

## Experience with Multi-Tier Grid MySQL Database Service Resiliency at BNL

**Tomasz Wlodek, Michael Ernst, John Hover, Dimitrios Katramatos, Jay Packard, Yuri Smirnov, Dantong Yu**

Brookhaven National Laboratory, Upton, NY, USA

tomw@bnl.gov

**Abstract.** We describe the use of F5's BIG-IP smart switch technology (3600 Series and Local Traffic Manager v9.0) to provide load balancing and automatic fail-over to multiple Grid services (GUMS, VOMS) and their associated back-end MySQL databases. This resiliency is introduced in front of the external application servers and also for the back-end database systems, which is what makes it "multi-tier".

The combination of solutions chosen to ensure high availability of the services, in particular the database replication and fail-over mechanism, are discussed in detail. The paper explains the design and configuration of the overall system, including virtual servers, machine pools, and health monitors (which govern routing), as well as the master-slave database scheme and fail-over policies and procedures. Pre-deployment planning and stress testing will be outlined. Integration of the systems with our Nagios-based facility monitoring and alerting is also described. And application characteristics of GUMS and VOMS which enable effective clustering will be explained.

We then summarize our practical experiences and real-world scenarios resulting from operating a major US Grid center, and assess the applicability of our approach to other Grid services in the future.

### 1. Introduction: GUMS and VOMS

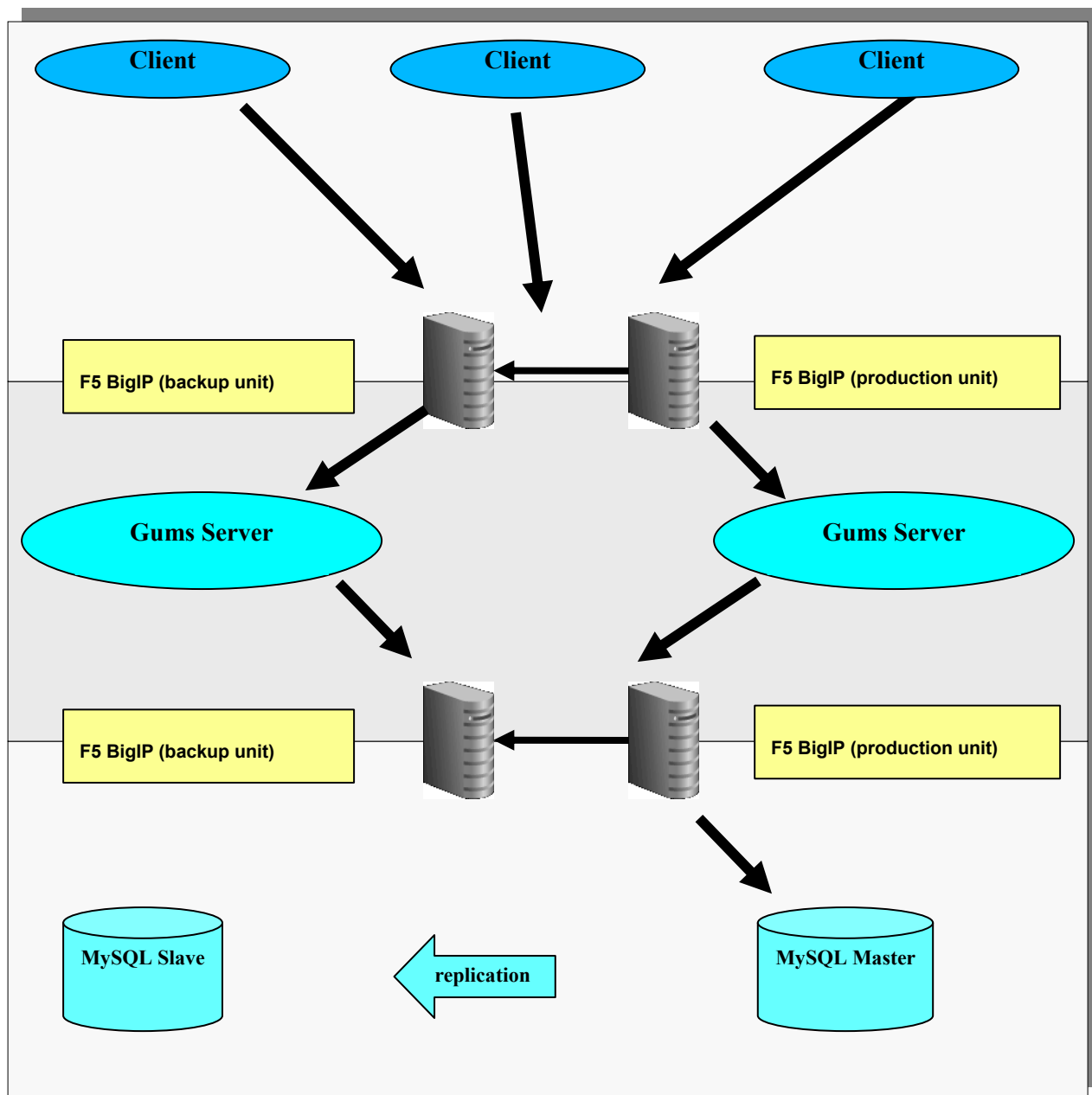
Brookhaven National Laboratory hosts the Tier-1 computing center serving the ATLAS experiment in the US.

The Grid User Management System (GUMS) [1] server provides mapping between external user identities and local UNIX accounts.

VOMS (Virtual Organization Management Service) [2] is a service used to generate short-lived grid credentials in order for users to authenticate to grid sites.

BNL runs a replica of the main ATLAS VOMS to serve as a backup in case the main service goes down, or network connectivity to Europe is broken.

As an online service with frequent requests (1-2 Hz), GUMS is run on two physical hosts (dual core Xeon, 2GB RAM). The VOMS on the other hand is less frequently called (on the average 1 request per ~5 minutes), therefore it is run on two virtual hosts.



Schematic view of the F5/GUMS/MySQL system

In order to assure high reliability the system was built around Big IP F5 switches [3] and replicated databases to assure that backup systems are ready to automatically replace faulty components at any time with no noticeable downtime from users' perspective. BigIP Local Traffic Manager (LTM) by F5 Networks is a network switch which allows resource allocation, load balancing, application switching, stateful fail over and traffic monitoring. Two Big IP F5 units are used: one serves as the production machine while the second one as a hot replacement backup which can be automatically put in place of the main unit should it became unavailable.

## **2. GUMS Servers**

Two independent GUMS servers have been installed on two different hosts. Clients do not connect to them directly. Instead they connect to an address assigned to the BigIP switch which then transparently routes the traffic to one of the two servers using a simple Round Robin algorithm.

The GUMS servers in turn retrieve user information and data from back end-databases. Again, the databases are not contacted directly but via F5 switch instead.

## **3. GUMS MySQL Servers**

Two GUMS database servers are installed on 2 Penguin Relion 1650SS machines, 8 cores Intel Xeon E5405 2.00GHz each, x86\_64 processor, 16 GB RAM and RHEL4 OS. MySQL [3] server version 5.1.29 is used. They form a master-slave system, any information written to or updated to master machine is replicated on the slave.

In normal circumstances the F5 switch is configured to connect only to the master, which in turn, is constantly monitored for availability. If the F5 detects a problem with the master, it will route all connections to the slave machine only. The system can operate in backup mode indefinitely, until operator decides that the condition, which caused the error, has been removed. The system will revert to using the master server only by explicit order from the human operator.

When the system runs in backup mode, the master and slave databases become unsynchronized: since GUMS server uses the slave database for write operations, the slave contains more recent information. Before resuming operation of the master database the operator needs to perform a dump of slave server and load its content to the master server to ensure that they contain identical information.

## **4. Monitoring of the GUMS-F5-MySQL system**

The status of the GUMS-F5-MySQL system is monitored by the Nagios [4] program by means of custom-written Nagios plugins.

The database replication is monitored by executing frequent comparisons of databases on both master and slave machines. Should a discrepancy be found Nagios sounds an alarm. In addition, the time it takes to propagate a database update from master to slave is regularly measured, and if it is found to be too long an alarm is raised. Long term experience tells us that in practice the slave is rarely more than a hundredth of a second behind the master database.

The F5 server is monitored by capturing SNMP traps sent by the switch to the Nagios server. The traps are decoded and those which provide useful information are then fed into Nagios using passive monitoring mode.

## **5. Stress testing**

Before using the system in production a number of tests were conducted to determine how it would behave in unusual circumstances.

First, we have measured the time delay in synchronization of both MySQL instances while the system was subjected to a large number of write requests, exceeding the expected load factor by at least an order of magnitude. The delay was found never to exceed 1.0 second, even for periods of very high load.

Second, we have simulated the connections at the rate corresponding to that expected in production and monitored the delay of database replication for long periods of time. The delay was found never to exceed 0.1 second during a few months long test run.

Then, we have simulated network interruptions by simulating load and unplugging the network connection between the MySQL servers to stop the replication. No data loss was observed and the replication automatically restarted after the network connection was restored.

The next step involved testing the behavior of the system when the disk on the MySQL server is full. We found out that the probe used by the F5 to determine the health of MySQL server can properly identify this situation and the F5 did switch to backup machine.

Finally we have powered off various machines in the system to simulate a hardware loss and checked the F5 reaction to verify that it does switch to backup mode. Those tests were successful too. No data loss was observed, regardless of which GUMS server or MySQL server was turned off.

## **6. GUMS Database Backup Policies**

In order to protect against unforeseen incidents a double pronged database backup policy was developed. First of all – the master database is constantly backed up by the virtue of being replicated to the slave system. In addition, once per day a full master database dump is performed. The database dump files are then exported to an external machine from which they are periodically written to tape. Together with the database contents, database configuration files are exported and archived.

As an additional precaution against losing the configuration files, they are inserted, once per day, to a dedicated table in the master server database itself. This provides us with the ability to track all configuration changes as they occur on a day to day basis. Together with rest of the master database contents the configuration information table is replicated onto the slave server, which protects it from accidental loss.

## **7. Experience with running the GUMS-F5-MySQL System**

The system has been in use since August 2009. During that time four separate incidents happened which could have had the potential to disrupt GUMS services. In each of those interruptions, the F5 switch reacted as expected and redirected the database traffic to the slave server.

Three of those incidents were related to network interruptions. Once the network problems subsided the system recovered easily and the databases could be easily re-synchronized and returned to normal operating mode.

The fourth incident was a result of faulty RAID card in the machine hosting master database server. The card failure resulted in total database loss. Fortunately, the F5 could immediately redirect the connections to the slave server and the system continued to operate without interruption. Later, when a replacement machine arrived from the manufacturer, the master server was rebuilt, the database was populated using a slave server dump, the slave and master were resynchronized, and the system returned to its original configuration. This incident alone demonstrated the necessity of using clustered systems for high availability solutions. Without an F5 front-end and master-slave database system, the loss of GUMS database would have led to a prolonged downtime for BNL services.

In each of those incidents no measurable interruption of GUMS services to the user community occurred.

## **8. Conclusions**

The GUMS-F5-MySQL system has proved to be highly reliable and fault resistant. During first year of operation it did not experience a single unscheduled downtime. In particular it survived without any interruption of service even a serious RAID card failure, which in a non-clustered system would lead to facility wide downtime and loss of CPU time.

Despite its good track record, we have decided to switch the GUMS backend databases from MySQL to Oracle, which happens to perform its own clustering. The decision is a result of a desire to concentrate all our clustered database efforts on one platform, namely Oracle. This abandonment of

MySQL does not imply that this database is not suitable for high availability servers, but is a result of non-technical considerations.

It should be stressed that even if we finally abandon the MySQL database as the GUMS backend in favor of Oracle we will still be committed to using the F5 switch for load balancing of GUMS servers. We have found this device to be highly useful not only for the GUMS system but for a wide array of other applications as well.

## References

- [1] <https://www.racf.bnl.gov/Facility/GUMS/1.3/>. Also see: “A scalable grid user management system for large virtual organization” , G. Carcassi, T. Carter, Z. Liu, G. Smith, J. Smith, J. Spiletic, T. Wlodek, D. Yu, X. Zhao, . Published in proceedings Interlaken 2004, Computing in high energy physics and nuclear physics. Full text available from <http://cdsweb.cern.ch/record/865727/files/p876.pdf>
- [2] <http://edg-wp2.web.cern.ch/edg-wp2/security/voms/voms.html>, <http://vdt.cs.wisc.edu/VOMS-documentation.html>
- [3] <http://www.f5.com/products/big-ip/>
- [4] <http://www.mysql.com/>
- [5] <http://nagios.org>