

High Availability Hands On Session

<http://fermigrid.fnal.gov>

Steven C. Timm

FermiGrid Services Group, Fermilab

with significant contributions from

Dan Yocum and Keith Chadwick

OSG Site Admins@IUPUI

August 6, 2009

Introduction

- FermiGrid has run Highly Available VOMS, GUMS, MySQL, SAZ, and Squid servers for 1 year now, with uptime stretches of 6 months with 99.9952% uptime.
- GUMS and SAZ rates are regularly 18k/hr.
- Available HA Technologies
- Overview of how we made GUMS and VOMS highly Available
- What services are next to make highly available
- Preview of tutorial

Planning a High Availability System

- Highly available facility, UPS preferable
- Redundant power, two different power panels.
- Heartbeat—ways for a system to check that the other system is alive.
- How to get the ip traffic forwarded to the right system or systems
- How and where is the state of the application stored
- How to keep the multiple masters in sync
- How to maximize throughput and minimize power

Example: GUMS

- All state information stored in MySQL database
- Web service on a single port, easily redirected if necessary
- High availability and high throughput essential
- Develop a shared-nothing system
- (no NFS, no SAN)
- GUMS can run in limited memory, but is one of several tomcat apps that wants the port 8443—good candidate for Xen.
- As a site administrator you should be running GUMS even if you are a small site. We are making the grid-mapfile instructions harder to find every time.

IP HA technologies

- FermiGrid considered three options
 - Round robin DNS
 - Linux Virtual Server
 - Switch-based load balancing.
- Selected Linux Virtual Server
 - Piranha product from Red Hat
 - Use Direct Routing (LVS-DR) method
 - Trying to extend this to IP tunneling.
- Listens on voms.fnal.gov, gums.fnal.gov, saz.fnal.gov, fg-mysql.fnal.gov, squid.fnal.gov, voms.opensciencegrid.org, osg-ress-1.fnal.gov, osg-ress-4.fnal.gov
- Re-directs connections based on IP+port to backend “real servers”
 - Uses weighted least connections (WLC) to schedule connection requests
 - Other algorithms available

HeartBeat

- In LVS-DR, real servers respond directly to client
 - LVS “pings” services for availability
 - Removes service from scheduling if unavailable
 - Adds service back in when available
 - LVS master server is fg5x0
 - LVS backup server is fg6x0
 - Active-passive configuration
 - Failover in 6 seconds “pulse” daemon
 - Active connections to backend servers are maintained
 - If a service or real server fails during an open connection, the connection is lost.

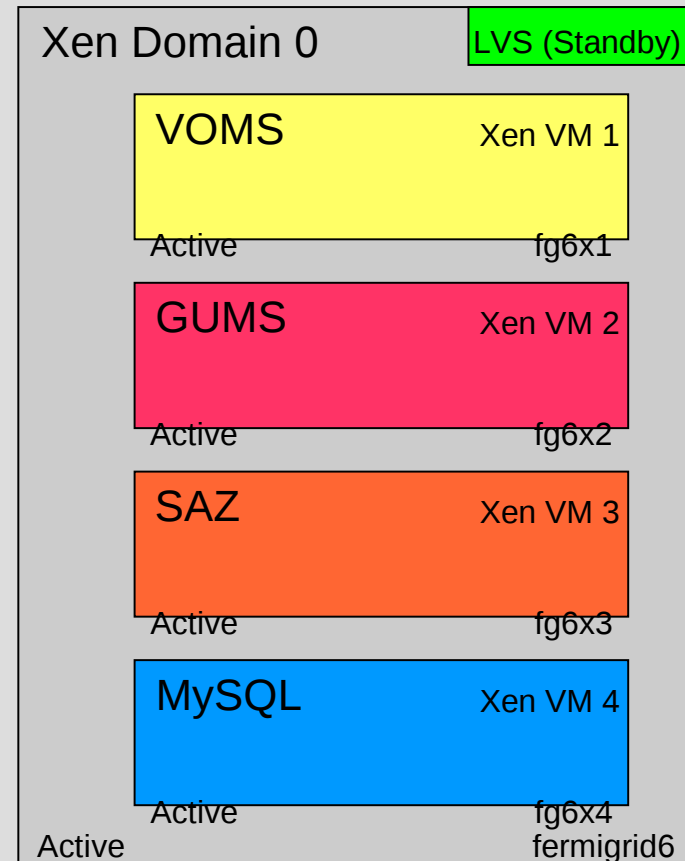
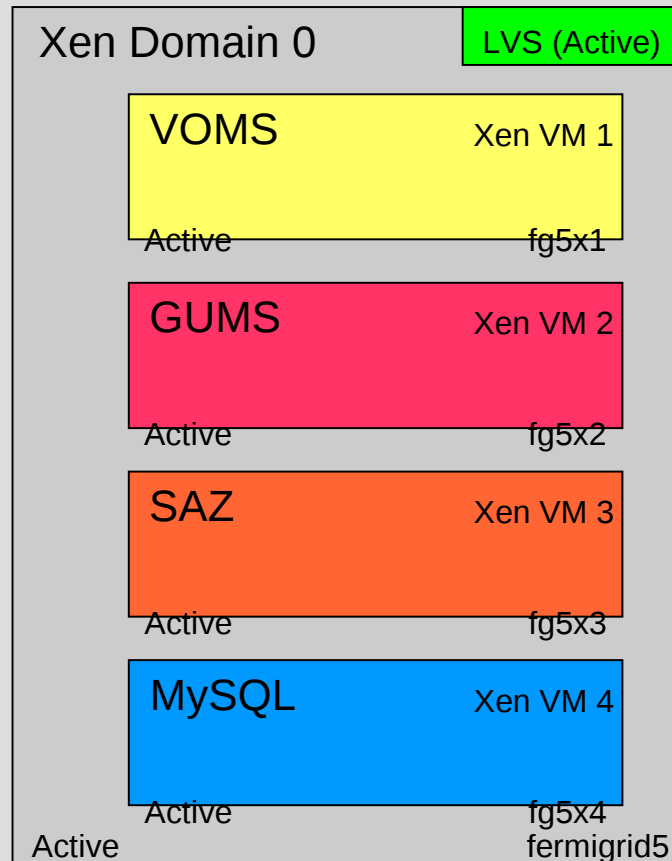
MySQL

- Investigated several technologies
 - MySQL Cluster
 - drbd – distributed remote block device
 - mysqlhotcopy
 - mysqldump + rsync
 - Multi-master replication (requires MySQL >v5.0.2)
- Decided on Multi-master replication
 - Scales well to 10 systems
 - currently only using 2
 - Near real-time replication, 1.1ms for 1KB record
 - Database server outages handled correctly
 - Logs replayed after server rejoins chain

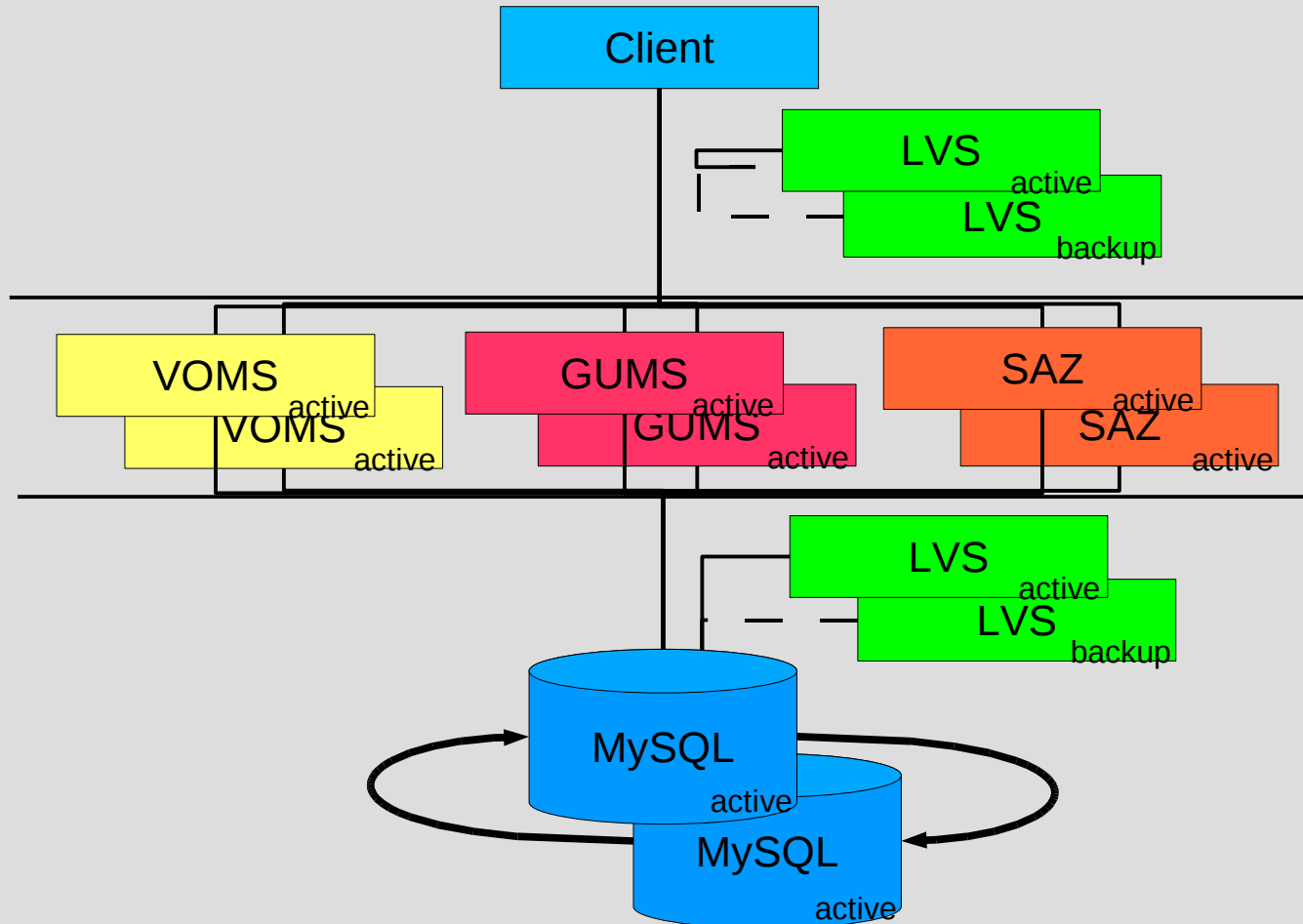
MySQL Replication

- Complete failure requires full copy (obviously)
 - >2 systems require special techniques to close the circular chain in the event of failure
- Service Replication
 - 2 VOMS “servers”, one each on fg5x1 & fg6x1
 - Each machine serves 12 VOs
 - 2 GUMS servers, one each on fg5x2 & fg6x2
 - 2 SAZ servers, one each on fg5x3 & fg6x3
 - All services use replicated MySQL servers on fg5x4 and fg6x4
 - No source code modifications

Organization of VM's



FermiGrid-HA



Note 1: All network connections are on the public network

Note 2: LVS directors displayed separately for convenience – they are the same in reality

Near-trivial HA

- Web-services listener for the BDII. No persistent state
 - GOC recently deployed this.
- ReSS Information Gatherer—No persistent state
 - Osg-ress-1 and osg-ress-4 were deployed in June.
- Ganglia—just set up two and have gmond send to both
- Condor Collector/Negotiator
 - Condor has its own HAD daemons that can transparently fail over the negotiator between several collector-running machines. All our production clusters running these.

Now comes the hard part

- MyProxy—state is stored in flat files on the file system
- condor_schedd—has native HA availability but requires shared FS between two machines.
- Pre-ws Globus gatekeeper / gsiftp
 - Needs state information in \$GLOBUS_LOCATION/tmp
- We have converted condor_schedd and globus gatekeeper to run on “service ip’s” so the service can move from machine to machine.
- NFS-lite also needs all the home directories of the users.
- NFS server replicated by DRBD is OK for myproxy, not for globus stuff.
- FermiGrid has HA “Bluearc” NFS server but don’t want to put this data across public net if possible.
 - Proxies moving across NFS in the clear have been sore point from beginning of OSG with no clear resolution.
- DRBD doesn’t always play nice with Xen kernels. Some kernel hackery may be required.

Hard Part, continued

- Gratia databases (>100GB) not amenable to multi-master MySQL replication
 - But Gratia has its own record-based replication strategy and we will use it to duplicate the collectors and mysqldb
 - Goal is to have one collector/mysqldb for queries and reports, the other one dedicated for inserts.
- WS-GRAM—Crazy file locking precludes NFS, but all state is in files somewhere.
- Spread out the HA across the Fermilab site in case of a building outage.

Xen and HA

- Many OSG services are now well-enough behaved to run in 2GB of RAM or less. (VOMS, GUMS). They are not big cpu loads either, especially not VOMS.
- But they all want the same ports and you want to start them independently
- Current servers have 8 cores, 16GB RAM
- We divide each into 4 or 5 Xen domains

Xen and HA continued

- Stock Xen 3.1.2 comes with RedHat/Centos/SL 5 update 2.
- Xen guest reboots in 5-10 seconds
- Base Dell PowerEdge server takes 3-5 minutes.
- We started with four services machines (one for each of voms, gums, saz, mysql) and condensed them into 2 physical servers, with 2 instances of each.
- This summer we used the HA features to move the whole system from old to new hardware without scheduling a downtime.
- In production we have used exclusively “Paravirtualized” Xen hosts.
- Test Bestman-xrootd gateway deployed with Fedora 10 and KVM —smooth operation but some file and network slowness.

Poor man's HA

- Split the cluster in half, each half with a head node so you will never lose more than $\frac{1}{2}$ cluster at once with a head node failure.
- Have 2-4 gatekeepers going into a single cluster so that cluster is still accessible. Also a must for throughput to big condor clusters.
- Install your single head node using Xen, it is much easier to move it to another piece of hardware later.

The HandsOn part

- Go to Google Docs
- (request an invite from StevenCTimm@gmail.com if necessary).
- Or grab the exported PDF file that is on the Indico page and/or the Twiki.
- <http://docs.google.com/Doc?docid=0AVY11NMVz7mPZGdmNjg0bWZfMmZiNWttbmQ0&hl=en>
- Major parts:
 - LVS Server
 - Pulse/heartbeat configuration
 - Multi-master mysql
 - Redundant GUMS.