HDB++ Status

L.Pivetta

on behalf of hdb++ team

R.Bourtembourg
JL.Pons
C.Scafuri
G.Scalamera
G.Strangolino
P.Verdier
L.Zambon

Requirements

- Written in C++
- Event-driven
- Architecture based on:
  - One or more archiving engines (EventSubscriber TANGO ds)
  - Configuration management (ConfigurationManager TANGO ds)
  - Libraries for data insertion and extraction (C++ and Java)
  - Data extraction TANGO ds / clients
- Fast
  - One database for slow and fast archiving (up to 1 Khz, luckily even more)
- Flexible
  - Easy to manage and maintain even without GUI frontends
- Self contained
  - Single source for all configuration parameters (TANGO DB)
- Modular
  - Abstraction libraries to support different database engines and schema
    - Support for existing HDB schema on MySQL
    - Support for **hdb++ new schema** with improved features (µs timestamp)
    - Support for **noSQL** back-end (Apache Cassandra)
    - Easily extensible to additional database/schema
- **Scalable**: same as TANGO, deploy as many DS as you need

Original specification

HDB++ design guidelines

HDB++ Archiver TANGO device server (EventSubscriber)

- event based
- all the **configuration** stored in the TANGO device
- one thread in charge of event(s) subscription and callback execution: fills a FIFO acting as producer
- one thread in charge of pushing data into the database; reads the FIFO as consumer
- device methods allow to perform the following per-instance operations:
    - start the archiving for all attributes
    - stop the archiving for all attributes
    - start the archiving for one attribute
    - stop the archiving for one attribute
    - read the number of attributes in charge
    - read the list of attributes in charge
    - read the configuration parameters of each attribute
    - read the number of working attributes
    - read the list of working attributes
    - read the number of faulty attributes
    - read the list of faulty attributes with diagnostics
    - read the size of the FIFO queue
    - read the number of attributes pending in the FIFO
    - read the list of attributes pending in the FIFO
- the EventSubscriber exposes some additional figures:
    - for each instance, total number of records per time
    - for each instance, total number of failures per time
    - for each attribute, number of records per time
    - for each attribute, number of failures per time
    - for each attribute, time stamp of last record
- the following operating states are foreseen:
    - **ON**: archiving running, everything works
    - **ALARM**: one or more attributes faulty or the FIFO size grows above high-mark threshold
    - **FAULT**: all attributes faulty
    - **OFF**: archiving stopped

Original specification

HDB++ design guidelines

HDB++ Configuration TANGO device server (ConfigurationManager)

- the ConfigurationManager TANGO device server shall be able to perform:
    - manage the request of archiving a new attribute
        - create an entry in the database if not already present
        - setup the attribute's archive event configuration
        - assig the attribute to one of the Archivers
            - following some rules of load balancing (not yet available)
            - to the specified Archiver
        - move an attribute from one Archiver to another
        - keep trace of which attribute is assigned to which Archiver
        - start/stop the archiving
        - remove an attribute from archiving
- the Configuration manager will expose some global statistics:
    - total number of Archivers
    - total number of working attributes
    - total number of faulty attributes
    - total number of events/s

Original specification

Database interface

- C++ API to address reading and writing to the database

libhdb++ : HDB++ abstraction layer
libhdb++mysql : HDB++ schema support, MySQL back-end
libhdb++cassandra : HDB++ schema support, Cassandra back-end
libhdbmysql : legacy HDB schema support, MySQL back-end

HDB++

- First release running
  - At ELETTRA
    - on FERMI since fall 2013 with MySQL back-end
    - on ELETTRA since spring 2014 with MySQL back-end
  - At the ESRF
    - since July 2014 with MySQL back-end
    - Since October 2014 with Cassandra back-end

- Release update almost twice per year
  - Bugfix
  - New functionalities

TANGO
Connecting things **together**

HDB++ meeting 25 June 2014

Short term support

Item	Request	To do	Who
1	In case of error, store the error description	Add varchar column to the data_type tables and related code to EventSubscriber	Elettra
2	Store quality factor in addition to data value	Add column to the data_type tables and related code to EventSubscriber	Elettra
3	Make some attribute configuration parameters available in the historical database	Add a table to store *display-unit*, *format-string* and *label*. The table will also contain att_conf_id and the three timestamps. Add a fifo and producer/consumer threads to subscribe to attribute-configuration-change events into EventSubscriber	Elettra
4	Multiple TANGO host support	Whenever multiple TANGO host are specified, the configuration manager stores attributes with the TANGO_HOST env variable prefix; EventSubscribers subscribe events from both TANGO domains and stores the data using the TANGO_HOST prefix	Elettra
5	Support for write-only attribute	Most probably a bugfix	Elettra
6	Additional indexing on att_name varchar in att_conf table for faster search	Modify att_conf table	Elettra
7	Historical database data extraction library	Core library written in C in order to be used also with php and python; C++ wrapping. Define the list of interface methods and send to ESRF ASAP	Elettra
8	GUI for Configuration Manager	java based GUI tool	ESRF

Mid term support

Item	Request	To do	Who
9	Monitor on periodic event based archiving	Provide timeout on periodic event subscriptions; can be implemented as an alarm timer in the configuration change event thread	Elettra
10	Array support. Currently arrays are stored as fixed length strings into varchar; this may lead to loss of precision. Alternate approaches: • store into native data type using an additional index to reconstruct the array • store in binary form • use native array support	Test postgres native array support	Elettra, ESRF
11	Device locking mechanism to avoid concurrent setups	Collaborative client/server locking mechanism	ESRF, Elettra
12	java-based extraction library	Pure java implementation of the same API as the C/C++ extraction library; depends on the C API	ESRF
13	Addititonal database table to store data statistics for scalar values	New table with start_time, stop_time, min, max, average; additional statistics may be useful; write a tango device server, based on the extraction library, to calculate and store the statistics; depends on the extraction library	ESRF, Elettra
14	Extraction library data format support	Attribute history data format, json data format, both?!?	Elettra, ESRF
15	NoSQL databases	NoSQL databases evaluation	ESRF
16	Data stream management system (DSMS)	InfluxDB evaluation	Elettra

HDB++ meeting 19 Sep 2014

Short term support

#	Request	To do	Who	When
1.1	In case of error, store the error description	Add varchar column to the data_type tables and related code to EventSubscriber	Elettra	Available ✓
1.2	Store quality factor in addition to data value	Add column to the data_type tables and related code to EventSubscriber	Elettra	Nov 2014
1.3	Make some attribute configuration parameters available in the historical database	Add a table to store *display-unit*, *format-string* and *label*. The table will also contain att_conf_id and the three timestamps. Add a fifo and producer/consumer threads to subscribe to attribute-configuration-change events into EventSubscriber	Elettra	Dec 2014
1.4	Multiple TANGO host support: Tango 8	The correct FQDN has to be specified or just the domain/family/member can be specified letting the ConfigurationManager select the right TANGO_HOST/port	Elettra	Available ✓
1.5	Support for write-only attribute	Bugfix	Elettra	Avail ✓
1.6	Additional indexing on att_name varchar in att_conf table for faster search	Modify att_conf table	Elettra	Dec 2014
1.7	Historical database data extraction library	Core library written in C++; C wrapper to be used also with php and python. Define the list of interface methods and send to ESRF ASAP	Elettra	Prototype and docs available
1.8	GUI for Configuration Manager	java based GUI tool	ESRF	Avail ✓
1.9	Configure polling, event parameters and alarm threshold programmatically for AttributeNokNumber in EventSubscriber to enable alarms		Elettra	Dec 2014 *new*
1.10	Do not count stopped attributes as faulty		Elettra	Dec 2014 *new*
1.11	Provide per-attribute errors list	Is last error/exception description sufficient? Or do we need to store the error stack/history?	t.b.d.	*new*
1.12	Provide max value for AttributePendingNumber	Implement AttributeMaxPendingNumber attribute	Elettra	Dec 2014 *new*

#	Request	To do	Who	When
1.13	Provide number-of-events counter per attribute; reset by ResetStatistics; store the timestamp of the last call to ResetStatistics in an attribute (seconds since EPOCH)	Already available with the Attribute Status command; should this be: • a command with FQDN as input value • an attribute vector type containing all the counters AttributeEvenNumberList	Elettra	Available ✓

HDB++ meeting 19 Sep 2014

Mid term support

#	Request	To do	Who	When
2.1	Multiple TANGO host support: Tango 9	Use the DbGetCSDbServerList command to verify the FQDN. Then we still face the problem how to manage the aliases; there are two possibilities: • the ConfigurationManager can replace the alias with the real hostname; not the desired behaviour • the DatabaseDS has to be modified in order to return also the alias name as a valid FQDN; the events will come from the proper FQDN (i.e. alias)	Elettra, ESRF	To be clarified. Is it necessary?
2.2	Monitor on periodic event based archiving	Provide timeout on periodic event subscriptions; can be implemented as an alarm timer in the configuration change event thread	Elettra	March 2015 *new*
2.3	Array support. Currently arrays are stored as fixed length strings into varchar; this may lead to loss of precision. Alternate approaches: • store into native data type using an additional index to reconstruct the array • store in binary form • use native array support • do nothing and rely to timestamp	Test postgres native array support	Elettra, ESRF	Available with Cassandra. Evaluate Postgres and all... ✓
2.4	Device locking mechanism to avoid concurrent setups	Collaborative client/server locking mechanism	ESRF, Elettra	Available in CM GUI ✓
2.5	java-based extraction library	Pure java implementation of the same API as the C/C++ extraction library; depends on the C API	ESRF	Prototype available ✓

#	Request	To do	Who	When
2.6	Addititonal database table to store data statistics for scalar values	New table with start_time, stop_time, min, max, average; additional statistics may be useful; write a tango device server, based on the extraction library, to calculate and store the statistics; depends on the extraction library	ESRF, Elettra	March 2015
2.7	Extraction library data format support	Attribute history data format, json data format, both?!?	Elettra, ESRF	t.b.d.
2.8	NoSQL databases	NoSQL databases evaluation	ESRF	Pro... available ✓
2.9	Data stream management system (DSMS)	InfluxDB evaluation	Elettra	March 2015
2.10	Store encrypted passwords in the device/class property		Elettra	March 2015

HDB++ meeting 09 Dec 2014

Short term support

#	Request	To do	Who	When
1.1	In case of error, store the error description	Add varchar column to the data_type tables and related code to EventSubscriber	Elettra	Available
1.2	Store quality factor in addition to data value	Add column to the data_type tables and related code to EventSubscriber	Elettra	Available
1.3	Make some attribute configuration parameters available in the historical database	Add a table to store *display-unit*, *format-string* and *label*. The table will also contain att_conf_id and the three timestamps. Add a fifo and producer/consumer threads to subscribe to attribute-configuration-change events into EventSubscriber	Elettra	Available
1.4	Multiple TANGO host support: Tango 8	The correct FQDN has to be specified or just the domain/family/member can be specified letting the ConfigurationManager select the right TANGO_HOST/port	Elettra	Available
1.5	Support for write-only attribute	Bugfix	Elettra	Available
1.6	Additional indexing on att_name varchar in att_conf table for faster search	Modify att_conf table	Elettra	Available
1.7	Historical database data extraction library	Core library written in C++; C wrapper to be used also with php and python. Define the list of interface methods and send to ESRF ASAP	Elettra	Prototype and docs available
1.8	GUI for Configuration Manager	java based GUI tool	ESRF	Available
1.9	Configure polling, event parameters and alarm threshold programmatically for AttributeNokNumber in EventSubscriber to enable alarms		Elettra	Available
1.10	Do not count stopped attributes as faulty		Elettra	Available
1.11	Provide per-attribute errors list	Is last error/exception description sufficient? Or do we need to store the error stack/history?		Available (current error)
1.12	Provide max value for AttributePendingNumber	Implement AttributeMaxPendingNumber attribute	Elettra	Available

#	Request	To do	Who	When
1.13	Provide number-of-events counter per attribute; reset by ResetStatistics; store the timestamp of the last call to ResetStatistics in an attribute (seconds since EPOCH)	Already available with the Attribute Status command; should this be: • a command with FQDN as input value • an attribute vector type containing all the counters AttributeEvenNumberList	Elettra	Available

HDB++ meeting 09 Dec 2014

Mid term support

#	Request	To do	Who	When
2.1	Multiple TANGO host support: Tango 9	Use the DbGetCSDbServerList command to verify the FQDN. Then we still face the problem how to manage the aliases; there are two possibilities: • the ConfigurationManager can replace the alias with the real hostname; not the desired behaviour • the DatabaseDS has to be modified in order to return also the alias name as a valid FQDN; the events will come from the proper FQDN (i.e. alias)	Elettra, ESRF	To be clarified. Is it necessary?
2.2	Monitor on periodic event based archiving	Provide timeout on periodic event subscriptions; can be implemented as an alarm timer in the configuration change event thread	Elettra	Available ✓
2.3	Array support. Currently arrays are stored as fixed length strings into varchar; this may lead to loss of precision. Alternate approaches: • store into native data type using an additional index to reconstruct the array • store in binary form • use native array support • do nothing and rely to timestamp	Test postgres native array support	Elettra, ESRF	Available with Cassandra. Evaluate Postgres and all... ✓
2.4	Device locking mechanism to avoid concurrent setups	Collaborative client/server locking mechanism	ESRF, Elettra	Available in CM GUI ✓
2.5	java-based extraction library	Pure java implementation of the same API as the C/C++ extraction library; depends on the C API	ESRF	Prototype available ✓

#	Request	To do	Who	When
2.6	Addititonal database table to store data statistics for scalar values	New table with start_time, stop_time, min, max, average; additional statistics may be useful; write a tango device server, based on the extraction library, to calculate and store the statistics; depends on the extraction library	ESRF, Elettra	March 2015 ?
2.7	Extraction library data format support	Attribute history data format, json data format, both?!?	Elettra, ESRF	t.b.d.
2.8	NoSQL databases	NoSQL databases evaluation	ESRF	Pro... available ✓
2.9	Data stream management system (DSMS)	InfluxDB evaluation	Elettra	March 2015 ?
2.10	Store encrypted passwords in the device/class property		Elettra	March 2015 ?

HDB++ meeting 08 March 2015

#	Request	To do	Who	When
1.1	Attribute config subscription issue with Java servers	Work is in progress in order to fix the Java server. Keep existing logic for event error management, that is both events should work, and wait for bug fix for java. Move event error cout to ERRORLOG stream in EventSubscriber.		?
1.2	It would be useful to add a new attribute in the ES which gives the sum of events received by this es device since the last reset. It is not very convenient to wait for 2 hours before to get the first stats.	The counters for number of received events are already available.	Elettra	Available ✓
1.3	The frequency since last reset could be an interesting information too.	Available in AttributeRecordFreq and AttributeFailureFreq. In order to make it available for archiving the following configuration by code has been added for the above attributes: • change event - absolute threshold: 1 • archive event - period: 3600000 - absolute threshold: 1	ESRF	Available ✓
1.4	It would be useful to add a new attribute in the cm device which would give the total sum of events for all the archivers since the last reset. For this data to be relevant, it would be important that the ResetStatistics would be sent at the same.	There is no mechanism to guarantee the consistency of the statistics period and the EventSubscriber startup time for all the EventSubscribers. No global statistics support.		X
1.5	It would be convenient to poll the xxxNumber attributes by code (3 seconds should be fine).	Moreover the ES device code has been optimized not to acquire the internal lock when reading attributes by using an additional thread. The same thread pushes the relevant events.	Elettra	Available ✓

#	Request	To do	Who	When
1.6	Cassandra "schema" supports all the TANGO data types, one table per type (60 tables).	Support for all TANGO types on hdb++ tables has been discussed: using 1 table per tango type. Modify hdb++ schema (Elettra), libraries (Elettra) and extraction libraries (Elettra and ESRF)	Elettra, ESRF	T.b.d. ✓
1.7	When an attribute archival is stopped, the event subscriber is still connected to the attribute and is still receiving events, but is simply not inserting the data into HDB. It would probably be cleaner if we would unsubscribe to the events when we stop the archiving of this attribute. It is resource consuming to maintain the connection for nothing. For instance, during a shutdown period, we stop the archival of thousands of attributes. The es are still connected to these attributes for nothing, generating useless network traffic and consuming CPU cycles for nothing. The subscription/unsbscription strategy could be optimized.	Rename the actual Stop/Start commands in Pause/Resume and add new Stop/Start that also unsubscribe/subscribe to events.	Elettra	Available ✓
1.8	Run/shutdown issue or more generic approach	Could be useful to be able to support several operating "modes". Some dedicated tables can be added to the existing TANGO database schema. Wait for ESRF proposal.	ESRF	T.b.d.
1.9	Vector (array) support	Cassandra already supports vectors. Evaluate blob based approach for vectors in MySQL.	Elettra	T.b.d.
1.10	Documentation	Update the hdb++ documentation.	Elettra, ESRF	T.b.d.

HDB++ design guidelines

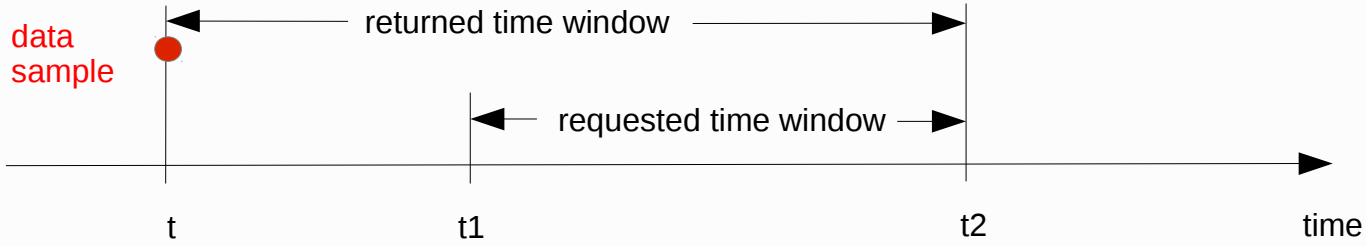
Data extraction

- C++ and Java native libraries

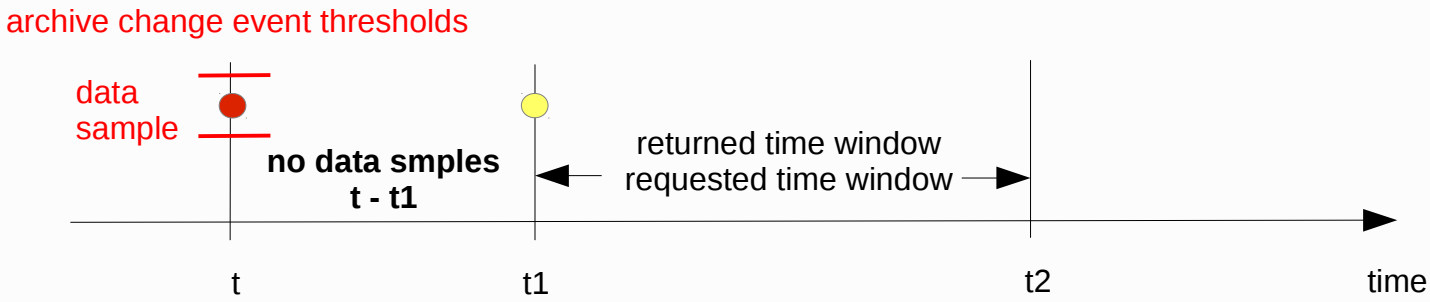
- The data extraction library shall be able to **deal with event based archiving;** the possible lack of data in the requested time window shall be properly managed:

  - returning some no-data-available error: in this case the reply contains no data

  - enlarging the time window to include some archived data; no fake samples have to be introduced



  - returning the value of the last archived data anyhow; the requested time interval is kept and the last available data sample returned; the validity of the data is guaranteed when **archive change event** is used, care must be taken in case of **archive periodic event**

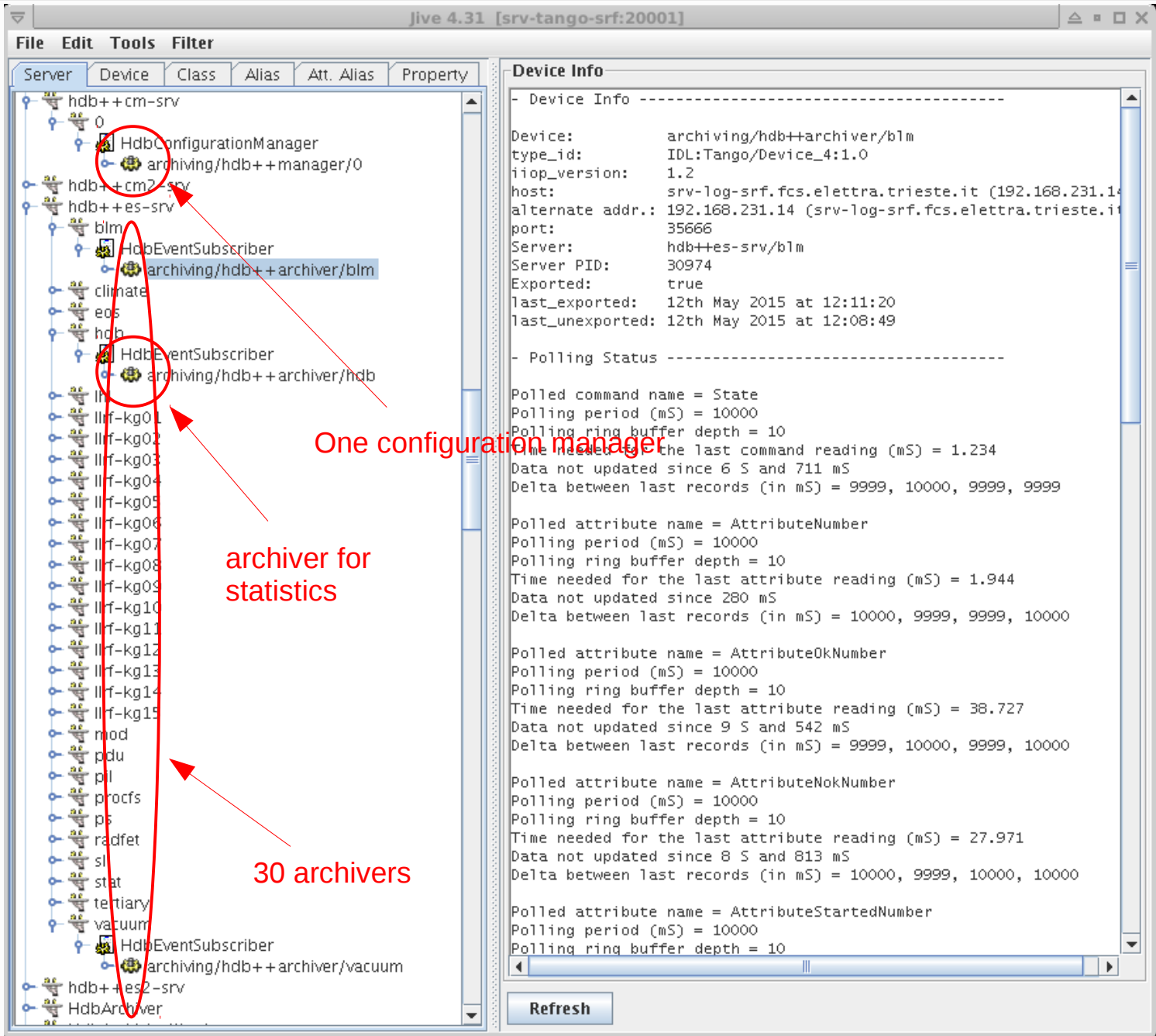


HDB++

FERMI setup

Legacy HDB schema

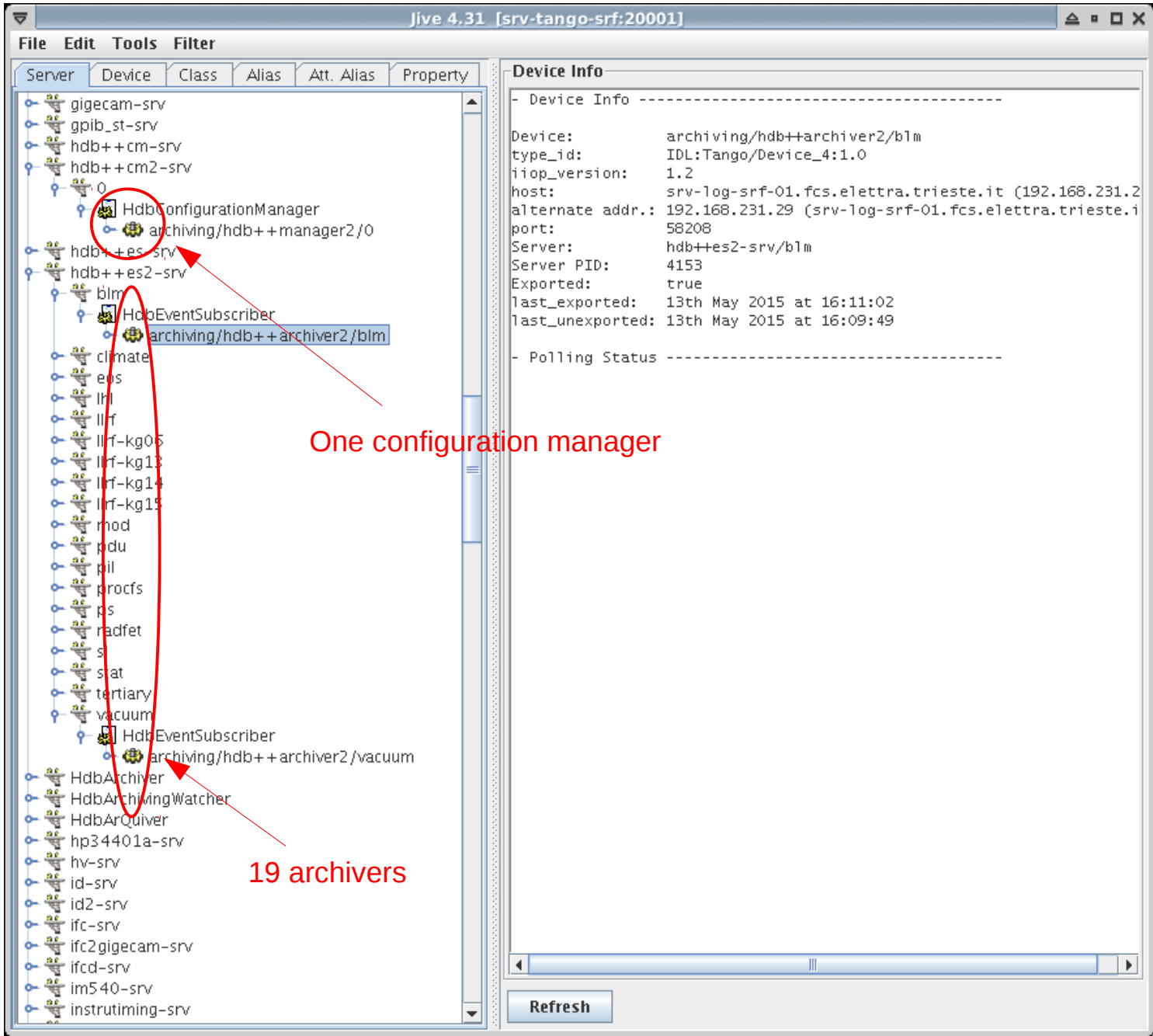
- 1 host
- 1 configuration manager
- **30** archivers
- functional partitioning
- 5605 attributes total
- from 1 to 1005 attributes per archiver



TANGO
Connecting things **together**

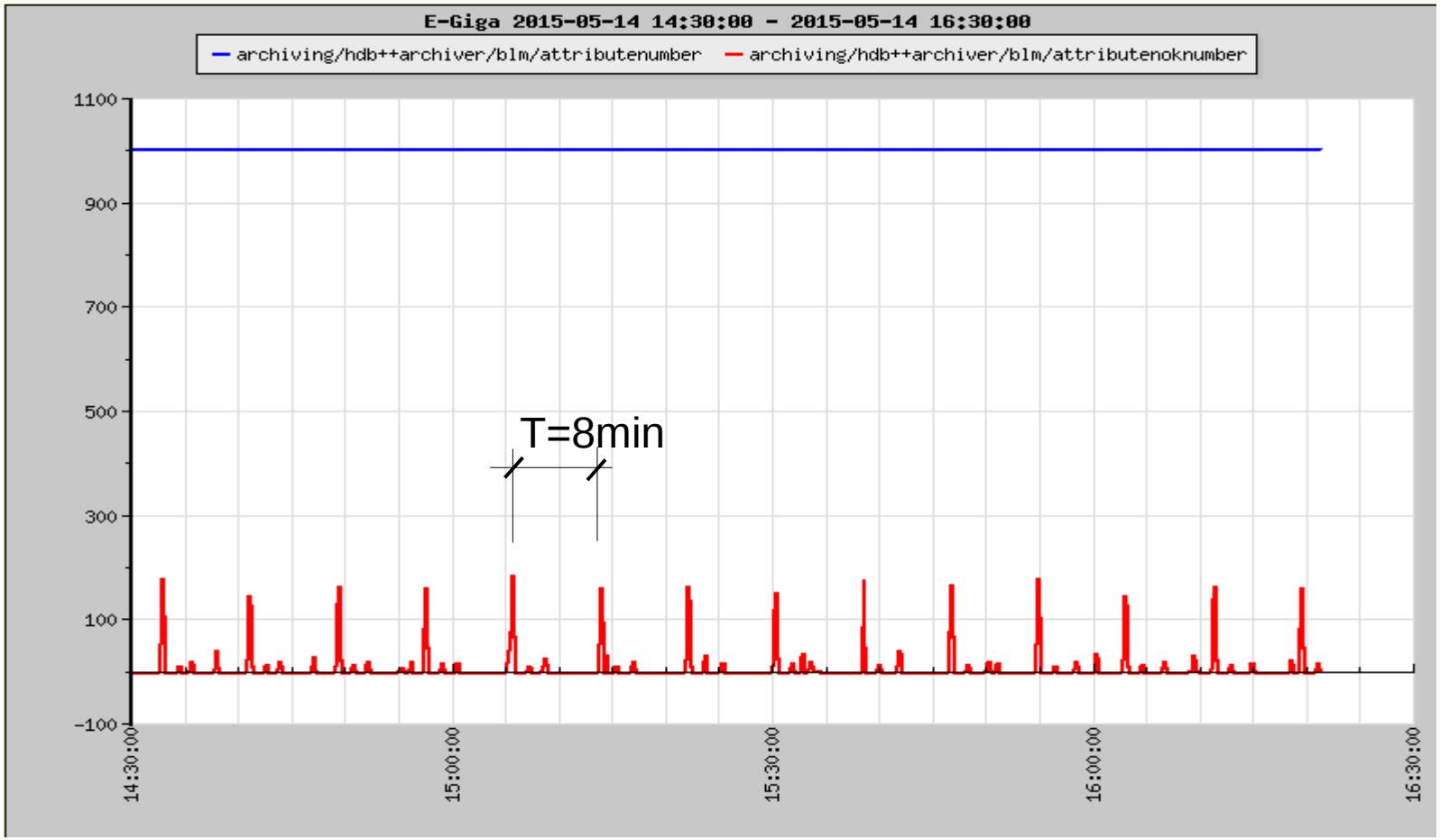
HDB++

FERMI setup
HDB++ schema
- 1 host
- 1 configuration manager
- **19** archivers
- functional partitioning
- 5605 attributes total
- from 1 to 1467 attributes per archiver



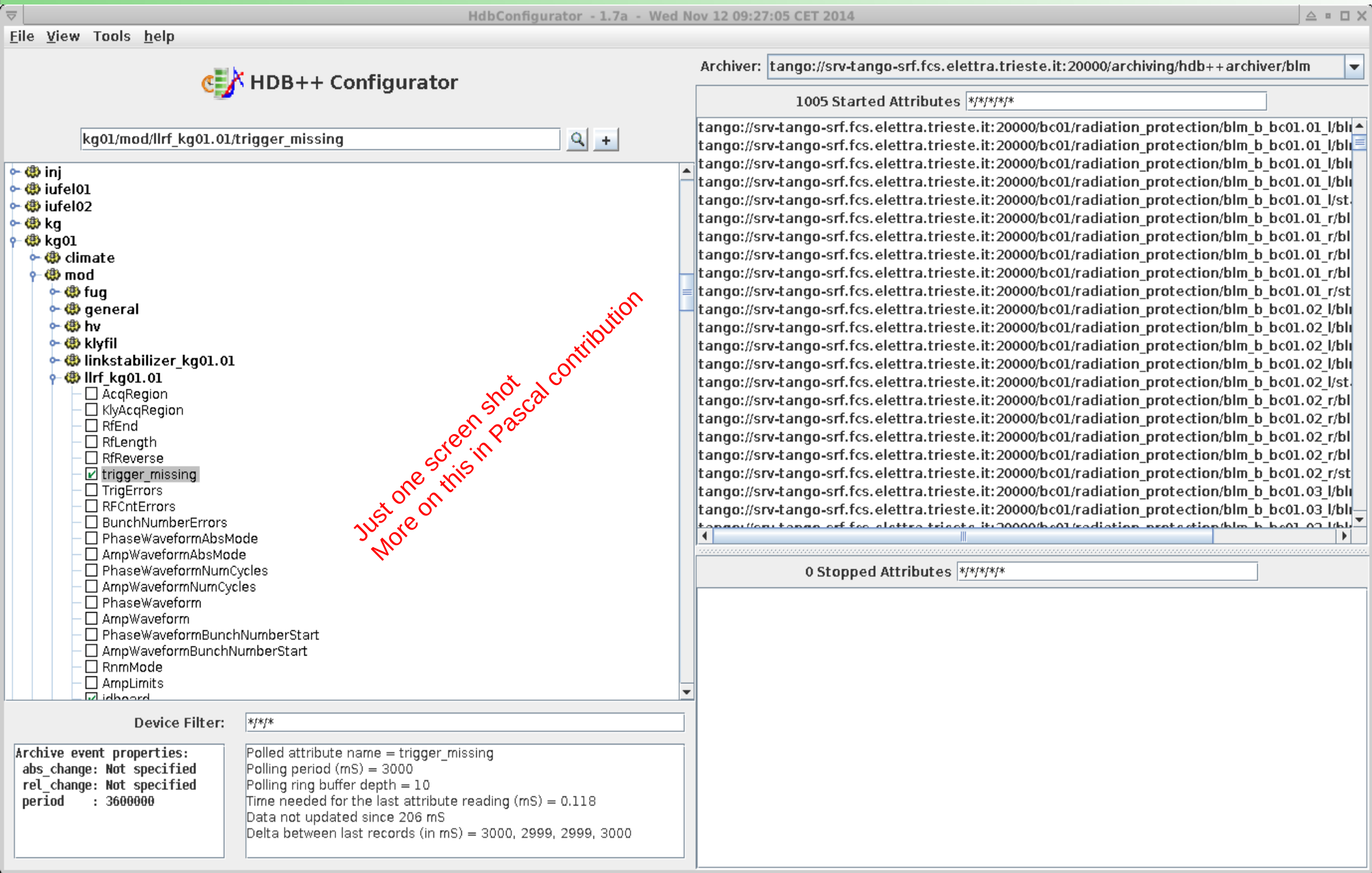
HDB++

Beam Loss Monitors
201 devices, 5 attributes archived on each device
4 instances, polling tuned: 10 devices per polling thread (50 attributes, cached, read time ~ 0.5 ms each)
Periodic archiving, T=60s
One archiver instance → 1005 attributes



TANGO
Connecting things **together**

HDB++

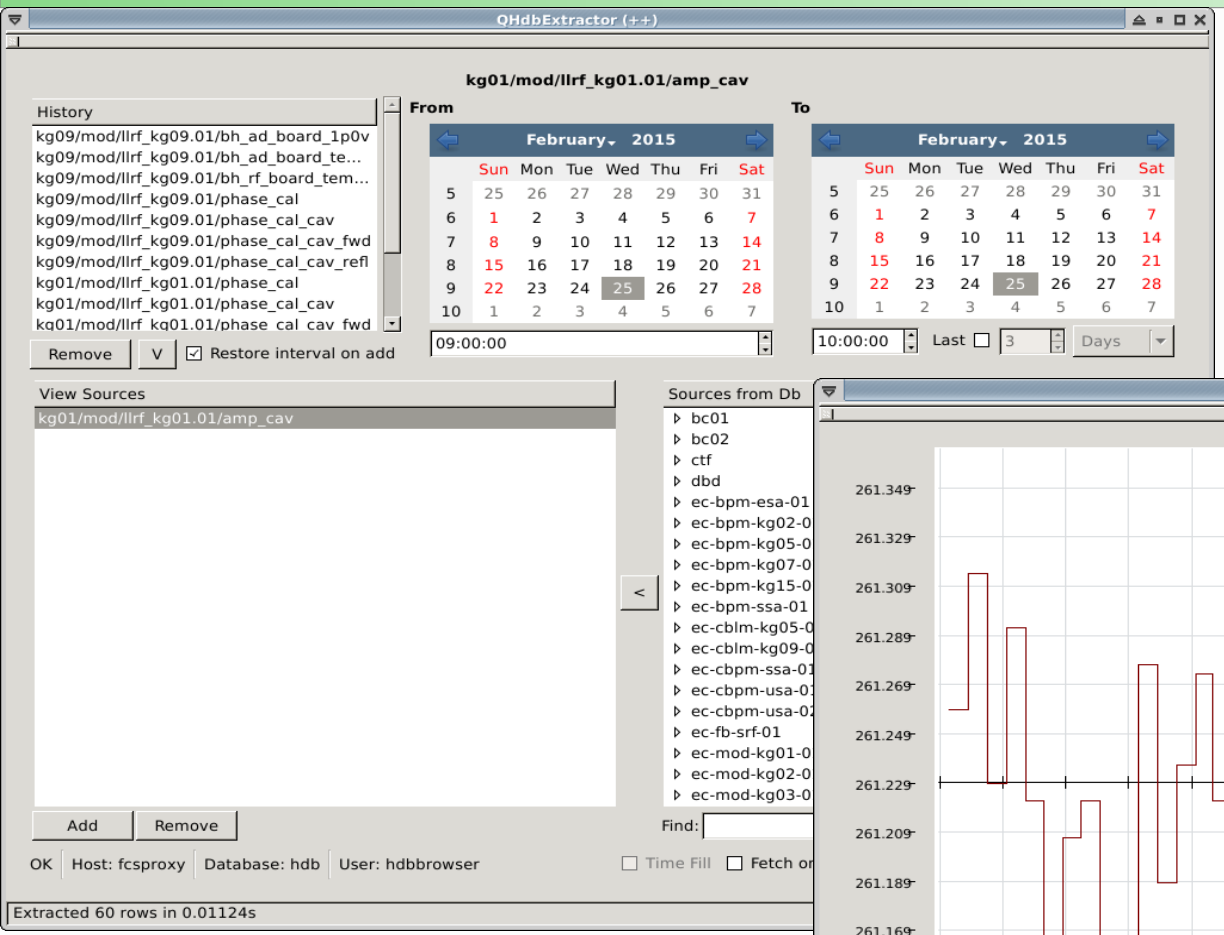


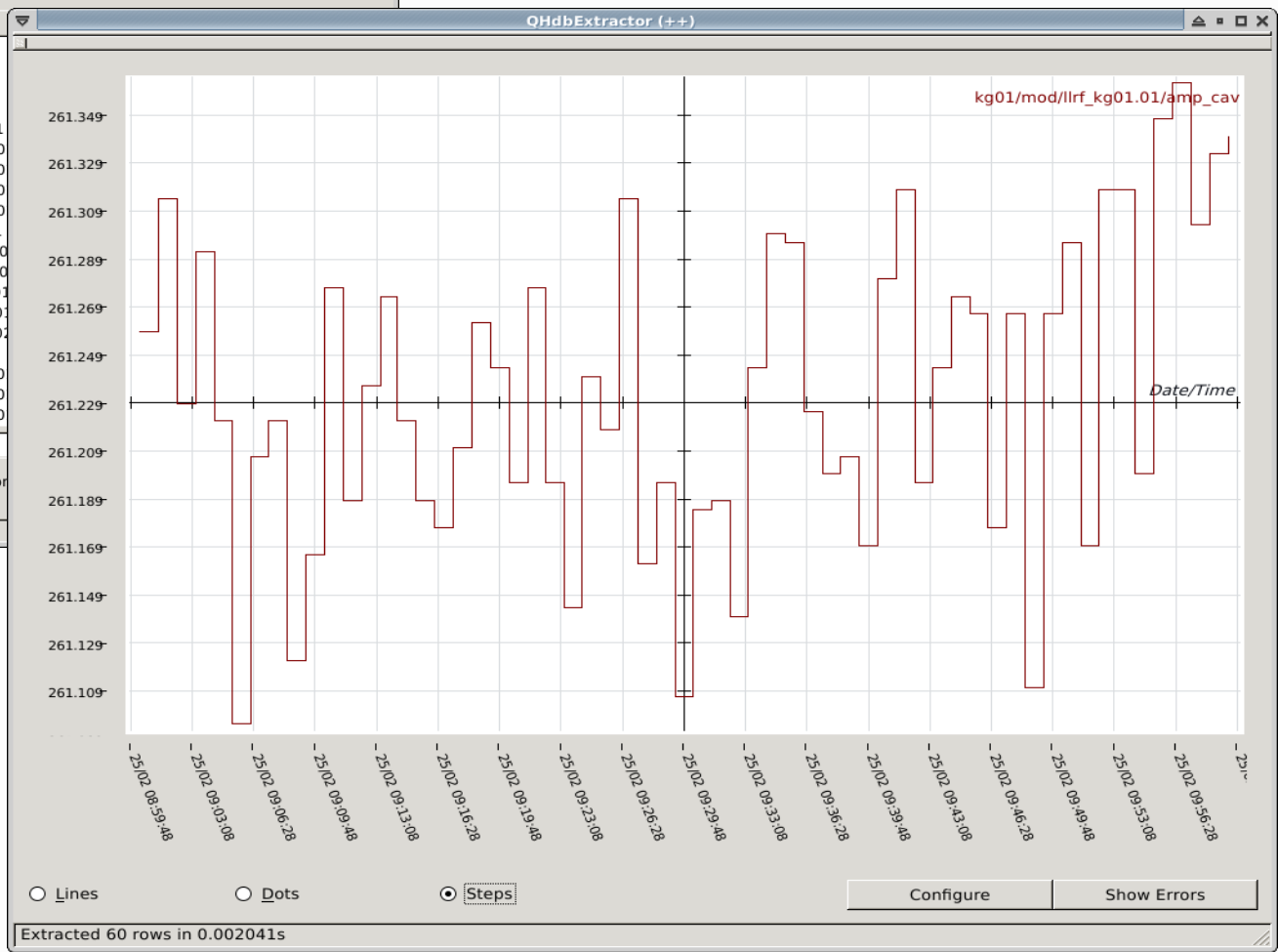
TANGO

Connecting things **together**

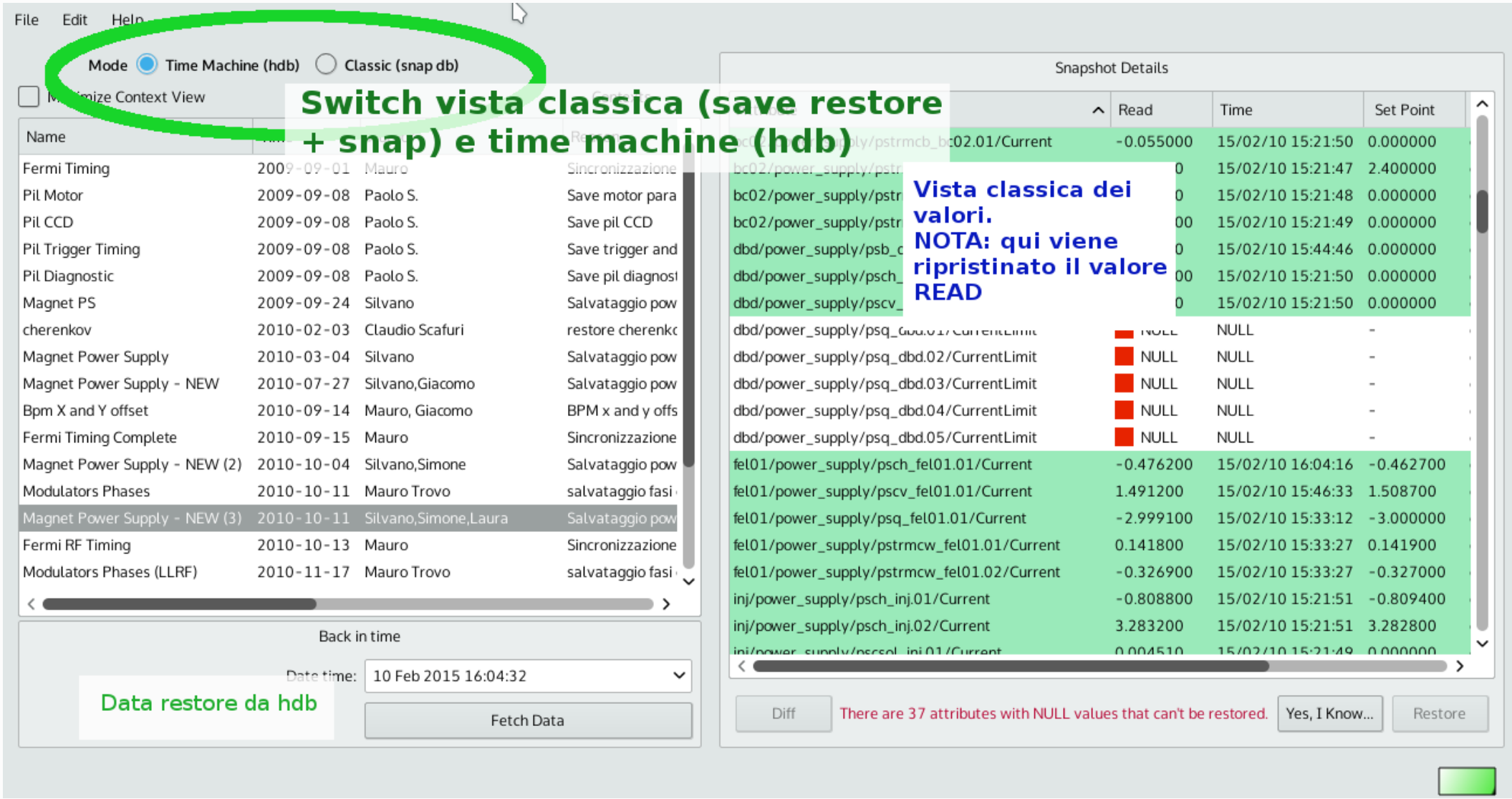
HDB++

Qhdbextractor GUI





The TimeMachine



What's missing?

Many things, including:

- Testing
- Documentation
- Installation instructions
- Packaging
  - Tarball
  - Debian packages