HDB++

L. Pivetta

lorenzo.pivetta@elettra.trieste.it

Elettra - Sincrotrone Trieste, Trieste, Italy

Introduction

- Novel Tango device server for Historical Data Base archiving
- Written in C++
- Fully event-driven
- Architecture based on:
  - One or more Event Subscriber DS
  - One Configuration Manager
  - One or more Data Extraction DS
  - Libraries for data insertion and extraction API
- HDB++ Tango DS Design Guidelines distributed to EC members last week (ELETTRA + ESRF joint effort)

Requirements

- Fast
- Efficient
- Reliable
- Event driven: each Tango DS knows when to archive something
- Flexible: easy to manage and maintain even **without** graphical frontends
- Self contained: single source for all configuration parameters (Tango DB)
- Modular: dedicated decoupling libraries to support different database engines, data insertion, data extraction
- Scalable: comes from Tango architecture... for free

Why C++?

A number of considerations lead to choose C++ language; amongst them:

- Efficiency
- Better support and maintenance for C++: Tango core development language as well as ZeroMQ
- Small API for DB access foreseen
- Well known approach at ELETTRA: Alarm DS, PLCs DS
- Existing ESRF development to archive from Tango CS to legacy historical DB
- New **additional** implementation for HDB archiving

Performance figures: assumptions

- Average number of insertion per second: 100 to 1K
- Peak number of insertion per second: 1K to 10K
- Number of different attributes to store: 10K to 100K
- Max size of data per attribute*frequency: 8 byte*1K = 8KB (but...)
- Do we allow to store 25 images per seconds? ...err ...no?!? (but...)

Performance numbers: setup

- Machine 1: 1 * Intel(R) Core(TM)2Duo E7500, 2.93GHz, 4GB RAM
- Machine 2: 2 * Intel(R) Xeon(R) CPU E5462, 2.80GHz, 16GB RAM
- Ubuntu 10.04 64 bit
- MySQL 5.6.11, MyISAM tables

```
mysql> describe att_00050;
+-------------+-------------+------+-----+----------------------------+-------+
| Field       | Type        | Null | Key | Default                    | Extra |
+-------------+-------------+------+-----+----------------------------+-------+
| time        | datetime(6) | NO   |     | 0000-00-00 00:00:00.000000 |       |
| read_value  | double      | YES  |     | NULL                       |       |
| write_value | double      | YES  |     | NULL                       |       |
+-------------+-------------+------+-----+----------------------------+-------+
```

```
mysql> describe att_00054;
+-------+-------------+------+-----+----------------------------+-------+
| Field | Type        | Null | Key | Default                    | Extra |
+-------+-------------+------+-----+----------------------------+-------+
| time  | datetime(6) | NO   |     | 0000-00-00 00:00:00.000000 |       |
| value | tinyint(4)  | YES  |     | NULL                       |       |
+-------+-------------+------+-----+----------------------------+-------+
```

- *double* and *tinyint* clients written in C

Client and DB on same host

host	client	instances	inserts/s	MySQL CPU load	note
machine 1	double	1	24000	99%	
machine 1	double	2	37700	155%	
machine 1	double	3	38000	155%	
machine 1	tinyint	1	23500	99%	

host	client	instances	inserts/s	MySQL CPU load	note
machine 2	double	1	20000	99%	
machine 2	double	2	38000	200%	
machine 2	double	3	58000	300%	
machine 2	double	4	77000	398%	
machine 2	double	5	88000	457%	
machine 2	double + tinyint	5 + 1	101000	516%	
machine 2	double + tinyint	5 + 2	113000	580%	
machine 2	double + tinyint	5 + 3	127000	640%	+ sys = full load
machine 2	double + tinyint	6 + 3	127000	640%	

Courtesy R.Passuello

Client and DB on different hosts

Client host Machine 1, server host Machine 2, 1 Gb/s eth connection

client host	server host	client	instances	inserts/s	MySQL CPU load	note
machine 1	machine 2	double + tinyint	4 + 3	49000	250%	
machine 1	machine 2	double + tinyint	5 + 3	55200	280%	
machine 1	machine 2	double + tinyint	2 * (5 + 3)	74000	450%	
machine 1	machine 2	double + tinyint	3 * (5 + 3)	89000	650%	
machine 1	machine 2	double + tinyint	4 * (5 + 3)	98000	730%	

Courtesy R.Passuello

Some real numbers: ESRF

```
Mon Apr 15 20:00:05 2013 - Duration: 23h59m57s - 4659 signals - 1669633 writes
Tue Apr 16 20:00:05 2013 - Duration: 23h59m57s - 4651 signals - 1564622 writes
Wed Apr 17 20:00:06 2013 - Duration: 23h59m58s - 4689 signals - 1709353 writes
Thu Apr 18 20:00:05 2013 - Duration: 23h59m57s - 4516 signals - 1544975 writes
Fri Apr 19 20:00:05 2013 - Duration: 23h59m57s - 4507 signals - 1625554 writes
Sat Apr 20 20:00:06 2013 - Duration: 23h59m58s - 4520 signals - 1682674 writes
Sun Apr 21 20:00:06 2013 - Duration: 23h59m57s - 4633 signals - 1796858 writes
Mon Apr 22 20:00:06 2013 - Duration: 23h59m57s - 4547 signals - 1701167 writes
Tue Apr 23 20:00:05 2013 - Duration: 23h59m56s - 4737 signals - 2256014 writes
Wed Apr 24 20:00:06 2013 - Duration: 23h59m58s - 4731 signals - 1835094 write
Thu Apr 25 20:00:06 2013 - Duration: 23h59m56s - 4540 signals - 1750970 writes
Fri Apr 26 20:00:06 2013 - Duration: 23h59m58s - 4522 signals - 1746862 writes
Sat Apr 27 20:00:06 2013 - Duration: 23h59m57s - 4531 signals - 1566629 writes
Sun Apr 28 20:00:06 2013 - Duration: 23h59m57s - 4510 signals - 1570094 writes
Mon Apr 29 20:00:05 2013 - Duration: 23h59m56s - 5045 signals - 1764713 writes
Tue Apr 30 20:00:06 2013 - Duration: 23h59m58s - 5060 signals - 1520570 writes
Wed May 01 20:00:05 2013 - Duration: 23h59m56s - 4965 signals - 1423531 writes
Thu May 02 20:00:05 2013 - Duration: 23h59m57s - 4847 signals - 1434059 writes
...
```

Courtesy P.Verdier

2.256.014 / 24 / 60 / 60 = 26 write/s mean value (no info on peak value, but...)

Migration issues

- Same DB tables: backward compatible for clients. Extended tables?
- Non event-based archiving: additional polling subsystem (Tango DS) may act as gateway polling the devices and generating events. **Not** part of the HDB++
- Legacy DB data migration (ESRF specific): to be addressed