

# VALIDATION OF A MYSQL-BASED ARCHIVING SYSTEM FOR ALBA SYNCHROTRON

S.Rubio-Manrique (CELLS-ALBA Synchrotron, Spain) G.Strangolino (ELETTRA, Italy) M.Ounsy, S.Pierre-Joseph Zephir (SOLEIL, France)

## Abstract

ALBA Synchrotron collaborates with SOLEIL and ELETTRA institutes in the improvement of the Archiving System for Tango. An open source Database engine (MySQL) has been chosen and the viability and limitations of a MySQL-based Archiving System have been evaluated in a test platform. Using Java-based data collectors both centralized and distributed architectures have been tested. It allowed to demonstrate the maturity of the system, being achieved the most critical requirements.

## INTRODUCTION

ALBA is a new Synchrotron Light Facility in Barcelona, Spain. The commissioning of Alba booster accelerator will start before the end of 2009. For commissioning and operation Alba needs an Archiving System keeping records of a large set of critical variables; in order to prevent failures, diagnose problems and optimize the Machine efficiency.

Most of actual accelerator archiving systems are based on proprietary databases (mostly Oracle), being the Jefferson Lab Accelerator Facility (Newport News, US) the unique Light Source that reported [1] a working Archiving System on an Open Source database, MySQL. Their success has been a good reference for our work.

Open Source database engines present several advantages respect to its commercial equivalents. Economic cost of licenses is the most obvious but it must be considered also the higher flexibility in the election of the hardware platform an operating system.

This study tests the MySQL capabilities by evaluating Alba archiving requirements on a MySQL-based Tango Archiving System. This work have been done in tight collaboration with the rest of members of TANGO community: Soleil, ESRF, Elettra and Desy institutes.

## ALBA Archiving System Requirements

These are the requirements for ALBA Archiving System, based on a survey on other Light Sources experience (ESRF, Soleil) [3]:

- Number of attributes to be archived: 6,000 for 2010 commissioning; 20,000 in 2012.
- Historic archiving: 10 seconds between values, all variables stored permanently.
- Temporary archiving: 1 second between values, 5 days round buffer.
- Online backup and export between databases must be available.

## TANGO ARCHIVING SYSTEM

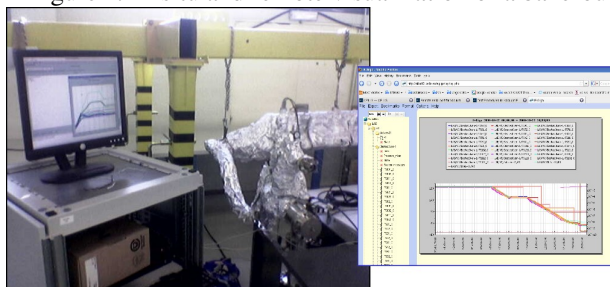
The Tango Archiving System was originally developed at Soleil Synchrotron Institute (Paris, France), under the terms of the TANGO collaboration [2] with the European Synchrotron Radiation Facility (ESRF, Grenoble, France) to develop a new distributed object-oriented Control System. Alba (Barcelona, Spain) and Elettra (Trieste, Italy) institutes joined later this collaboration and compromised to continue the development and implementation of the Archiving System.

## Existing systems

The Tango Archiving System is actually integrated in the Soleil's Machine Control System [3], but using a proprietary Oracle Database architecture. An alternative MySQL version of the Archiving System exists, but it has been used only in Beamlines and/or for testing purposes[4] or relatively small control systems (Alba [5], Elettra booster).

In the mark of the Tango collaboration Alba took the responsibility of evaluating the maturity of the Archiving System and refine its full deployment. Diagnostic, Vacuum, Electronics and Optics laboratories at Alba have been using the Tango Archiving System since April 2007.

Figure 1. In-situ and remote visualization of a bake-out.



## Architecture

Amongst others, Tango Archiving is performed by two types of control processes (Tango Device Servers):

- Manager: singleton dedicated to manage the configuration of the variables to store.
- Archivers: processes dedicated to data collection, validation and database insertion.

Other software processes and utilities are used for data verification, extraction and visualization verification for both supported databases (MySQL and Oracle). Most of existing tools are Java-based, although new developments appeared later in python, C++ and php.

The Archivers have three types of implementations depending of the mode of archiving. The modes evaluated in this report are Historical (HDB) and Temporary (TDB). A third mode (Snapshot) exists for save and restore operations and keeping archiving configurations.

Each type of archiver has been customized to solve specific performance issues and each of the modes has its own database schema, although they share both configuration and visualization tools.

### Historical Archiving

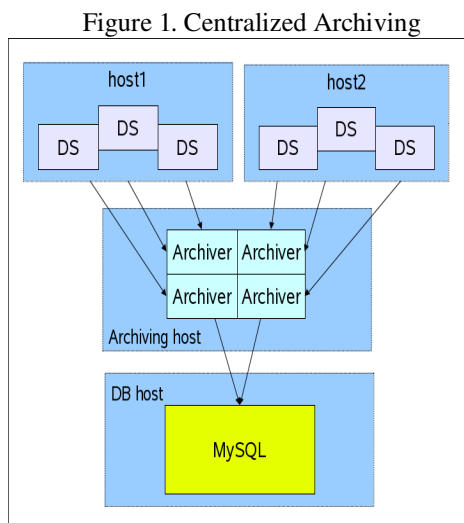
The Historical Archiver (HDB) is focused on permanent storage of variables read at low speeds (>10 seconds between readings) and raises issues related to permanent storage: backup, restoring, maintenance, time of access, data pruning, disk space.

### Temporary Archiving

The Temporary Archiver (TDB) stores variables within 1 second periods, using a round-buffer with a customizable size between 3 and 7 days. TDB Archiver becomes the most critical as its performance requirements are much higher. To reduce the amount of connections to MySQL the values are stored in a cache and inserted into the database in bunches of 600 values (10 minutes).

### Deployment

Standard deployment of the Archiving System centralizes all archiving processes in a single archiving server. Each archiver process is on charge of reading periodically or on demand an specific set of attribute values. This values will be stored in the database depending of filters previously specified.



Critical parameters for performance tuning are the total number of Archiver processes, the number of attributes assigned to each Archiver process and the number of collector threads used.

### Storage Engine and Backup

Online Backup and Restoring tests are actually available only using MyIsam storage engine, which is not fully transactional and presented some inconsistencies. Tests with InnoDB and table partitions are going to be done in the next months using the databases obtained from this report.

One of the main drawbacks of the actual MySQL implementation is the lack of millisecond resolution on timestamps (available in Oracle or PostgreSQL). A solution to this problem has not been implemented yet, but some proposals consider adding an extra column to existing database tables.

## PERFORMANCE BENCHMARKS

A benchmark in 4 phases has been scheduled to test the performance of Tango Archiving System and compare it with Oracle's performance as reported at Soleil [4]:

- First, a Tango Archiving System have been installed.
- Second, Tango device servers (PySignalSimulator) have been prepared to generate the data for the tests.
- Third, MySQL performance have been evaluated on with Temporary archiving.
- Fourth, an Historical archiving is evaluated.

### Generation of the Data

The size of the benchmark has been constrained to 6000 PySignalSimulator devices with 1 attribute each. This number has been considered enough to compare MySQL performance with existing Soleil's Oracle-based archiving reports [3], recording 5000 attributes. MySQL *maxconnections* and *maxuserconnections* parameters have been modified to a value of 20,000 to perform the tests.

Table 1, Benchmark Platform Specs, that allowed to generate, collect and archive the data in a single computer.

Processor	2 Dual XEON, 2.33 GHz
RAM	16GB, 667 MHz
Hard Disk	6146 GB; 10,000 rpm
OS	OpenSuSE Linux, 10.2, 64 bits
MySQL (/var)	335 GB
Cache files (/tmp)	200 GB

### System startup

The process of startup of devices and Archivers is an slow and complex process that stressed too much the control system. Deadlocks may occur if Archivers are not started sequentially and the archiving load must be well distributed; in addition many problems appeared if all the information was not available before starting the Archiving System. Once Startup phase is finished the system becomes stable and reliable.

## RESULTS

Watcher utilities provided by Soleil have been used to verify that all Archivers were working and the amount of values registered for each attribute was at least 95% of generated. For each archiving mode the CPU and disk usage have been evaluated.

### *TdbArchiver test*

The Temporary Archiving system has been tested storing 4000 tango attribute values per second. To store all this information 20 Archiver processes have been used, with 5 collector threads each. A 7 days round buffer required a 43GB database and 100GB of cache files.

### *HdbArchiver*

The Historical Archiving system has been tested storing 6000 tango attributes in a 10 seconds period. 20 Archiver processes have been used with 5 collector threads each. The disk space used by MySQL tables is incremented in 750Mb/Day.

### *Latency and Data losses*

A rate of 3600 mesures/hour was configured for all attributes, but an average number of 3579 mesures/hour were found in tables. First cause of data losses were drifts observed in data acquisition threads, due to Java sleep() method unaccuracy.

After compensating this error still some data losses were observed (0.58% of the total in 1 week). Debugging the process of exporting 20 attributes (load of an Archiver device server, with 5 threads collecting 4 variables each) it has been observed that sometimes it is abnormally large and longer than the export operation period. At that point the device forces the next export to the database and the resting unexported data is lost, reasons for this abnormal export times have not been found yet.

### *Performance Limits*

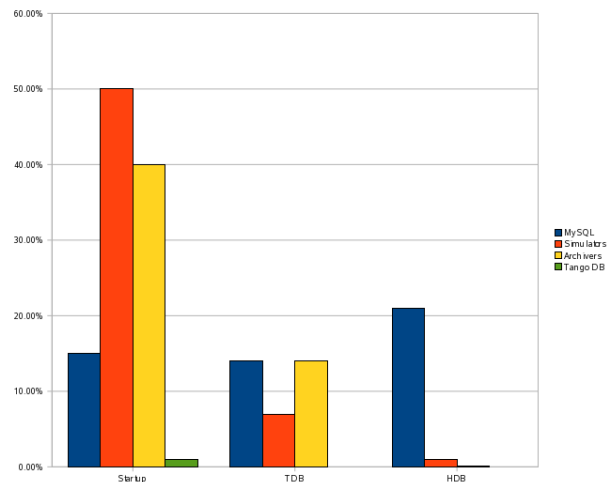
It has not been possible to start HdbArchiver and TdbArchiver tests at the same time as the drifts in all communications between devices provoked continuous timeouts and the whole system entered in a deadlock state. Event-based archiving have been tested mostly at Elettra but the high CPU usage of the notification service (notifd) didn't allowed to store big number of attributes.

The CPU load will be distributed through the system to solve all problems referred to CPU usage, separating both databases and moving data collection processes on each controls host. Deadlocks have been solved using a "dedicated" configuration for each host Archiver becoming *blind* to the rest of the system; but making the configuration much more complex.

With high loads (>3000 attributes) the Historical Archiving performed worse than TDB due to the big number of MySQL connections being continuously open

and closed. To solve this problem the same bulk upload system used by TDB must be implemented, but adding latencies to clients and data visualization.

Figure 4. CPU Usage Vs. Archiving Mode



## CONCLUSIONS

Once all the problems related to deployment tuning and system startup have been solved the MySQL database reached most of our requirements. Comparing its performance with Oracle [4] it's remarkable that Oracle needed two servers to perform the archiving of a 3 days round buffer of 5000 attributes/second using 60 GB disk space, while MySQL needed only 1 server and achieved the same number of attributes using 43 GB of disk space for a 7 days round-buffer, reducing 2.73 times the disk usage.

The lack of timestamp resolution and the difficulties for online backup are still a disadvantage of MySQL, but its good performance proved that Open Source databases are mature for big and intensive applications; being PostgreSQL a good alternative to solve this issues. This test have also demonstrated the maturity of Tango's Archiving System, but pointed out the problems that must be solved in the next releases to increase its reliability.

## REFERENCES

- [1] M. H. Bickley "A MySQL-Based Data Archiver: Preliminary Results", 2007, JLAB-TN-07-063, Jefferson Lab, Newport, US
- [2] A.Götz, E.Taurel, J.L.Pons, P.Verdier, J.M.Chaize, J.Meyer, F.Poncet, G.Heunen, E.Götz, A.Buteau, N.Leclercq, M.Ounsi, "TANGO a CORBA based Control System", Proceedings of ICALEPCS 2003, Gyeongju, Korea
- [3] S. P. Joseph, M. Ounsi "Status of Soleil's Archiving System", Tango Collaboration Meeting 2007, Paris, France
- [4] E. Taurel, "Testing the Tango Archiving System", 2004, ESRF, Grenoble, France
- [5] S. Rubio "Status of ALBA Archiving", 2007, CCD-CT-GD-0032, Bellaterra, Spain