HDB++ status

Lorenzo Pivetta

on behalf of the HDB++ collaboration

HDB++

- https://github.com/tango-controls-hdbpp

- Four meetings during 2020 (telco)
  - Last meeting: 2020.11.06
    - Improve documentation on readthedocs
    - Support TimescaleDB in PyTangoArchiving
    - Revisit work on the ELK back-end

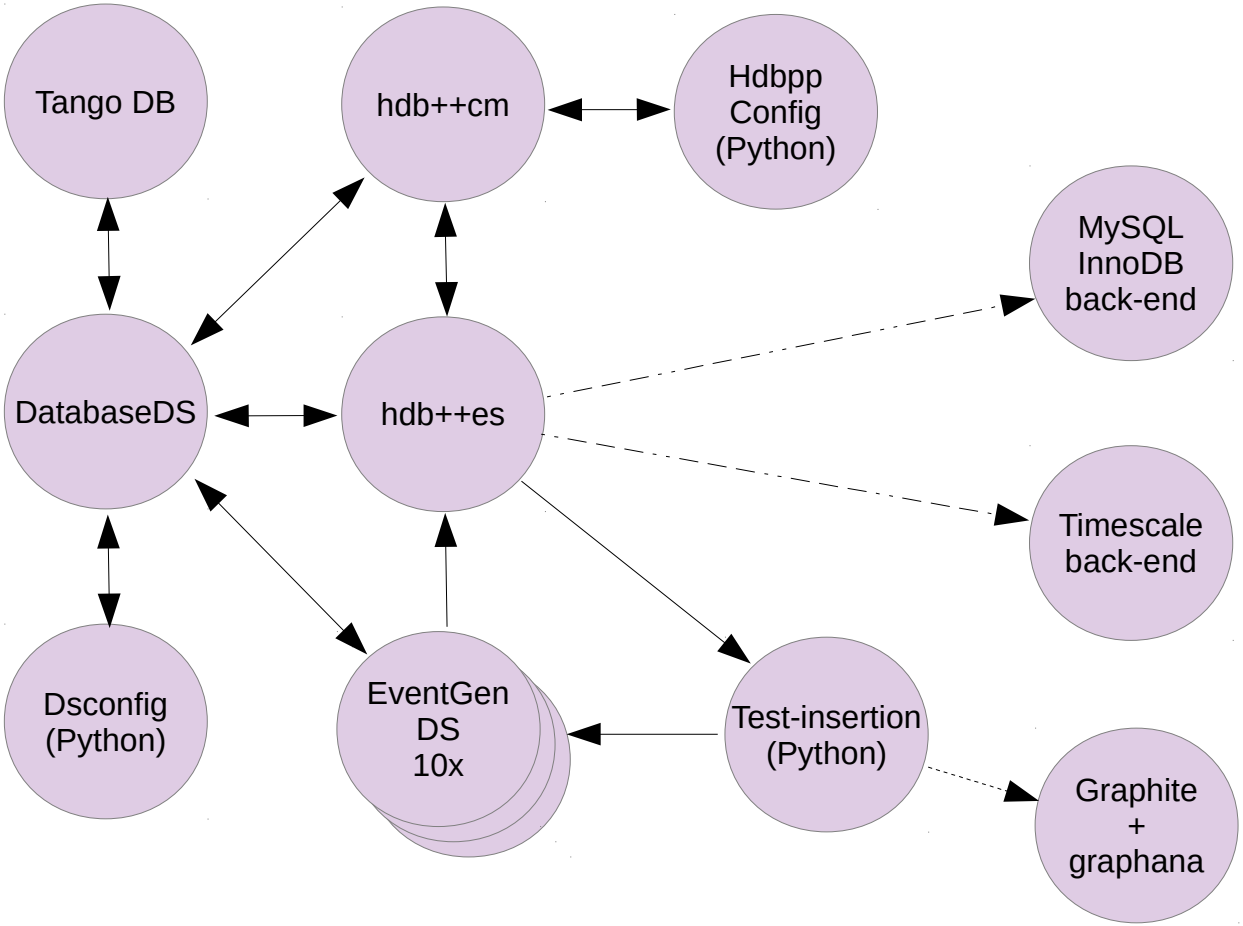
- HDB++ still complex to build
  - Get stable master on github
  - Provide a guide to build with cmake

- Require C++14

- Hdbpp-benchmark
  - Preliminary results with batch insertion – hit ZeroMQ HWM

HDB++

Benchmarking setup (Docker)



TimescaleDB
Max event rate ~1700 ev/s
Number of events archived **1341288**
Disk increase of DB partition **425** MB
Test Duration **1700** seconds

MySQL/InnoDB
Max event rate ~4000 ev/s
Number of events archived **3463359**
Disk increase of DB partition **164** MB
Test Duration **2500** seconds

Preliminary tests with batch insertion:

TimescaleDB / MySQL InnoDB
Max event rate ~12000 ev/s
(hit ZeroMQ HWM)

Additional investigation in progress

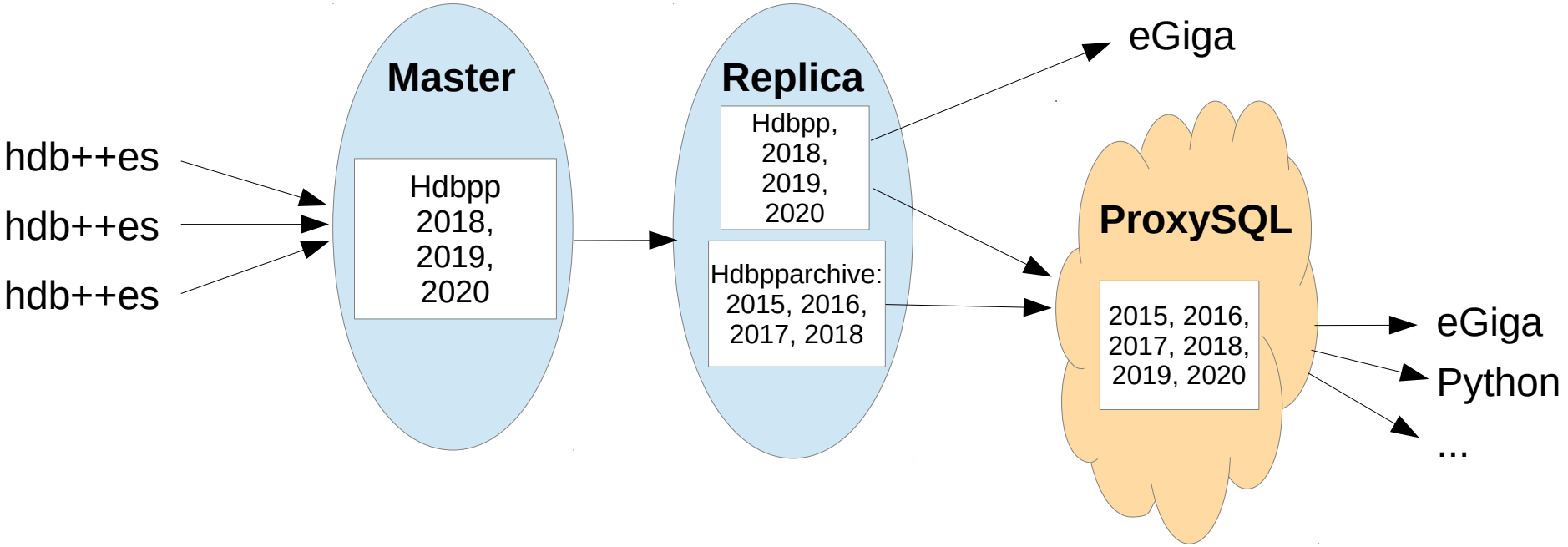
HDB++ @ Elettra

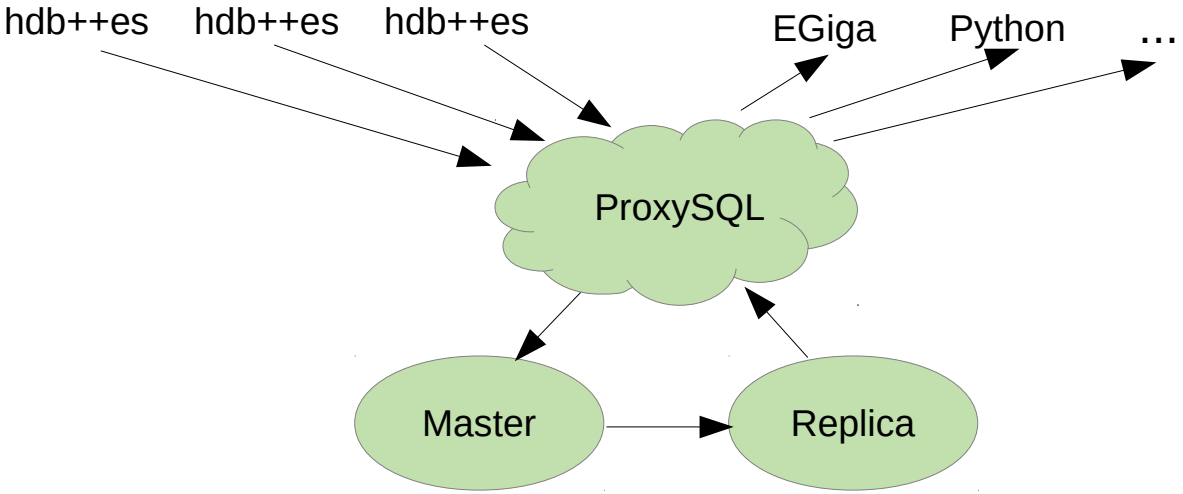
Fermi
- Running since 2015
- ~12300 attributes from 8 Tango facilities
- ~5000 ev/minute mean; peaks up to 48.5K ev/minute
- Context based archiving -> ~30 archiving strategies defined
- 54 EventSubscriber + 5 ConfigurationManager
- 1 MySQL back-end
- ~450 GB on disk

Elettra
- Running since 2016
- ~2650 attributes
- ~1850 ev/minute mean
- Legacy HDB schema, some Java HDB still archiving on the same DB
- 15 EventSubscriber + 1 ConfigurationManager
- 1 MySQL back-end
- ~700 GB on disk (includes old data)

Infrastructure (buildings facility)
- ~200 attributes (new, growing to ~1000)
- 1 MySQL back-end

HDB++ @ Elettra

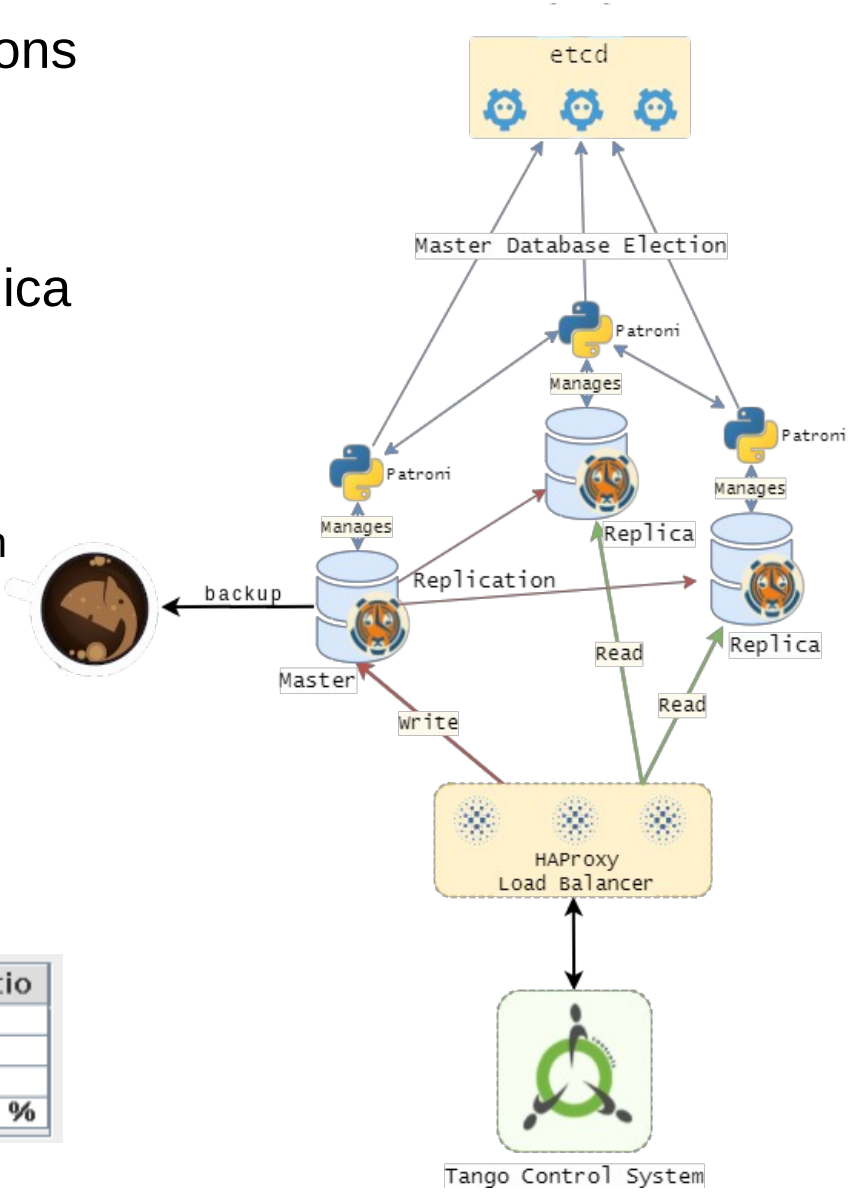




- Query caching
- Application layer proxy
- Fail-over support
- Query routing
- Query rewrite
- Advanced topology

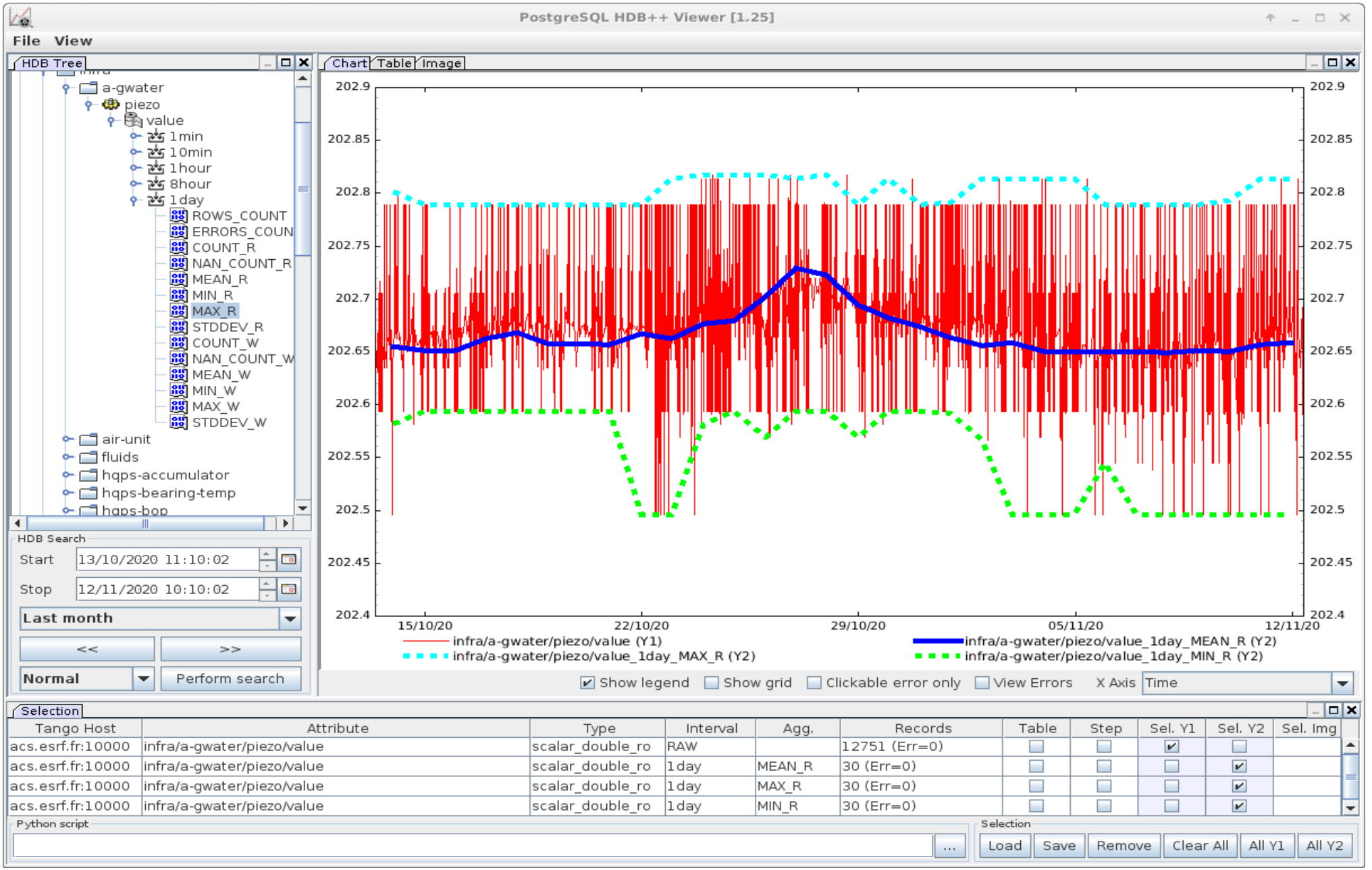
HDB++ @ ESRF

- High availability design based on recommendations by TimescaleDB
- Fully redundant, no single point of failure
- One entry point via HA proxy
- Our design uses a Master and one or more Replica nodes
  - All writes to Master via proxy
  - All reads to Replicas via proxy
  - Reduces load on Master when users query for data
  - An extra replica is solely used for backup using barman
- Features
  - TTL
  - Data aggregation on scalar and array
  - Enum support
- Key figures
  - Database size 1.6 TB
  - Insert/s ~500



	Managers	Subscribers	Attributes	Attr Ratio
Accelerator	1	31	8898	92.50 %
ID	27	27	582	6.05 %
D	15	15	139	1.45 %
TOTAL	43	73	9619	100.00 %

HDB++ @ ESRF

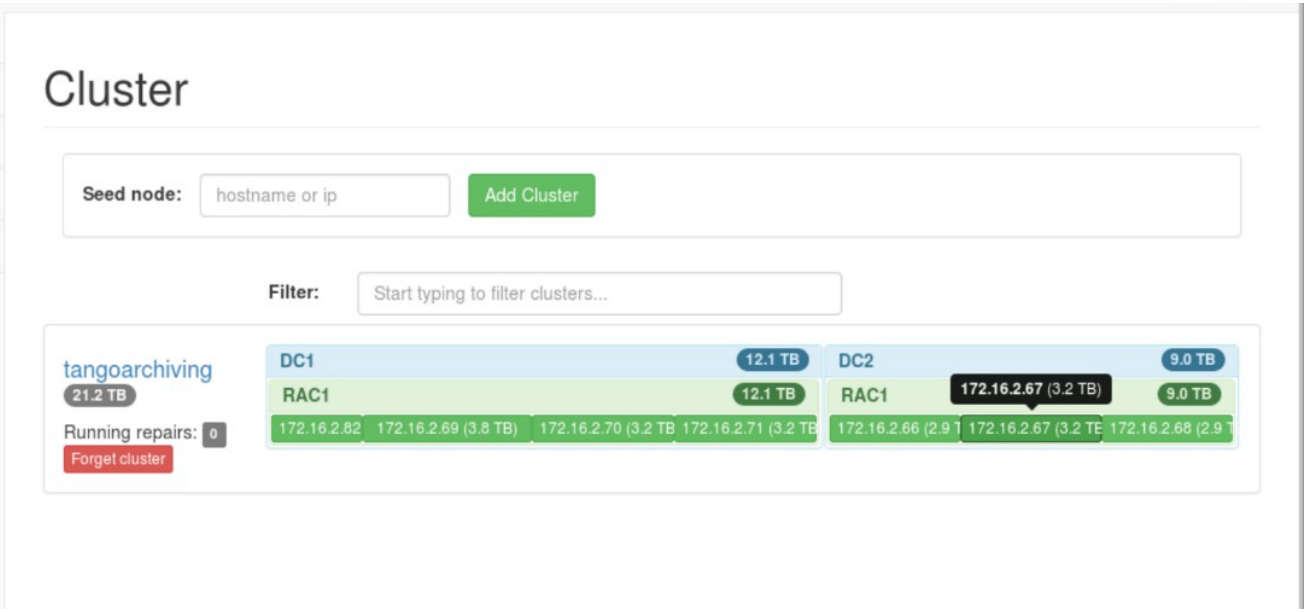

PostgreSQL HDB++ Viewer [1.25]
File View
HDB Tree
a-gwater
piezo
value
1min
10min
1hour
8hour
1day
ROWS_COUNT
ERRORS_COUN
COUNT_R
NAN_COUNT_R
MEAN_R
MIN_R
MAX_R
STDDEV_R
COUNT_W
NAN_COUNT_W
MEAN_W
MIN_W
MAX_W
STDDEV_W
air-unit
fluids
hqps-accumulator
hqps-bearing-temp
hqps-bop
HDB Search
Start 13/10/2020 11:10:02
Stop 12/11/2020 10:10:02
Last month
<<
>>
Normal
Perform search
Chart Table Image
202.9
202.85
202.8
202.75
202.7
202.65
202.6
202.55
202.5
202.45
202.4
15/10/20
22/10/20
29/10/20
05/11/20
12/11/20
infra/a-gwater/piezo/value (Y1)
infra/a-gwater/piezo/value_1day_MEAN_R (Y2)
infra/a-gwater/piezo/value_1day_MAX_R (Y2)
infra/a-gwater/piezo/value_1day_MIN_R (Y2)
Show legend
Show grid
Clickable error only
View Errors
X Axis
Time
Selection
Tango Host | Attribute | Type | Interval | Agg. | Records | Table | Step | Sel. Y1 | Sel. Y2 | Sel. Img
acs.esrf.fr:10000 | infra/a-gwater/piezo/value | scalar_double_ro | RAW | | 12751 (Err=0)
acs.esrf.fr:10000 | infra/a-gwater/piezo/value | scalar_double_ro | 1day | MEAN_R | 30 (Err=0)
acs.esrf.fr:10000 | infra/a-gwater/piezo/value | scalar_double_ro | 1day | MAX_R | 30 (Err=0)
acs.esrf.fr:10000 | infra/a-gwater/piezo/value | scalar_double_ro | 1day | MIN_R | 30 (Err=0)
Python script
...
Selection
Load
Save
Remove
Clear All
All Y1
All Y2

HDB++ @ SKA

- SKA-docker project contains the Docker files for running the archiver on a Mariadb backend
  - https://gitlab.com/ska-telescope/ska-docker
- One helm chart developed for running the HDB++ application
  - The result k8s application is composed by the following services
    - One configuration manager
    - One event subscriber
    - Mariadb
- Automatic testing implemented in BDD
- Future work:
  - Implement scalability mechanisms
  - Updates to the latest changes for the libraries
  - Have a noSql backend database (Elasticsearch)

HDB++ @ MAX IV

- Central Cassanda database
- HDB++ running in accelerators and 14 beamlines
- Close to 5 TB of data
- Evaluating migration to TimescaleDB
  + Better scalability
  + Better mainteinability
  + Community support
  - Data migration
  - New client development



HDB++ @ ALBA

- Archiving split into 6 MariaDB databases (1 server) for easier and faster maintenance
- Not using archive_event, archiving on change instead to improve devices performance
- 13366 attributes, 340 G/month, 6 DB, 1-2 partitions per month, 10 subscribers per DB + 10 periodic archivers

Partitioning and decimation

- Tables partitioned monthly
- Data older than 6 months decimated to a separate DB
- indexing/partitions/decimation methods in PyTangoArchiving.hdbpp.maintenance

Data split in 6 Databases

- **hdbacc** 969 attrs / 35G month
- **hdbct** 1977 attrs / 70G month
- **hdbdi** 796 attrs / 75G month
- **hdbrf** 3493 attrs / 50G month
- **hdbpc** 1977 attrs / 10G month
- **hdbvc** 4154 attrs / 100G month

Archiving clients

- https://github.com/tango-controls/PyTangoArchiving library used for extraction
- eGiga used on beamlines
- shell/python tools plus qwt-based taurus widgets, pyqtgraph on development

HDB++ @ ALBA

Periodic archiving on HDB++

- ALBA developed an archiver for inserting periodical data without events
  - https://github.com/ALBA-Synchrotron/PyHdbppPeriodicArchiver
  - https://github.com/ALBA-Synchrotron/libhdbppinsert
- Configuration via PyTangoArchiving.HDBpp api (*periodic* methods)
- Use cases
  - Archiving through firewall
  - Fix period archiving required by some users
  - Legacy control system with bad polling/notifd performance