The libraries implementing the back-end are now dynamically loaded. A specific set of key=value entries can be specified for each back-end. Configuration for different back-ends is supported via the LibConfigurationManager TANGO device, that simplifies archiving configuration for each attribute, specifying when it has to be archived or not. Time-to-live parameter, available on a per-attribute basis. The Cassandra back-end is using CassandraTTI native feature underneath to implement the time-to-live. With dynamic loading of specific libraries switching back-ends can be done on-the-fly and is as simple as changing a property. Partition and maintenance scripts are now available for HDB++ and MySQL. The HDB++ tools, such as extraction libraries and GUIs, followed HDB++ evolution to help the user to take full advantage of the new features.

**HDB++**

- **TANGO archiving system** exploits the TANGO archive event, that can be triggered on threshold comparison and/or periodic basis. The HDB++ architecture is fully event based; therefore, a part of HDB++ setup consists of conveniently configure TANGO devices to send events as required. The TANGO archiving system consists of two main components, namely the EventSubscriber TANGO device, or archiving engine, and the ConfigurationManager TANGO device, that simplifies archiving configuration and management.

**Context-based archiving**

The support for context-based archiving allows to define an archiving strategy for each attribute. A strategy is a list of contexts for which the attribute has to be archived. When an EventSubscriber is set to a context, only attributes that have this specific context in their strategy are archived, and all the remaining attributes are automatically stopped. The strategy is saved in the AttributeList device property.

**Disk usage**

To reduce disk usage, when required. HDB++ can be configured with a reduced MySQL schema. Some of the columns of the tables used to support TANGO data types have been made optional: two of the three timestamp columns, e.g. the TANGO event timestamp and the insertion timestamp, the attribute quality and the error description carried by the event. The back-end abstraction library can be configured to ignore the optional columns (lightschema=true) or can autodetect the existing columns in the tables.

**HDB++ diagnostics GUI, FERMI.**

- EventSubscriber instances: 10
- ConfigurationManager instances: 3
- Number of attributes archived: ~2200
- Average events/minute rate: ~1200
- Average rate: ~8100
- Average events/minute: ~7300
- Archive disk size: ~400GB

**Graphical user interfaces**

The Configurator GUI supports specifying per-attribute archiving strategies and time-to-live. The HDB++ Viewer can now display the attribute configuration history and attribute quality factor.

**Web-based viewer for HDB++ under development at MaxIV, based on node.js, Babel, React and Redux** and managed by Webpack. The back-end uses aiohttp and Boker/datashader.

- HDB++ at ESRF
  - EventSubscriber instances: 36
  - ConfigurationManager instances: 1
  - Number of attributes archived: ~8200
  - Archive disk size: ~850GB

- Cassandra at ESRF: 3 nodes Production Datacenter, 3 nodes Analytics Datacenter

**Graph plot made with eGiga2m web based tool.**

- HDB++ TANGO ARCHIVING SYSTEM
- NEW DEVELOPMENTS FOR THE HDB++ TANGO ARCHIVING SYSTEM

L. Pivetta, G. Scalamera, G. Strangolino, L. Zambon - ELETTRA-Sincrotrone Trieste
R. Bourtembourg, J.L. Pons, P. Verdier - ESRF
S. Rubio-Manrique - ALBA

https://github.com/tango-controls
https://github.com/tango-controls/hdb++