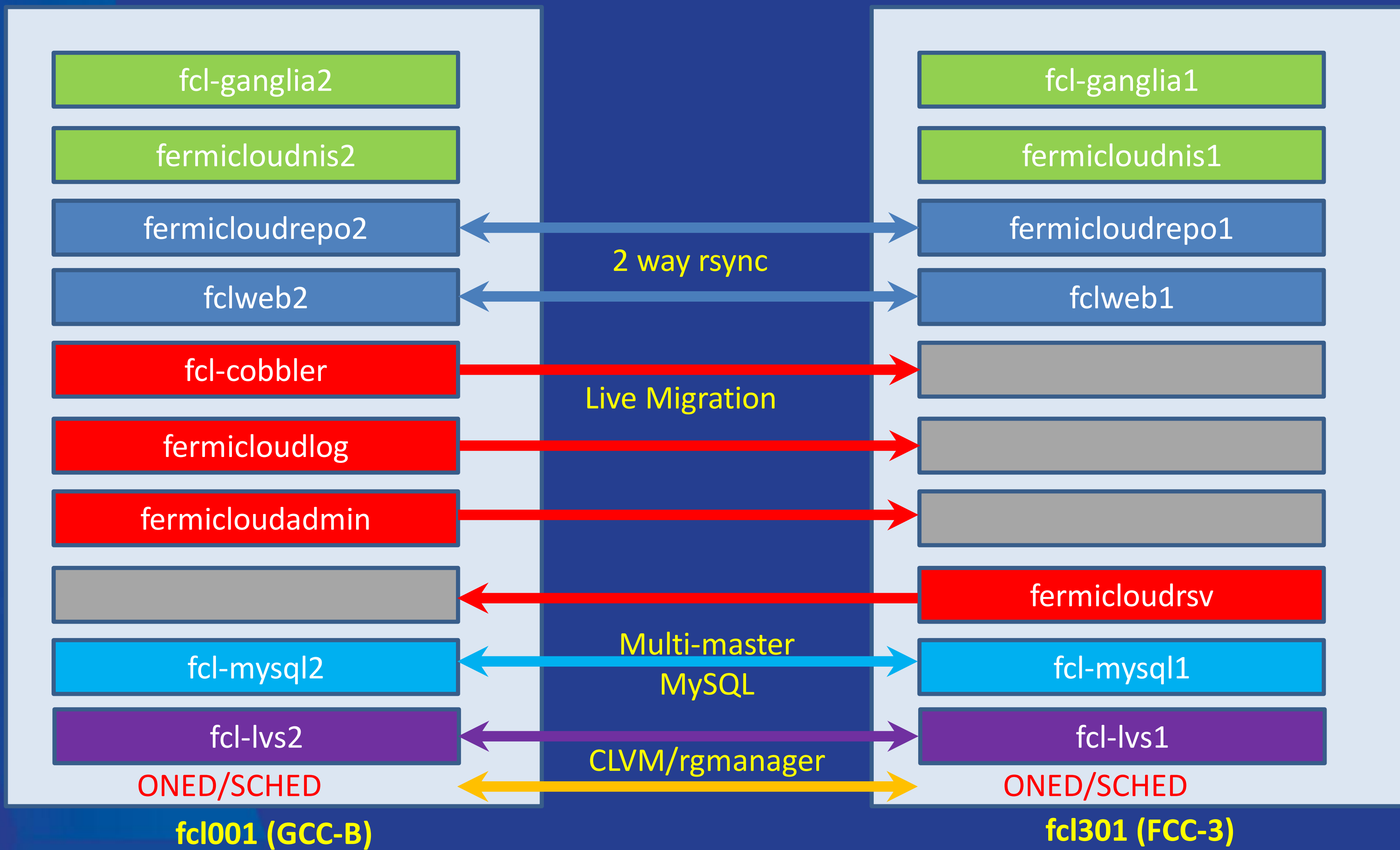
Persistent data volumes—can move quickly with machines,

Can relaunch virtual machines in surviving building in case of building failure/outage,

# FermiCloud – Network & SAN "Today"



# FermiCloud-HA
# Head Node Configuration



# Security

Main areas of cloud security development:

Secure Contextualization:

- Secrets such as X.509 service certificates and Kerberos keytabs are not stored in virtual machines (See following talk for more details).

X.509 Authentication/Authorization:

- X.509 Authentication written by T. Hesselroth, code submitted to and accepted by OpenNebula, publicly available since Jan-2012.

Security Policy:

- A security taskforce met and delivered a report to the Fermilab Computer Security Board, recommending the creation of a new Cloud Computing Environment, now in progress.

We also participated in the HEPiX Virtualisation Task Force,

- We respectfully disagree with the recommendations regarding VM endorsement.

# Current FermiCloud Focus

Virtual Provisioning:

- "Grid Bursting"
  - Extending our grid cluster into FermiCloud and Amazon EC2 via KISTI vcluster utility.
- "Cloud Bursting"
  - Using native OpenNebula burst feature.
  - Using Condor and GlideinWMS.
- Idle VM Detection/suspension

# Current FermiCloud Focus 2

Interoperability:

- User guides on how to convert VM images to and from FermiCloud/OpenNebula
- VirtualBox, VMWare, OpenStack, Amazon EC2, Google Cloud.
- API studies for EC2 compatibility

Scientific Workflows:

- CMS using High Level Trigger farm @ CERN as Cloud.
- GlideinWMS/Condor/OpenStack for reconstruction.
- Replicate success now with NOvA neutrino simulation workflow
- On Amazon EC2, OpenNebula, OpenStack

On-demand Services:

- Scalable batch submission and data movement services.

# Reframing Cloud Discussion

Purpose of Infrastructure-as-a-service:

On demand only?

No—a whole new way to think about IT infrastructure both internal and external.

Cloud API is just a part of rethinking IT infrastructure for data-intensive science (and MIS).

Only as good as the hardware and software it's built on.

Network fabric, storage, and applications all crucial.

Buy or build?

Both!  Will always need some in-house capacity.

Performance hit?

Most can be traced to badly written applications or misconfigured OS.

# FermiCloud Project Summary - 1

Science is directly and indirectly benefiting from FermiCloud:

- CDF, D0, Intensity Frontier, Cosmic Frontier, CMS, ATLAS, Open Science Grid,…

FermiCloud operates at the forefront of delivering cloud computing capabilities to support scientific research:

- By starting small, developing a list of requirements, building on existing Grid knowledge and infrastructure to address those requirements, FermiCloud has managed to deliver a production class Infrastructure as a Service cloud computing capability that supports science at Fermilab.
- FermiCloud has provided FermiGrid with an infrastructure that has allowed us to test Grid middleware at production scale prior to deployment.
- The Open Science Grid software team used FermiCloud resources to support their RPM "refactoring" and is currently using it to support their ongoing middleware development/integration.

# FermiCloud Project Summary - 2

The FermiCloud collaboration with KISTI has leveraged the resources and expertise of both institutions to achieve significant benefits.

vCluster has demonstrated proof of principle “Grid Bursting” using FermiCloud and Amazon EC2 resources.

Using SRIOV drivers on FermiCloud virtual machines, MPI performance has been demonstrated to be >96% of the native “bare metal” performance.

The future is mostly cloudy.

# Acknowledgements

None of this work could have been accomplished without:

- The excellent support from other departments of the Fermilab Computing Sector – including Computing Facilities, Site Networking, and Logistics.
- The excellent collaboration with the open source communities – especially Scientific Linux and OpenNebula,
- As well as the excellent collaboration and contributions from KISTI
- Talented summer students from Illinois Institute of Technology.