FermiGrid-HA
(Delivering Highly Available Grid Services using Virtualisation)

Keith Chadwick

Fermilab

chadwick@fnal.gov

Outline

Who is FermiGrid?

What is FermiGrid?

- Current Architecture & Performance

Why FermiGrid-HA?

- Reasons, Goals, etc.

What is FermiGrid-HA?

FermiGrid-HA Implementation

- Design, Technology, Challenges, Deployment & Performance

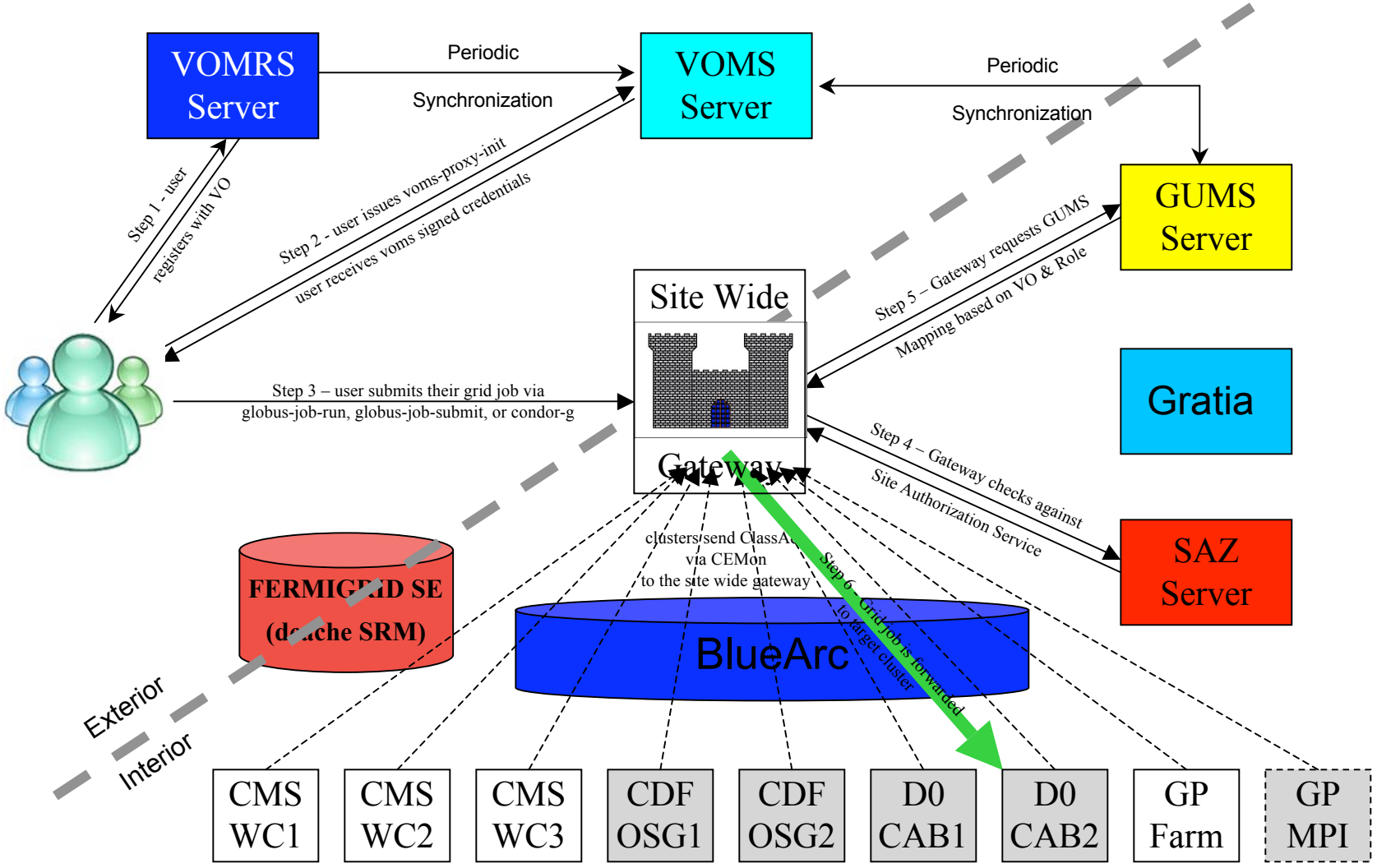
Conclusions

Future Work

FermiGrid - Personnel

Eileen Berman, Fermilab, Batavia, IL 60510	berman@fnal.gov	
Philippe Canal, Fermilab, Batavia, IL 60510	pcanal@fnal.gov	
Keith Chadwick, Fermilab, Batavia, IL 60510	chadwick@fnal.gov	*
David Dykstra, Fermilab, Batavia, IL 60510	dwd@fnal.gov	
Ted Hesselroth, Fermilab, Batavia, IL, 60510	tdh@fnal.gov	
Gabriele Garzoglio, Fermilab, Batavia, IL 60510	garzogli@fnal.gov	
Chris Green, Fermilab, Batavia, IL 60510	greenc@fnal.gov	
Tanya Levshina, Fermilab, Batavia, IL 60510	tlevshin@fnal.gov	
Don Petravick, Fermilab, Batavia, IL 60510	petravick@fnal.gov	
Ruth Pordes, Fermilab, Batavia, IL 60510	ruth@fnal.gov	
Valery Sergeev, Fermilab, Batavia, IL 60510	sergeev@fnal.gov	*
Igor Sfiligoi, Fermilab, Batavia, IL 60510	sfiligoi@fnal.gov	
Neha Sharma Batavia, IL 60510	neha@fnal.gov	*
Steven Timm, Fermilab, Batavia, IL 60510	timm@fnal.gov	*
D.R. Yocum, Fermilab, Batavia, IL 60510	yocum@fnal.gov	*

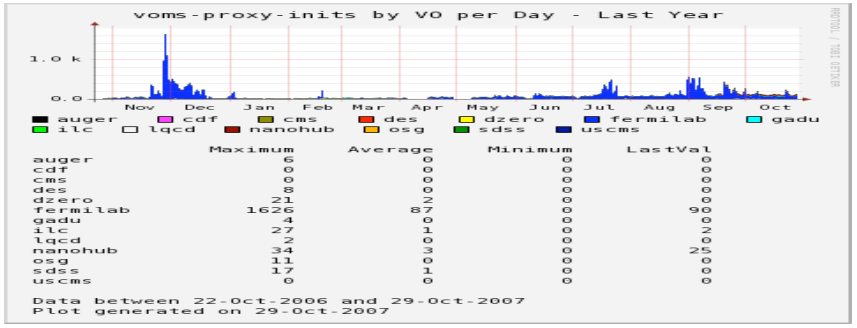
FermiGrid - Current Architecture



FermiGrid - Current Performance

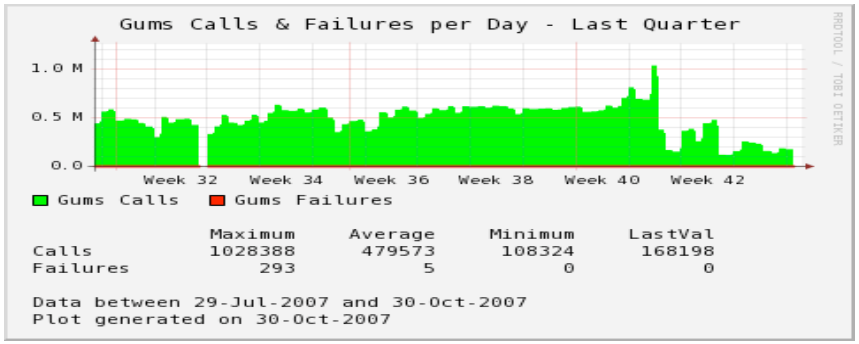
VOMS:

- Current record ~1700 voms-proxy-inits/day.
- Not a driver for FermiGrid-HA.



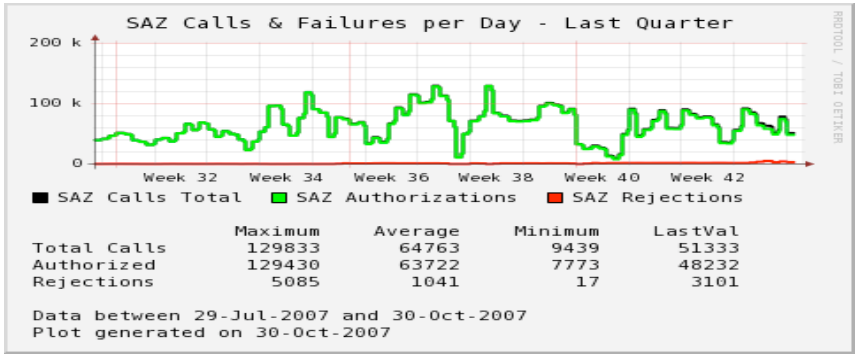
GUMS:

- Current record > 1M mapping requests/day
- Maximum system load <3 at a CPU utilization of 130% (max 200%)



SAZ:

- Current record > 129K authorization decisions/day.
- Maximum system load <5.



Why FermiGrid-HA?

FermiGrid core services (GUMS and/or SAZ) control access to:

- Over 2000 systems with more than 9000 batch slots (today).
- Petabytes of storage (via gPlazma which calls GUMS).

An outage of either GUMS or SAZ can cause 5,000 to 50,000 “jobs” to fail for each hour of downtime.

Manual recovery or intervention for these services can have long recovery times (best case 30 minutes, worst case multiple hours).

Automated service recovery scripts can minimize the downtime (and impact to the Grid operations), but still can have several tens of minutes response time for failures:

- How often the scripts run,
- Scripts can only deal with failures that have known “signatures”,
- Startup time for the service,
- A script cannot fix dead hardware.

FermiGrid-HA - Requirements

Requirements:

- Critical services hosted on multiple systems (n ≥ 2).
- Small number of "dropped" transactions when failover required (ideally 0).
- Support the use of service aliases:
  - VOMS: fermigrid2.fnal.gov -> voms.fnal.gov
  - GUMS: fermigrid3.fnal.gov -> gums.fnal.gov
  - SAZ: fermigrid4.fnal.gov -> saz.fnal.gov
- Implement "HA" services with services that did not include "HA" in their design.
  - Without modification of the underlying service.

Desirables:

- Active-Active service configuration.
- Active-Standby if Active-Active is too difficult to implement.
- A design which can be extended to provide redundant services.

FermiGrid-HA - Technology

Xen:

- SL 5.0 + Xen 3.1.0 (from xensource community version)
  - 64 bit Xen Domain 0 host, 32 and 64 bit Xen VMs
- Paravirtualisation.

Linux Virtual Server (LVS 1.38):

- Shipped with Piranha V0.8.4 from Redhat.

Grid Middleware:

- Virtual Data Toolkit (VDT 1.8.1)
- VOMS V1.7.20, GUMS V1.2.10, SAZ V1.9.2

MySQL:

- Multi-master database replication.

FermiGrid-HA - Challenges #1

Active-Standby:

- Easier to implement,
- Can result in "lost" transactions to the backend databases,
- Lost transactions would then result in potential inconsistencies following a failover or unexpected configuration changes due to the "lost" transactions.
  - GUMS Pool Account Mappings.
  - SAZ Whitelist and Blacklist changes.

Active-Active:

- Significantly harder to implement (correctly!).
- Allows a greater "transparency".
- Reduces the risk of a "lost" transaction, since any transactions which results in a change to the underlying MySQL databases are "immediately" replicated to the other service instance.
- Very low likelihood of inconsistencies.
  - Any service failure is highly correlated in time with the process which performs the change.

FermiGrid-HA - Challenges #2

DNS:

- Initial FermiGrid-HA design called for DNS names each of which would resolve to two (or more) IP numbers.
- If a service instance failed, the surviving service instance could restore operations by "migrating" the IP number for the failed instance to the Ethernet interface of the surviving instance.
- Unfortunately, the tool used to build the DNS configuration for the Fermilab network did not support DNS names resolving to >1 IP numbers.
  - Back to the drawing board.

Linux Virtual Server (LVS):

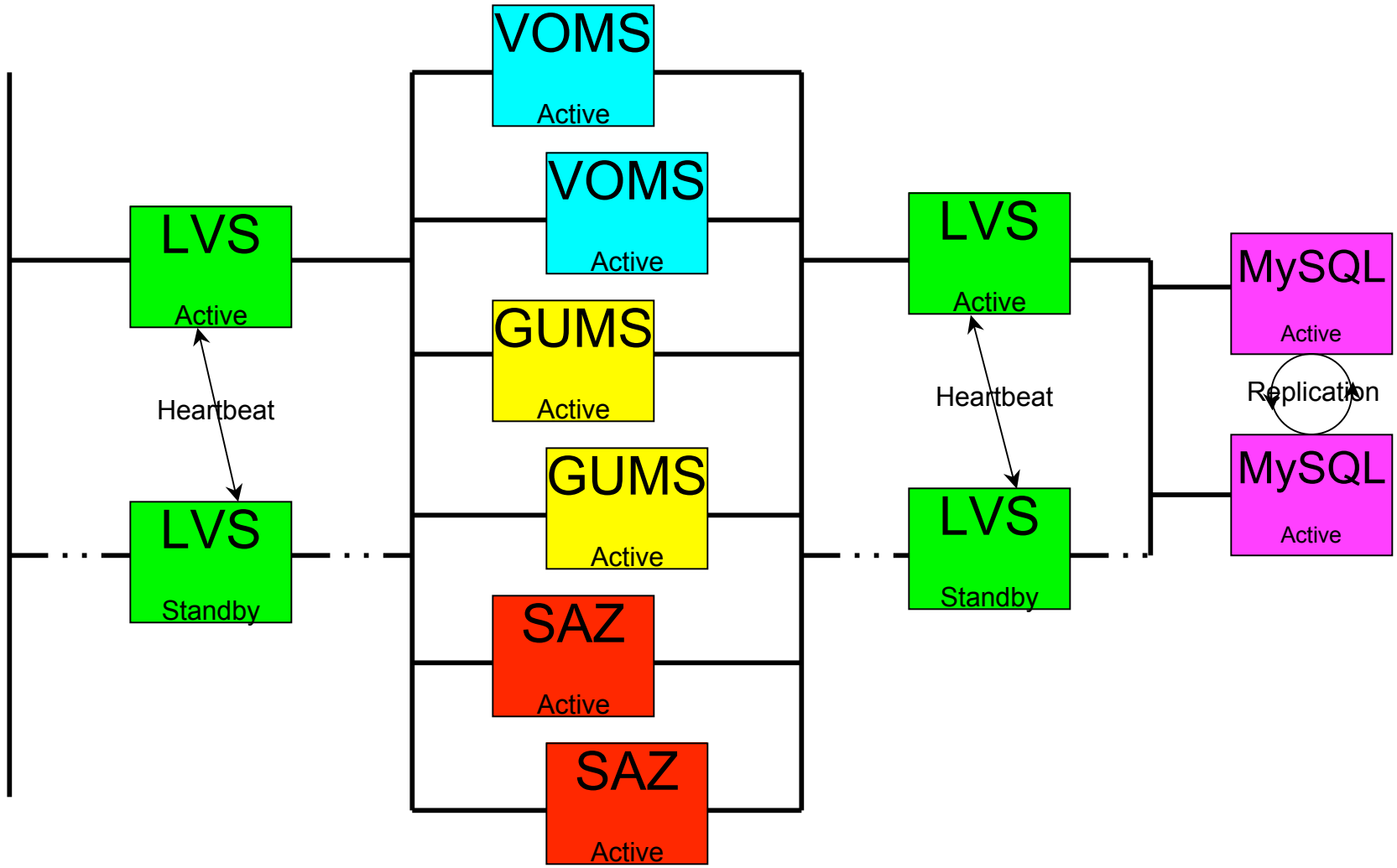
- Route all IP connections through a system configured as a Linux virtual server.
  - Direct routing
  - Request goes to LVS director, LVS director redirects the packets to the real server, real server replies directly to the client.
- Increases complexity, parts and system count:
  - More chances for things to fail.
- LVS director must be implemented as a HA service.
  - LVS director implemented as an Active-Standby HA service.
  - Run LVS director as a special process on the Xen Domain 0 system.
- LVS director performs "service pings" every six (6) seconds to verify service availability.
  - Custom script that uses curl for each service.

FermiGrid-HA - Challenges #3

MySQL databases underlie all of the FermiGrid-HA Services (VOMS, GUMS, SAZ):

- Fortunately all of these Grid services employ relatively simple database schema,
- Utilize multi-master MySQL replication,
  - Requires MySQL 5.0 (or greater).
  - Databases perform circular replication.
- Currently have two (2) MySQL databases,
  - MySQL 5.0 circular replication has been shown to scale up to ten (10).
  - Failed databases “cut” the circle and the database circle must be “retied”.
- Transactions to either MySQL database are replicated to the other database within 1.1 milliseconds (measured),
- Tables which include auto incrementing column fields are handled with the following MySQL 5.0 configuration entries:
  - auto_increment_offset (1, 2, 3, … n)
  - auto_increment_increment (10, 10, 10, …)

FermiGrid-HA - Component Design



FermiGrid-HA - Host Configuration

The fermigrid5&6 Xen hosts are Dell 2950 systems.

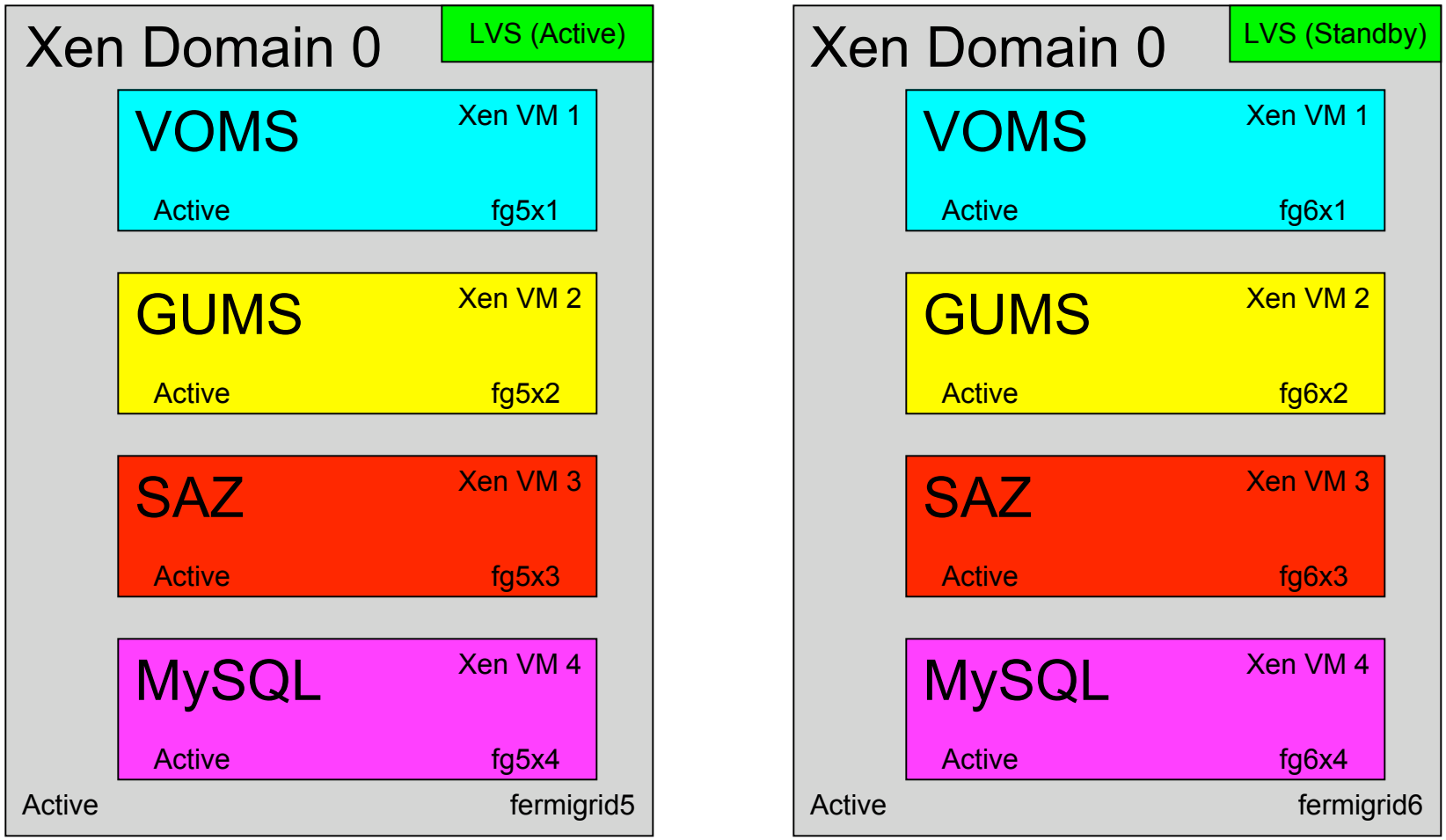
Each of the Dell 2950s are configured with:

- Two 3.0 GHz core 2 duo processors (total 4 cores).
- 16 Gbytes of RAM.
- Raid-1 system disks (2 x 147 Gbytes, 10K RPM, SAS).
- Raid-1 non-system disks (2 x 147 Gbytes, 10K RPM, SAS).
- Dual 1 Gig-E interfaces:
  - 1 connected to public network,
  - 1 connected to private network.

System Software Configuration:

- LVS Director is run on the Xen Domain 0s.
- Each Domain 0 system is configured with 4 Xen VMs.
- Each Xen VM, dedicated to running a specific service:
  - VOMS, GUMS, SAZ, MySQL

FermiGrid-HA - Actual Component Deployment



FermiGrid-HA - Performance #1

Stress tests of the FermiGrid-HA GUMS deployment:

- The initial stress test demonstrated that this configuration can support >4.3M mappings/day.
  - The load on the GUMS VMs during this stress test was ~1.2 and the CPU idle time was 60%.
  - The load on the backend MySQL database VM during this stress test was under 1 and the CPU idle time was 92%.
- A second stress test demonstrated that this configuration can support ~9.7M mappings/day.
  - The load on the GUMS VMs during this stress test was ~9.5 and the CPU idle time was 15%.
  - The load on the backend MySQL database VM during this stress test was under 1 and the CPU idle time was 92%.
- GUMS uses hibernate which is why the backend MySQL database VM load did not increase between the two measurements.
- Based on these measurements, we'll need to start the planning for a third GUMS server in FermiGrid-HA when we hit the ~7.5M mappings/day mark.

FermiGrid-HA - Performance #2

Stress tests of the FermiGrid-HA SAZ deployment:

- The SAZ stress test demonstrated that this configuration can support ~1.1M authorizations/day.
  - The load on the SAZ VMs during this stress test was ~12 and the CPU idle time was 0%.
  - The load on the backend MySQL database VM during this stress test was under 1 and the CPU idle time was 98%.
- The SAZ server does not (currently) use hibernate.
  - This change is “in the works”.
- The SAZ server (currently) performs a significant amount of parsing of the users proxy to identify the DN, VO, Role & CA.
  - This will change as we integrate SAZ into the Globus AuthZ framework.
  - The distributed SAZ clients will perform the parsing of the users proxy.
- We also take a careful look at the SAZ server to see if there are optimizations that can be performed to improve the performance.

FermiGrid-HA - Performance #3

Stress tests of the combined FermiGrid-HA GUMS and SAZ deployment:

- Using a GUMS:SAZ call ratio of ~7:1
- The combined GUMS-SAZ stress test which was performed yesterday (06-Nov-2007) demonstrated that this configuration can support ~6.5 GUMS mappings/day and ~900K authorizations/day.
  - The load on the SAZ VMs during this stress test was ~12 and the CPU idle time was 0%.

FermiGrid-HA - Production Deployment

Our plan is to complete the FermiGrid-HA stress testing and deploy FermiGrid-HA in production during the week of 03-Dec-2007.

In order to allow an adiabatic transition for the OSG and our user community, we will run the regular FermiGrid services and FermiGrid-HA services simultaneously for a three month period.

FermiGrid-HA - Future Plans

Redundant side wide gatekeeper:

- We have a preliminary “Gatekeeper-HA” design...

We will also be installing a test gatekeeper to receive Xen VMs as Grid jobs and execute them:

- This a test of a possible future dynamic “VOBox” or “Edge Service” capability within FermiGrid.

Prerequisites:

- We will be “recycling” the hardware that is currently supporting the non-HA Grid services,
- So these deployments will need to wait until the transition to the FermiGrid-HA services has completed.

FermiGrid-HA - Ancillary Services

FermiGrid also runs/hosts several ancillary services which are not critical for the Fermilab Grid operations:

- Squid,
- MyProxy,
- Syslog-Ng,
- Ganglia,
- Metrics and Service Monitoring,
- OSG Security Test & Evaluation (ST&E) tool.

As the FermiGrid-HA evolution continues, we will evaluate if it makes sense to "HA" these services.

FermiGrid-HA - Conclusions

Virtualisation benefits:

+ Significant performance increase,
+ Significant reliability increase,
+ Automatic service failover,
+ Cost savings,
+ Can be scaled as the load and the reliability needs increase.

Virtualisation drawbacks:

- Significantly more complex design,
- Many “moving parts”,
- Many more opportunities for things to fail,
- Many more items that need to be monitored.

Any Questions?