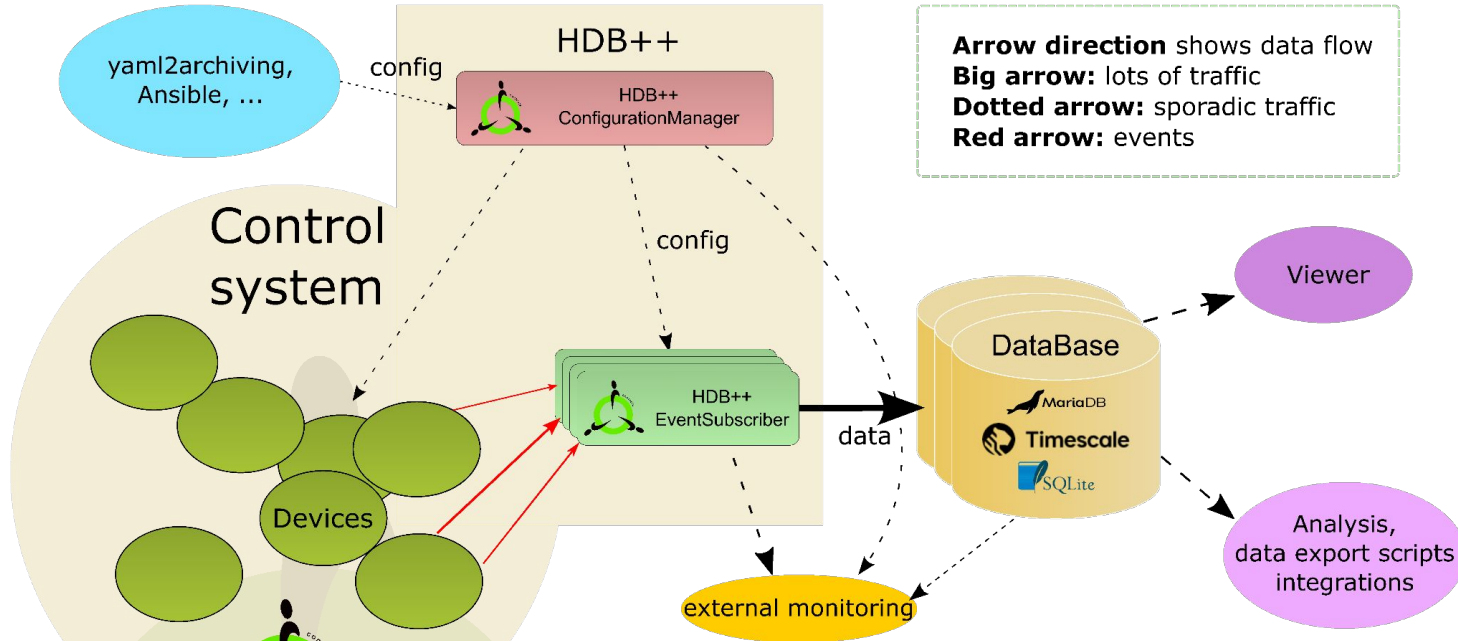


HDB++: What's new?

ALBA
DESY
Elettra
ESRF
MAX IV
SKAO
Soleil



HDB++ High-performance event-based Archiving System for Tango



The number of attributes per EventSubscriber is tuned depending on your events/s ratio and DB engine (see each institute slides).

Alternate "mods" like change-event archiving, manual insertion and client-based polling are also available

Supported backends

DB backend	Support level	Remarks
<u>Timescale</u>	Active	Recommended backend
<u>MySQL/MariaDB</u>	Active	Manual partitioning/maintenance
<u>Legacy MariaDB/MySQL</u>	Unsupported	Deprecated
<u>ElasticSearch</u>	Abandoned	
<u>PostgreSQL</u>	Not actively supported	Status unknown, timescale library should be compatible
<u>SQLite</u>	Not actively supported	Ideal for testing
<u>Cassandra</u>	Unsupported	Deprecated

Supported clients

HDB++ client	Platform	Supported backends	Support level
<u>Python extraction library</u> pyhdbpp	Web/Desktop	MSQL, MySQL, MariaDB, Timescale, legacy HDB/TDB	Active
<u>Java extraction library</u>	Web/Desktop	Oracle (legacy schema), PostgreSQL, MySQL (current and legacy schema), Cassandra	Active
Matlab binding available	Desktop	based on Java extraction library	Maintainer welcome
<u>C++ extraction library</u>	Web/Desktop	MySQL, MariaDB	Unsupported
<u>eGiga</u>	Web	MySQL, MariaDB, Timescale	Active
<u>HDB viewer</u>	Desktop	based on Java extraction library	Active
<u>Grafana</u> (with pyhdbpp)	Web	based on pyhdbpp-grafana-connector	Active
<u>Grafana</u> (direct query)	Web	Any DB that is supported by Grafana	Unsupported
<u>archviewer</u>	Web	Timescale	Active
<u>tango_browser</u>	Desktop	based on pyhdbpp	Active

Everything developed by the community is available at:
<https://gitlab.com/tango-controls/hdbpp>

tango-controls / hdbpp

This application is a GUI for ContextManager device server it is able to show context for all event subscribers It is also able to set the context pe...

H hdbpp-benchmark

H hdbpp-cm

Tango device server for the HDB++ Configuration Manager

H hdbpp-cm-es

Tango device server able to export HDB++ Configuration Manager and HDB++ Event Subscriber devices.

H hdbpp-configurator

HDB++ Configurator GUI and java device server

H hdbpp-es

Tango device server for the HDB++ Event Subscriber.

H hdbpp-metrics

HDB++ metrics work

H hdbpp-mysql-project

H hdbpp-pyhdbppperiodicarchiver

Device server to insert periodic data into hdb++ databases - requires libhdbppinsert - ask ALBA developers for updates controls-software@cell...

H hdbpp-tickets

The official place to create an issue/ticket related to HDB++ when it impacts several repositories or you don't know where to create your issue.

H hdbpp-timescale-project

hdbpp-timescale-project

H hdbpp-viewer

Java HDB++ viewer

L libhdbpp

Interface library for HDB++.

ALBA - Accelerators

Running HDB++ since 2018 (HDB/TDB since 2008); configured using PyTangoArchiving, accessed **pyhdbpp** (**libhdbpp-python**). (We did **NOT** migrate old archiving, we just added it as a **pyhdbpp backend**).

Using **56 CHANGE** Event Subscribers and **48 Periodic Archivers for legacy systems**.
PyHdbppPeriodicArchiver for periodic data and now **YATES** for buffer-based subscription.

HDB++ o MariaDB host storing 6 months of raw data (19779 attrs, 6 dbs, 6TB in total, partitions at 15 days)
All historical data is decimated to a secondary host (1 value every 10 seconds, **6TB in total since 2010**).

hdbacc: 943 attributes, 42 events/second, 437 GB, 4+4 archivers

hdbct: 3984 attributes, 202 events/second, 576 GB, 10+11 archivers

hdbdi: 3905 attributes, 874 events/second, 2198 GB, 12+12 archivers

hdbpc: 3127 attributes, 60 events/second, 1112 GB, 6+6 archivers

hdbrf: 3950 attributes, 226 events/second, 525 GB, 8+8 archivers

hdbvc: 4976 attributes, 1115 GB, 114 events/second, 16+15 archivers

hdbts: 883 attributes, 230GB, **TimescaleDB!**, 876 ev/second, 12 archivers

hdbts2: 883 attributes, 130GB, **TimescaleDB-light!**, 1039 ev/second, 1 archiver! (**YATES** by A.Ollé)

ALBA - Beamlines

- Agreed new data policy retention with beamlines. 2 years of full resolution data for all variables, then forever retention of data at 10s/value only for vacuum/eps values.
- All 13 beamlines (Tango9) using HDB++ on MariaDB
- Several problems solved between MariaDB and MySQL versions (MR in progress)

Fermi

HDB++ running since 2015

1 MySQL back-end, hdb++ schema

~17000 attributes from 8 Tango facilities

~9000 ev/minute; peaks up to 53.5K ev/minute

Context based archiving -> ~30 archiving strategies defined

59 EventSubscriber + 5 ConfigurationManager

~410 GB on disk - master (current + 2 previous years)

~410 GB + 1 TB on disk - replica (data from 2015)

Elettra

~~HDB++ running from 2016 to 2025~~

~~1 MySQL back-end, hdb++ schema (legacy HDB schema dropped 2021)~~

~~~6000 attributes~~

~~~5000 ev/minute~~

~~Context based archiving -> 7 archiving strategies defined~~

~~21 EventSubscriber + 1 ConfigurationManager~~

~~~460 GB on disk - master~~

~~~460 GB on disk - replica~~

Infrastructure (buildings facility)

1 MySQL back-end, hdb++ schema

~10000 attributes

18 EventSubscriber + 1 ConfigurationManager

ProxySQL

Used to make different DBs visible as a single one

For instance Fermi+Infrastructure, Elettra+Infrastructure, Fermi current (last 2 years) + Fermi archived (10 years) data

Elettra 2.0

New. Starting late 2026 - beginning of 2027

TimescaleDB ?



ESRF's Database setup (TimescaleDB)

- 1997 - 2018.
 - 1 database engine with 2 databases.
 - No replication
 - Successful recovery from a disk failure, database was lost but we could rebuild it from backups.
- From 2019.
 - Hdb++ with timescaledb backend.
 - Running on a cluster with 3 machines.
- Data compression is set up for data over 1 year old.
- Uses TTL to reduce disk usage, a lot of bugs were fixed!
- Database size ~ 9To
- Stores about 700 events/s.

Upgrade to PostgreSQL 18+ and TimescaleDB 2.26+ is planned during the summer shutdown

Configuration using “yml2archiving”

Interest in archiving at beamlines is growing - few beamlines have their own dashboards in Grafana to monitor the accelerator and the experiment status

Setup

- One HDB++ setup (single archiver) per BL
- One for accelerator (42 archivers, ~90% of data volume)
- Single Postgres database cluster, 3 nodes (1 write, 2 read-only)
- Separate schema for each control system

Some statistics

- ~ 24000 attributes
- ~ 3000 events per second
- ~ 20 control systems
- ~ 250 billion rows in TimescaleDB
- ~ 8TB disk space used (with compression)

Engineering Data Archive (EDA)

- Current deployments:
 - SKA Low (Australia)
 - SKA Mid (South Africa)
 - CI/CD clusters
- Based on Kubernetes & Helm/Helmfile
 - Timescale server in dedicated namespace
 - PV for Timescale data storage
 - Individual DBs per deployment available
 - Still not always used 😞 Not good when used in CI/CD pipelines: One DB, many runs → lots of garbage in same DB
- Retrieval
 - archviewer (questions: support for complex structures, displaying of quality factor)
 - pyhdbpp in Jupyter notebooks
 - pyhdbpp-rest/pyhdbpp-grafana-connector for Grafana & Octopus



| SPC cluster | MCCS cluster total | MCCS per station |
|-------------------------|---------------------------|--------------------------|
| 60 devices | devices | 79 devices |
| 57 device servers | 1169 device servers | 13 device servers |
| 220 attributes archived | 34195 attributes archived | 9000 attributes archived |

First results @SKA Low (~1300 of ~130k dipoles, "The 1% telescope" 😊)

- Single ES not enough (exactly as predicted by HDB++ community)
 - Will have to deploy more ES
 - Unfortunately deployment not flexible enough at the moment (Helm charts)
 - Currently "hacked" deployments

Despite all the issues: Users are really happy to have historical attribute data

Challenges:

- Data retrieval without leaking credentials
- "The data is not plotted the way I want it!" --> Direct DB access from Grafana

- **The system is scaling fine for insertions.**
- **Timescale proved to be a featurefull backend:**
 - Compression is very efficient ~ x10 times compression
 - Aggregates work - some work on averaging images needs to be done
 - Easy configuration
 - Low maintenance, it just works
 - Jobs backend to ease automation
 - In place updates working flawlessly
 - DB and extensions upgrades via the StackGres operator
 - PG 16.4, 18.1 and 18.4
- **No TTL policy yet agreed but we need it.**
- **Chunks size must be monitored**
- **Maintenance tasks, such as deleting attributes can take a lot of time and disk space, but they are manageable.**
- **Some problems with replicas staying behind:**
 - network/ceph/PostgreSQL related
- **Locks are frequent: timescaledb issue**

- Libhdb++-timescale
 - Bug fixes (duplicate events support, libpqxx7, ...)
 - CI/CD setup for automatic testing
- HdbEventSubscriber/HdbConfigurationManager:
 - Bug fixes (solved recurrent crashes)
 - CI/CD setup for automatic compilation
- pyhdbpp (mysql/mariadb/timescale/parquet)
 - Active development, new config modes
- pyhdbpp-grafana-connector: New middle layer between Grafana and pyhdbpp, proxy for connections
- New REST API! (spin-off from grafana connector to become a generic tool) (ongoing MR)

Python3 package for data extraction

> `pip3 install pyhdbpp + [backendlib]`

- Common API for MariaDB, MySQL and TimeScaleDB
- AbstractReader object provides generic extraction interface
- MultiDB reader allows to merge data from multiple data sources
- Dedicated DB back-end implementation is loaded at runtime
- Connection setup is stored in .yaml / environment or Tango properties
- **Taurus Widget** available! (pyqtgraph)

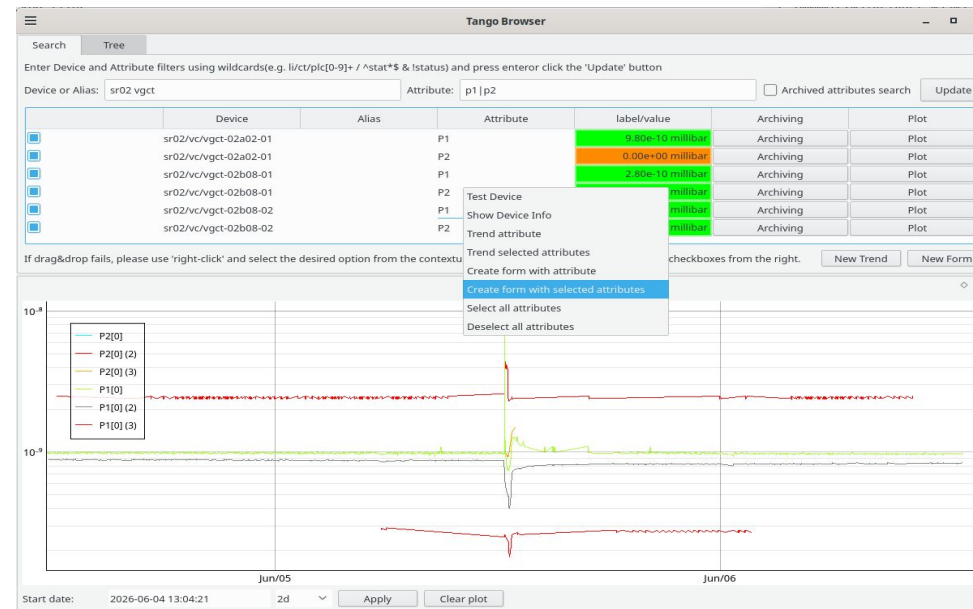
<https://gitlab.com/tango-controls/hdbpp/libhdbpp-python>

<https://gitlab.com/tango-controls/hdbpp/libhdbpp-tangobrowser>

```
import pyhdbpp

rd = pyhdbpp.get_default_reader() #or class/config
rd.get_attribute_values('sr/vc/01/pressure',
                        '2023-06-03', '2023-06-04')
```

```
Out: ((1685782812.169, 1.4e-08, 0),
      (1685782842.011, 1.5e-08, 0), ...)
```



- Elastic? SQLite? ... just inherit AbstractReader and contribute with your own implementation!
- New Tango browser based on it : <https://gitlab.com/tango-controls/hdbpp/libhdbpp-tangobrowser>
- Configuration API still pending (split in PyTangoArchiving; yaml2archive; pyhdbts)

MR!: New pyhdbpp configuration methods (by order of priority):

- **Tango database**
- **.yaml files**
- **Environment variables**
- **API arguments**

New **AbstractDB.isFeatureImplemented('FeatureName')** to allow backends to gradually support new features without breaking API (e.g. online decimation).

Current implemented features: **TimeFormat, TimeZone, Decimation**

Multidb schema capable of **plotting data from Timescale and Mariadb** (or any other backend) together. Used from Grafana DB to plot together data coming from different HDB++ systems (e.g. Machine + BL).

First results on comparing performance extraction from MariaDB and Timescale using pyhdbpp for both (same data on two servers):

- **TimescaleDB**, 2,688,034 values, **34 seconds**
- **MariaDB** (Mylsam), 2,699,752 values, **22 seconds!!**

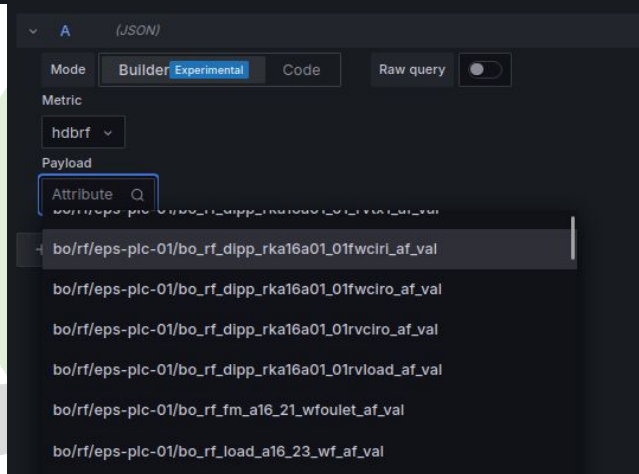
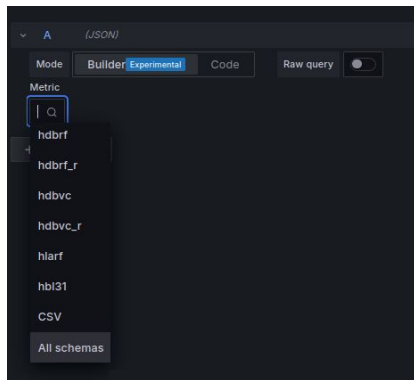
We suspect the difference in time may be related to ligh-schema and time-conversion methods.

Pyhdbpp Grafana plugin

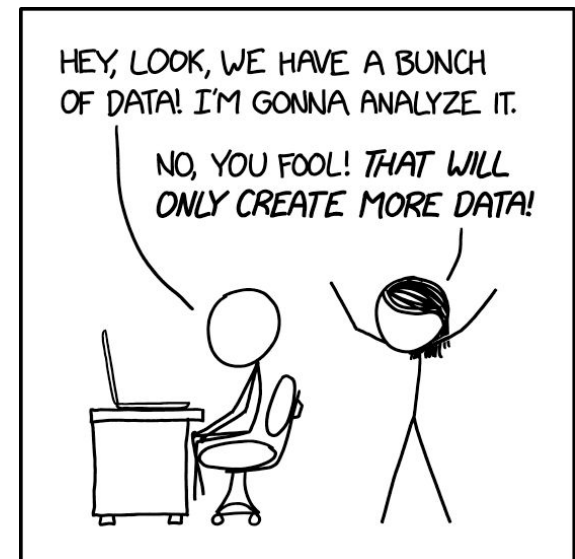
In addition to tango_browser and taurus, now grafana is available for plotting HDB++ (from mysql, mariadb, postgresql or timescaleDB backends).

Using pyhdbpp.multidb schema, it allows to plot data from multiple HDB++ databases. We are working on a more generic rest API with multiple enhancements. (Thanks SKAO for the MR)

<https://gitlab.com/tango-controls/hdbpp/libhdbpp-grafana-connector>



- **CI/CD**
 - Work in progress. HdbEventSubscriber, HdbConfigurationManager, libhdbpp-timescale, hdb++ timescale database docker images.
- **HDB++ workshop this Thursday!**
 - Current setup and questions in the morning
 - Discuss the future of HDB++ in the afternoon!
- **Explore the new Tango features:**
 - Alarm event
 - Telemetry for profiling
 - Software versions



<https://xkcd.com/2582/>

This artwork is licensed under a [Creative Commons Attribution-NonCommercial 2.5 License](#).

Keeping track of our HDB++ deployments to facilitate comparison and architectural design for new facilities

(always in progress, please add your results [here](#))

| Institute | ALBA | ESRF | SKAO MID-ITF | SKAO MID-PSI | SKAO LOW-ITF | SKAO AAVS3 | ELETTRA | ELETTRA |
|--------------------------------------|---|---|---|-------------------------------------|--------------------------|--------------------------|-------------------------------|---------------------------------|
| Db Engine | MariaDB | TimescaleDB | TimescaleDB | TimescaleDB | TimescaleDB | TimescaleDB | MySQL/InnoDB | MySQL/InnoDB |
| Used for | vacuum | Generic | Generic | Generic | Generic | Generic | Fermi FEL | Elettra SR |
| mode | :change_events, inserts | events+periodic (periodic is discouraged) | events+periodic (periodic is discouraged) | +periodic (periodic is discouraged) | periodic is discouraged) | periodic is discouraged) | archive events | archive events |
| Hdb++ version | 1.x | current | current | current | current | current | current | current |
| Packaging/distribution | debian package | compile-in-place | docker images | docker images | docker images | docker images | compile-in-place | compile-in-place |
| OS | Debian 9 | ubuntu 20.04 | Kubernetes | Kubernetes | Kubernetes | Kubernetes | Ubuntu 18.04 | Ubuntu 18.04 |
| RAM | 128 Gb | 128 Gb | Varies(32G) | Varies(2x128G) | Varies(32G) | Varies(2x128G) | 32 Gb master/64 Gb replica | 16 Gb master/64 Gb replica |
| N CPUs | 24 | 64 | Varies(16) | Varies(2x64) | Varies | Varies(2x128) | 8 master/4 replica | 8 master/4 replica |
| Db Size (Gb) | 514 | 4200 | 1.8 | 0.4 | 77 | 5 | 350 | 332 |
| Timespan | 3 months | 4 years 4 months | 6 months | 9 months | 1 month | 2 years | 2 years | 2 years |
| N attributes | 5159 | 17772 | | | | | 17326 | 6067 |
| N attributes pushing events | 4505 | all? | | | | | all | all |
| N scalar double attributes | 3461 | 15006 | | | | | | |
| from N devices | 806 | 6579 | ~10 | ~10 | ~10 | ~10 | | |
| N subscribers | 16 | 84 | 1 | 1 | 1 | 1 | 59 | 19 |
| partitioning | 322.4375 | 211.5714286 | 0 | 0 | 0 | 0 | 293.6610169 | 319.3157895 |
| max partition size | 30 days from couple of days for double to a month | | | | | | 2 months for double, 1 year c | 2 months for double, 1 year oth |
| events/second (hdb++es) | 64G | ~64Gb | | | | | 29 Gb | 35 Gb |
| events/second (stored) | 137.133 | 700 | Not measured yet | Not measured yet | Not measured yet | Not measured yet | 150 | 85 |
| events/second/attribute (average) | 0.03 | | | | | | 0.008 | 0.014 |
| events/second/subscriber (average) | 8.5708125 | 8.33333333 | #VALUE! | #VALUE! | #VALUE! | #VALUE! | 2.542372881 | 4.473684211 |
| max events/second/attribute | 14.36 | | | | | | 16.5 | 17 |
| max events/second/subscriber | 1 | | | | | | 2.8 | 1.5 |
| max events/second/attribute (stored) | | | | | | | | |
| max events/second/device | 452 | | | | | | | |
| bytes/event | 62018 | | | | | | | |
| bytes/second | | | | | | | | |

- #hdbpp channel on Mattermost (not for long, stay tuned for new tool announcement)
<https://mattermost.hzdr.de/tango-controls/channels/hdbpp>
- HDB++ announcement mailing list: hdbpp@tango-controls.org → [subscribe here](#)
- Report issues on <https://gitlab.com/tango-controls/hdbpp/hdbpp-tickets>
- Source code available under <https://gitlab.com/tango-controls/hdbpp>
- Docker images available at:
 - <https://developer.skao.int/projects/ska-tango-images/en/stable/how-to/hdbpp-with-docker-compose.html>
- Device servers, libraries and clients available in Conda-forge:

```
conda install -c conda-forge libhdbpp-timescale hdbpp-cm hdbpp-es
```

```
conda install -c conda-forge taurus pyhdbpp pyqtgraph
```

QUESTIONS?

Archiving Workshop:

Thursday at 9:30 at Marie Curie meeting Room!!

(if possible, follow the docker link and download/install the images in advance)

