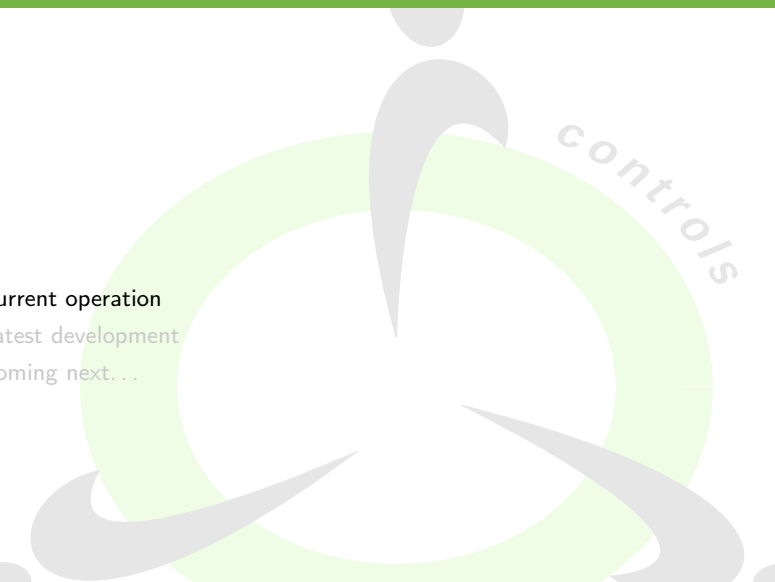


# ESRF - Hdb++ operation

---

Damien Lacoste  
June 10 2020



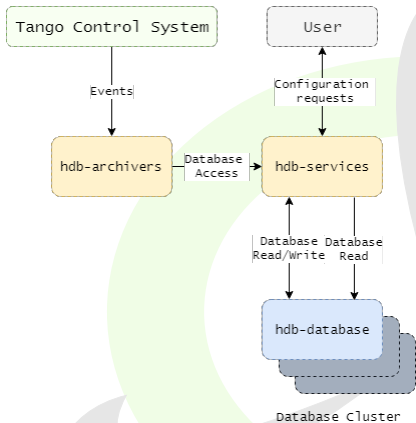
Current operation

Latest development

Coming next...

controls

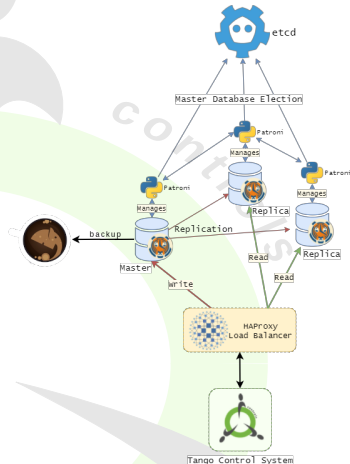
# Hdbpp project architecture



- ▶ Deployment is on virtual/real servers to allow us to utilise high end ex Cassandra servers for database nodes.
- ▶ Each database is deployed on its own server with its Patroni process. Allows the node to be down without taking out other critical software.
- ▶ Opted for a single instance of etcd initially, with option to cluster at a later point. (recommended)
- ▶ Deployment was split logically as follows :
  - ▶ **hdb-archivers** aggregates the Tango Device Servers into one or more virtual servers.
  - ▶ **hdb-services** aggregates all the service software onto a single server. Can be scaled if required.
- ▶ We can not scale write performance, but read performance can be scaled with additional Replica nodes.

# Cluster design

- ▶ Based on recommendations by TimescaleDB
- ▶ Single point of access via proxy
- ▶ Our design uses a Master and one or more Replica nodes.
  - ▶ All writes to Master via proxy.
  - ▶ All reads to Replica's via proxy.
  - ▶ Reduces load on Master when users query for data.
  - ▶ An extra replica is solely used for backup using barman.
- ▶ On Master failure, one Replica is promoted to Master.
  - ▶ Proxy is dynamically updated via Confd (see ClusterDesign document) on promotion.
- ▶ Each database node is deployed with an instance of Patroni.
  - ▶ Patroni manages PostgreSQL process itself.
  - ▶ Replication is also handled via Patroni automatically
- ▶ New failure points, etcd, HAProxy and Patroni managed by systemd to partially mitigate problems.





Current operation  
Latest development  
Coming next...

controls

# Aggregates, DevEnum support

- ▶ Aggregates using continuous aggregates from timescaleDB
  - ▶ Aggregates data is kept even after deleting data with TTL feature.
  - ▶ Aggregates computation performed in place.
  - ▶ Real time aggregates available on last timescaledb version.
- ▶ Trouble with arrays aggregates.
  - ▶ Need to implement helper methods in the db to compute them in place.
  - ▶ Long computation time.
  - ▶ Uses huge amount of memory and disk space to complete.
- ▶ DevEnum support through insert hooks in the database.

Connected to http://hdb-services:10666/apl/v1

| General                     | Cluster nodes | Database        |
|-----------------------------|---------------|-----------------|
| hdb                         |               |                 |
| Database size<br>639.729 GB |               |                 |
| Attributes number<br>8434   |               |                 |
| TTL                         |               | Backup          |
| Never executed.             |               | Never executed. |

- ▶ React frontend on a rest backend
- ▶ Simple dashboard with key metrics
  - ▶ Size and row numbers
  - ▶ Cluster status
  - ▶ Services status
- ▶ Custom alarms levels

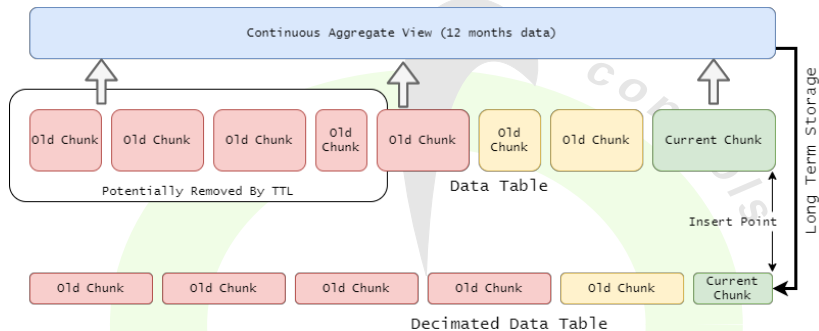


Current operation  
Latest development  
Coming next...

controls



# Long term storage



- ▶ Can we keep all the data in one table?
- ▶ TimescaleDb compression is quite performant

- ▶ <https://github.com/tango-controls-hdbpp/hdbpp-timescale-project>
- ▶ Build system based on cmake can grab all the dependencies and build the whole project in one go.
- ▶ Docker images available to run the hdbpp timescaledb database for testing purpose, or real deployment ?
- ▶ All services can be run in a self contained docker image or deployed on a machine.

A large, light gray stylized figure of a person stands in the background. The figure has a simple oval head, a rounded torso, and a pointed bottom. A large, light green circular ring is positioned around the figure's waist. The word "controls" is written in a light gray, sans-serif font, curving along the right side of the green ring.

Any Questions?

Thanks !