Tango Archiving @ ALBA

Sergi Rubio Manrique, ALBA Controls Group

Tango Meeting, Synchrotron Soleil, 26th May 2010

Introduction

- ALBA has been using Soleil's Tango Archiving System since 2006, for all laboratories and installation projects.
- Our laboratories database contains data from 800 attributes and uses 5.6 Gb of data storage.
- The archiving system has been used successfully during our Linac and Booster commissionings since 2008.
- Commissioning and pre-commissionning operations archived 3751 (HDB) attributes using 16Gb of data storage.

Status

Archiving

- Data stored by domains:
  - Linac-LT: 555
  - Booster: 3150
  - Storage Ring: 46
- Data stored by subsystem:
  - Control: 191
  - RF: 370
  - Vacuum: 633
  - Magnets: 764
  - Diagnostics/Liberas: 1747

Snap

- 375 attributes in 23 contexts
- All dedicated to Magnets Power Supplies

Requirements

LI+LT+BO commissioning (July 2010):

- LI/CT: 106
- RF: 370
- VC: 670
- MA: 600 (ramping power supplies)
- DI(libera): 1222

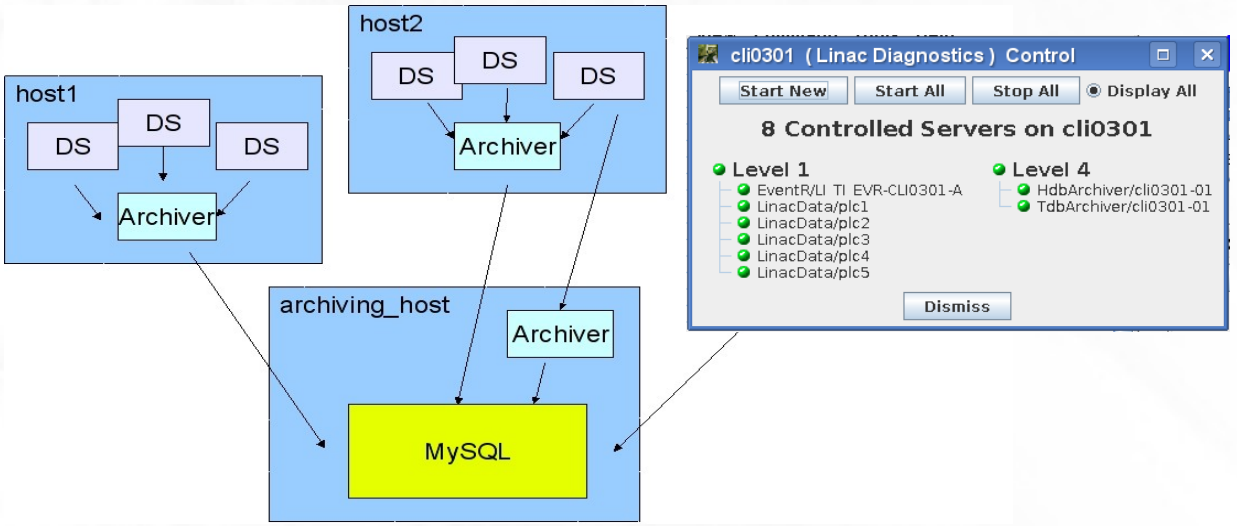
Total for BO commissioning: 2968

Storage Ring Commissioning (November 2010)

- LI: 106
- RF: 2000
- VC: 1975
- MA: 1080
- DI: 3792

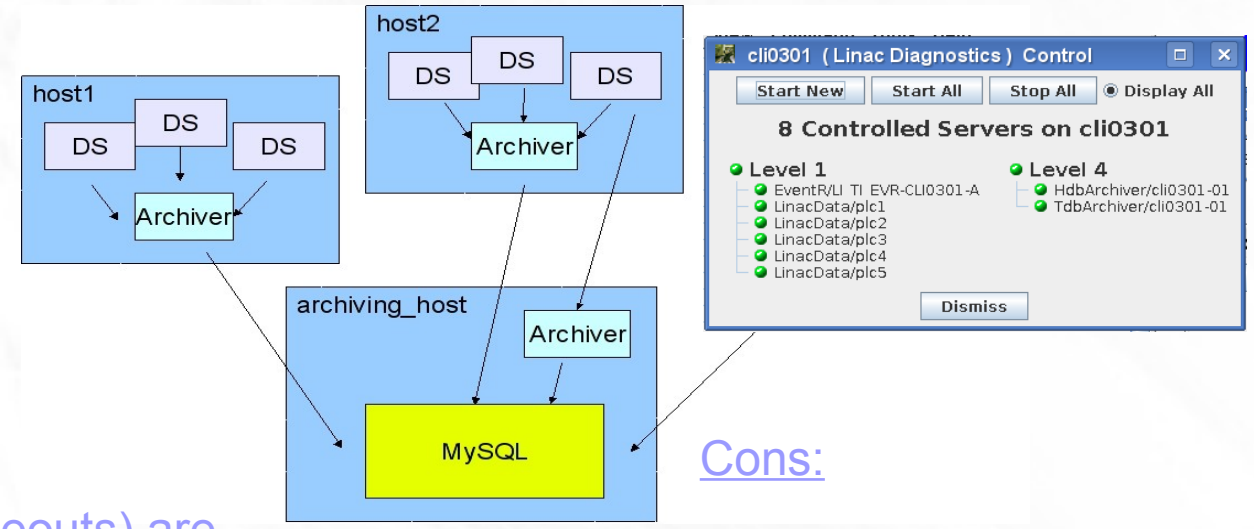
TOTAL: 8953 HDB attributes

Distributed Architecture



- Most of HdbArchiver processes are running in the same CPU's where data is generated (each archiver device manages 50 attributes).
- Attributes managed by archivers are specified using the “reservedAttributes” property of each archiver.
- The PyTangoArchiving API takes care of registration of attributes for each HdbArchiver, distribution of the load and configuration of Tango Starter processes (5 devices/server).

Distributed Architecture



Pros:

- Problems (timeouts) are isolated from the rest of the system.
- The central server is just dedicated to MySQL management.
- Runlevels ensure that attributes are available for the archiver.

Cons:

- Configuration split in two databases, archiving and Tango.
- We cannot use Mambo to manage this topology.
- Configuring the archiving became much more complicated.

Archiving Configuration

- Configuration of the archiving modes for the machine has been done using .CSV forms.
- One reason for it was that we wanted to have all the control over the archiving system in the first stages.
- It was also a factor that Mambo didn't allowed to distribute the load as we wanted.
- The LoadArchivingConfiguration script uses the information of the Tango Database to create dedicated HdbArchiver instances wherever we need them.
- Created CSV files are used to configure archiving, but also to perform periodical checks by subsystem.

Archiving Configuration

	A	B	C	D	E	F	G	H
1	Host	Device	Attribute	Type	ArchivingMode	Periode >15	MinRange	MaxRange
2								
3	#The # is used for	@ for special command	periodic is always stored	absolute needs a m	absolute works with %			
4	#Special Comman	@COPY used to copy/paste the configuration of a device to other						
5								
6	#This header lines are mandatory!!!							
7	@LABEL	PC_LT						
8	@AUTHOR	David Yepez						
9	@DATE	13/05/2010						
10	@DESCRIPTION	General Power Converters Archived data.						
11								
12	#dserver host	domain/family/member	attribute	HDB/TDB/STOP	periodic/absolute/relative			
13								
14	cpc0402	lt01/pc/bend-01	@DEFAULT	HDB	periodic	86400		
15			CurrentSetpoint	HDB	absolute	15	0	0
16			Current	HDB	absolute	15	0	0
17			Voltage	HDB	absolute	60	0	0
18			MachineState	TDB	absolute	15	1	1
19			ErrorCode	TDB	absolute	15	1	1
20								
21	cpc0402	lt01/pc/bend-02	@COPY:lt01/pc/bend-01					
22	cpc0402	lt01/pc/q-01	@COPY:lt01/pc/bend-01					
23	cpc0402	lt01/pc/q-02	@COPY:lt01/pc/bend-01					
24	cpc0402	lt01/pc/q-03	@COPY:lt01/pc/bend-01					
25	cpc0402	lt01/pc/q-04	@COPY:lt01/pc/bend-01					
26	cpc0402	lt01/pc/q-05	@COPY:lt01/pc/bend-01					
27	cpc0402	lt01/pc/q-06	@COPY:lt01/pc/bend-01					
28	cpc0402	lt01/pc/q-07	@COPY:lt01/pc/bend-01					
29	cpc0402	lt01/pc/q-08	@COPY:lt01/pc/bend-01					
30	cpc0402	lt01/pc/q-09	@COPY:lt01/pc/bend-01					

Archiving Server

We have two identical servers dedicated to archiving, one for our machine database and the other for tests and benchmarking.

Processor	2 Dual XEON, 2.33 GHz
RAM	16GB, 667 MHz
Hard Disk	6146 GB; 10,000 rpm
OS	OpenSuSE Linux, 11.1, 64 bits
MySQL (/var)	335 GB
Cache files (/tmp)	200 GB

In both we are running MySQL 5.0 and OpenJDK 1.6.0 on a SuSE 11.1 distribution. Although we are probably going to migrate to SUN's java.

We use InnoDB engine for configuration tables and MyISAM for data (Soleil's defaults), but we are studying other storage engines to solve problems related to backups.

MySQL Tests Results

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 propietary 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

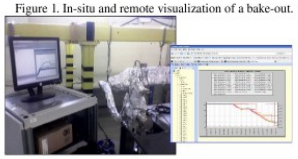
The Tango Archiving System was originally developed at Soleil Synchrotron Institute (Paris, France), under the terms of the TANGO collaboration [2] with the European Synchrotron Radiation Facility (ESRF, Grenoble, France) to develop a new distributed object-oriented Control System. Alba (Barcelona, Spain) 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 propietary 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):

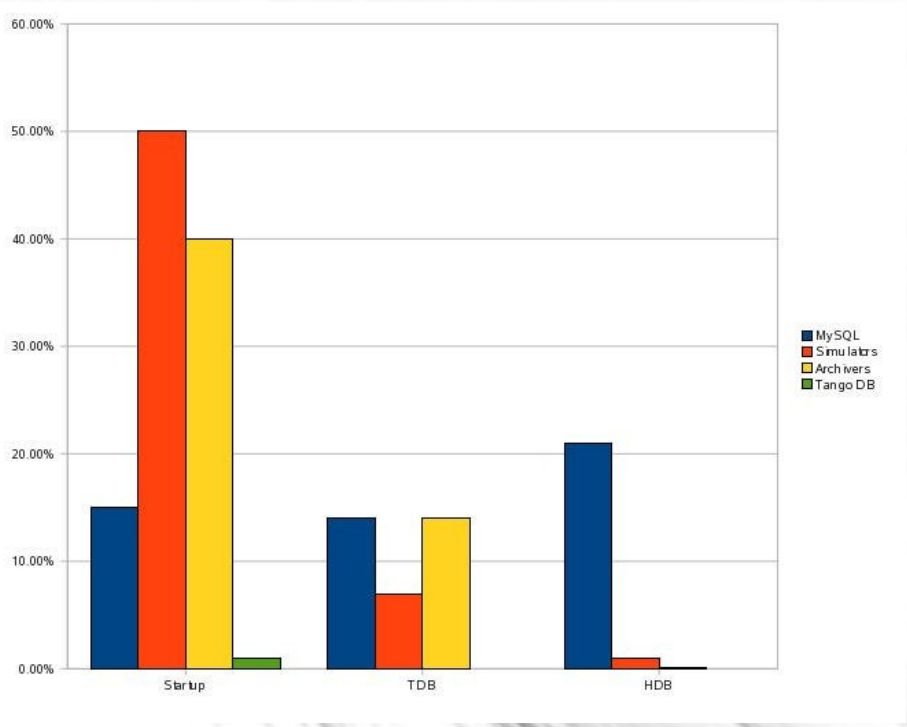
- Manager: singletone dedicated to manage the configuration of the variables to store.
- Archivers: processes dedicated to data collection, validation and database insertion.

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.

- TDB:
  - 4000 attributes per second.
  - 20 Archiver servers, 5 devices each.
  - MySQL uses 43Gb for 7 days round buffer.
  - Temp files needed 100Gb of disk space.
- HDB:
  - 6000 attributes at 10 seconds each.
  - 20 Archiver servers, 5 devices each.
  - MySQL grows at 750Mb/day.
- 99% of generated data was successfully archived (due to Java sleep tolerances).
- But: the lack of millisecond resolution in timestamps and the problems with life backup/restore are still a disadvantage of MySQL.
- Disk usage seems 2-3 times smaller than Oracle's (it can be related to lower timestamp precision).
- The paper is available in Tango's site (Icalepcs2009 Papers folder).

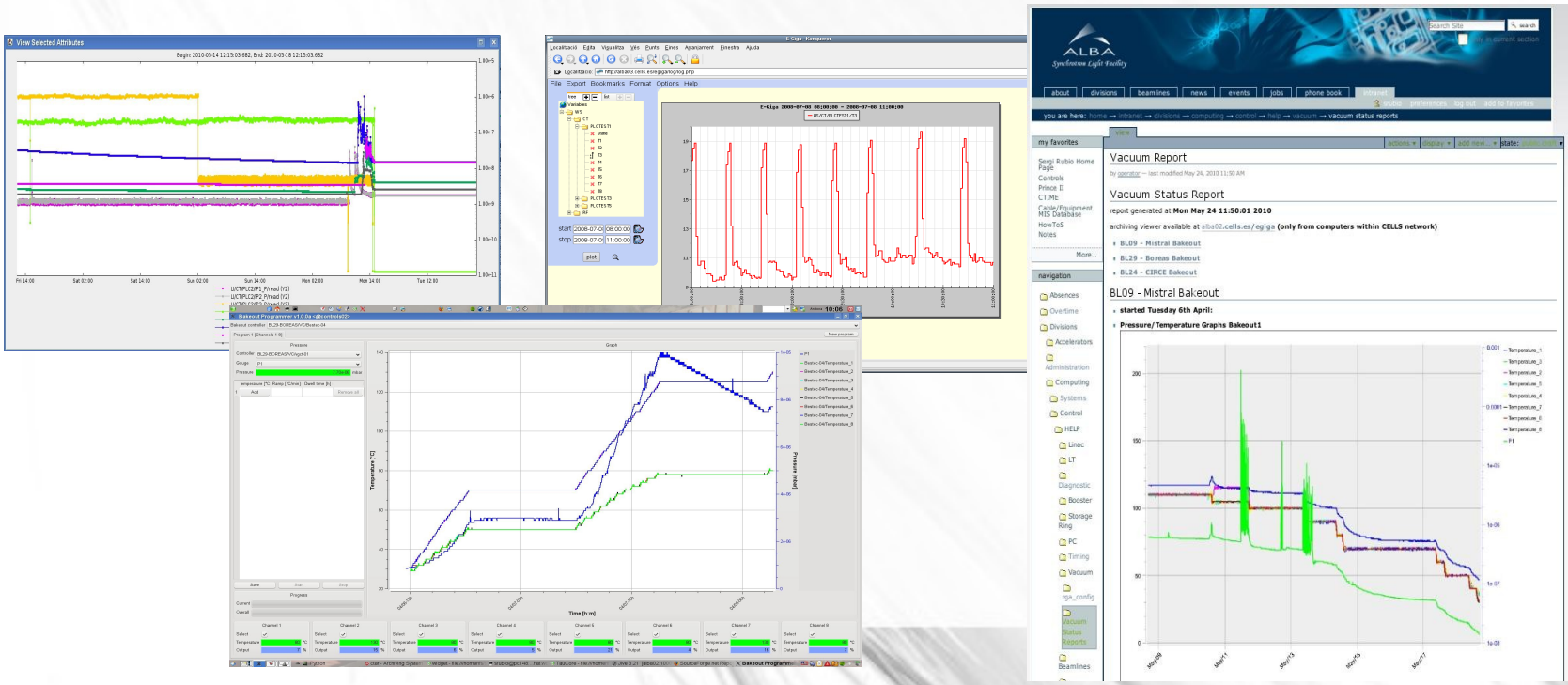
CPU Usage

- CPU Usage matters as it affect errors in Java sleeps and data losses.
- With high loads (>3000 attributes) the Historical archiving performed worse than Temporary due to the amount of separated MySQL insertions.
- To solve this issue a bulk upload system like the used by TDB could be implemented (using a cache and inserting once every 10 minutes).



User Applications

- Our approach for both archiving and snapshot is to integrate them in the existing applications rather than having a separate archiving GUI.
- Tau Trend widgets allowed to visualize archived data in Tau user applications and other python-based tools, like automatic html reports.
- For data that cannot be visualized by trends/labels we still use Mambo Viewer.
- E-Giga is still the application most used in laboratories since 4 years go.



Pending things

- Distributed TDB is still to be fully deployed and tested, it is more difficult to manage than HDB due to temporary files usage in diskless systems.
- Mambo has some problems of case when navigating the Tango database (e.g. LI and li domains are shown separately).
- Archiving of certain types (Strings, States, Images) has still to be tested for all archiving modes .
- We had certain problems in the last migrations of databases that forced us to investigate different storage engines and/or partitioning of tables to make backup/restoring processes more reliable.
- Event-based archiving has been discarded until problems with notifd will be solved (huge memory and cpu usage).
- Snap saving/restoring doesn't perform well with big contexts (>100 attributes).

Conclusions

- The development done by Soleil provide us of a powerful tool that allowed us to have a good archiving system since the beginning of the project.
- The eternal pending issue is still coordination and communication of changes between releases, as they are still mostly unknown.
- We also need an easy mechanism to solve differences in the approach (e.g. case sensitiveness, events) and make easier to merge our developments.
- But again, we really have to thanks Soleil for the effort and the great job they did!

Merci pour votre attention