ESS/MAX-IV Control System Workshop 2023

Data archiving. Selection criterion and data retention policies for control systems.

Agenda

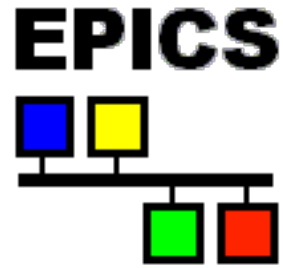
- 09:00 – Start of workshop
- 09:10 – ESS presentation (João Paulo Martins)
- 09:20 – MAX-IV presentation (Johan Försberg)
- 09:30 – Group discussion
- 10:00 – Coffee break
- 11:30 – Group presentations
- 12:00 – Lunch

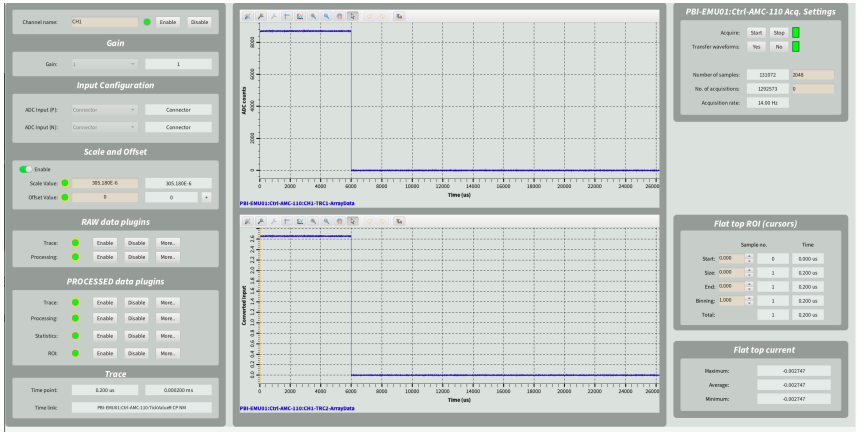
ESS Introduction

- Slides from latest ICALEPCS: “Status of the European Spallation Neutron Source Controls” (Timo Korhonen)

EPICS

- ESS uses EPICS as its “software package” for the Control System;
- EPICS has a simple architecture based on servers (IOCs), network and clients;
- The IOC is the local controller software:
  - Handles the communication/interface to hardware;
  - Handles the network communication within the Control System Network (server/publisher);
  - Models the device it controls in a “Process Database” made of “records” with a number of “fields”;
  - Almost every “field” of an IOC record is accessible via network as a “Process Variable (PV)”
  - Each PV must have a unique name that will serve as its identifier to the Control System;
- Clients and servers (IOCs) transactions are fundamentally reading/writing PV values;

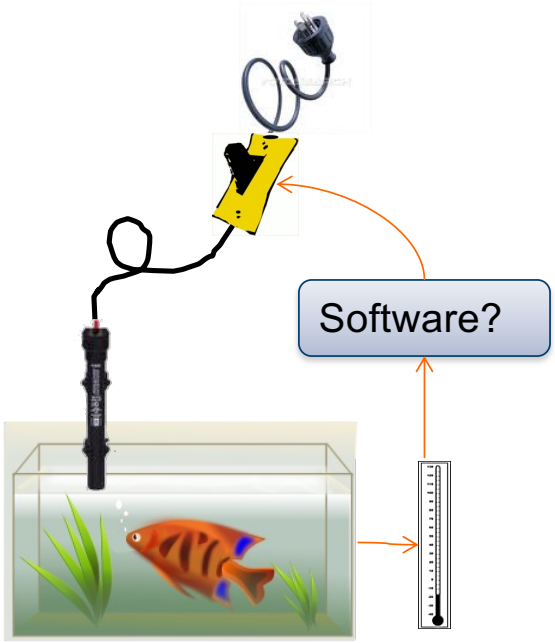


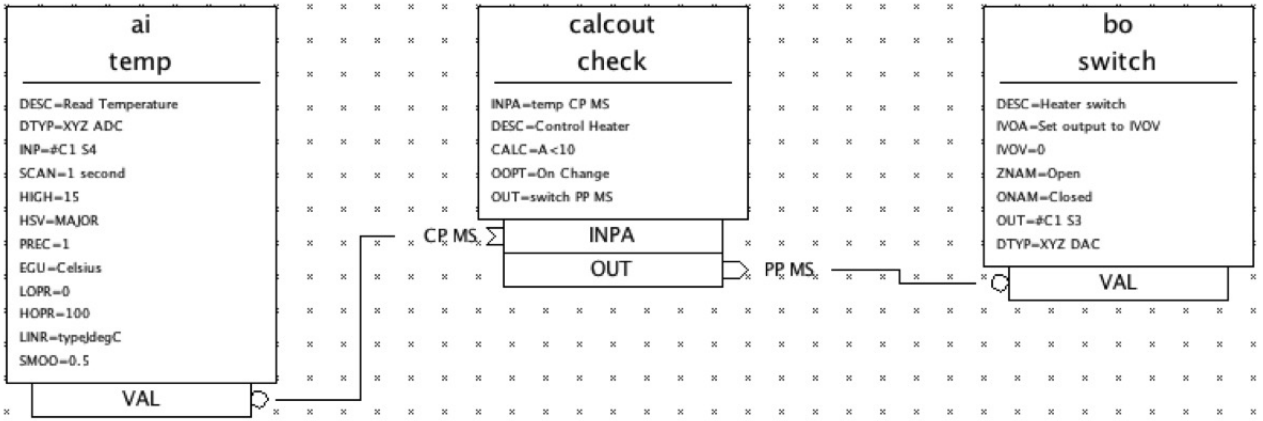


EUROPEAN SPALLATION SOURCE

MAX IV

Example of an IOC: Fishtank





EUROPEAN SPALLATION SOURCE

MAX IV

The Process Variable (PV)

- The PV is the data structure that represents a variable of the Control System:
  - The value of a sensor;
  - The command to turn ON/OFF a device;
  - The current position of a motion system;
  - The set-point of the position of a motion system;
  - The name of the current shift-leader operator;
- The entire facility is “parametrized” in terms of PVs;
- Any client in the EPICS network can read/write a PV just by knowing its unique name;
- PV and Record are different things!
- The archiving system is archiving PVs;

```
NTScalar :=

structure
    scalar_t    value
    string      descriptor  :opt
    alarm_t     alarm       :opt
    time_t      timeStamp   :opt
    display_t   display     :opt
    control_t   control     :opt
```

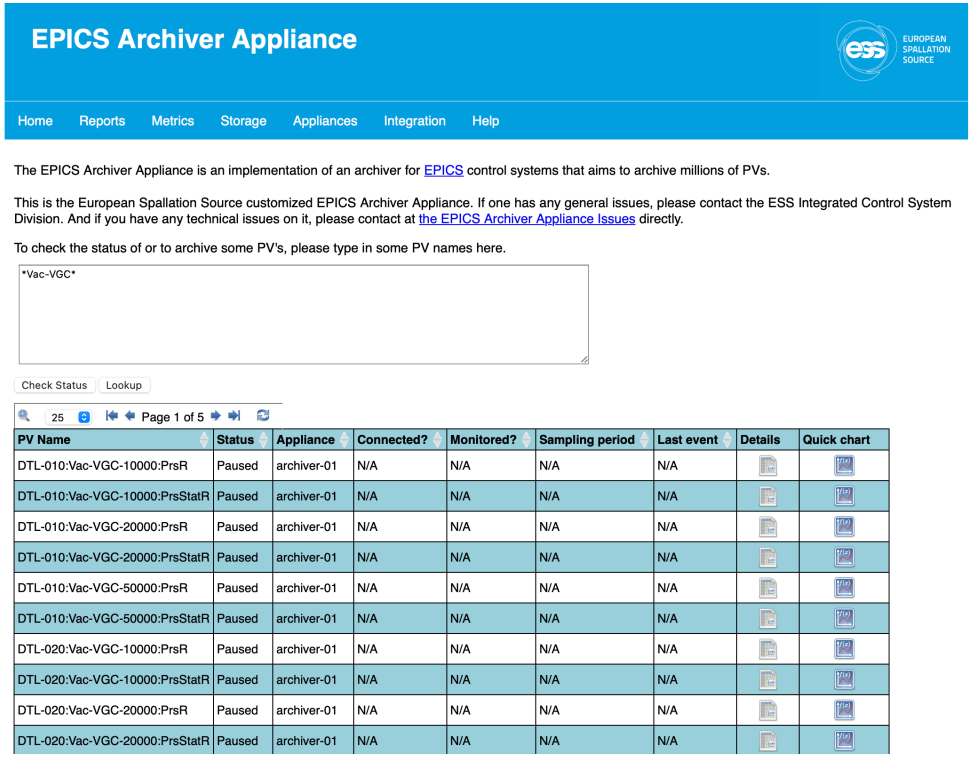
```
NTScalarArray :=

structure
    scalar_t[]  value
    string      descriptor  :opt
    alarm_t     alarm       :opt
    time_t      timeStamp   :opt
    display_t   display     :opt
    control_t   control     :opt
```

ess EUROPEAN SPALLATION SOURCE

MAX IV

The Archiver Appliance

- The Archiver Appliance is the main service that captures the PV values and stores them into a time-series data package into persistent storage;
- The Archiver Appliance also provides additional features for the user to interact with the data:
  - Inserting new PVs to the archiving system*;
  - Checking the archiving status of a certain PV;
  - Data visualization via browser;
  - API to fetch historic data sets;
  - Integrated to CS-Studio (GUI platform);
- The Archiver Appliance is a client in the Control System topology;
- The Archiver is fundamentally used to store EPICS data of the entire ESS facility, which does not include the Neutron Experiments;

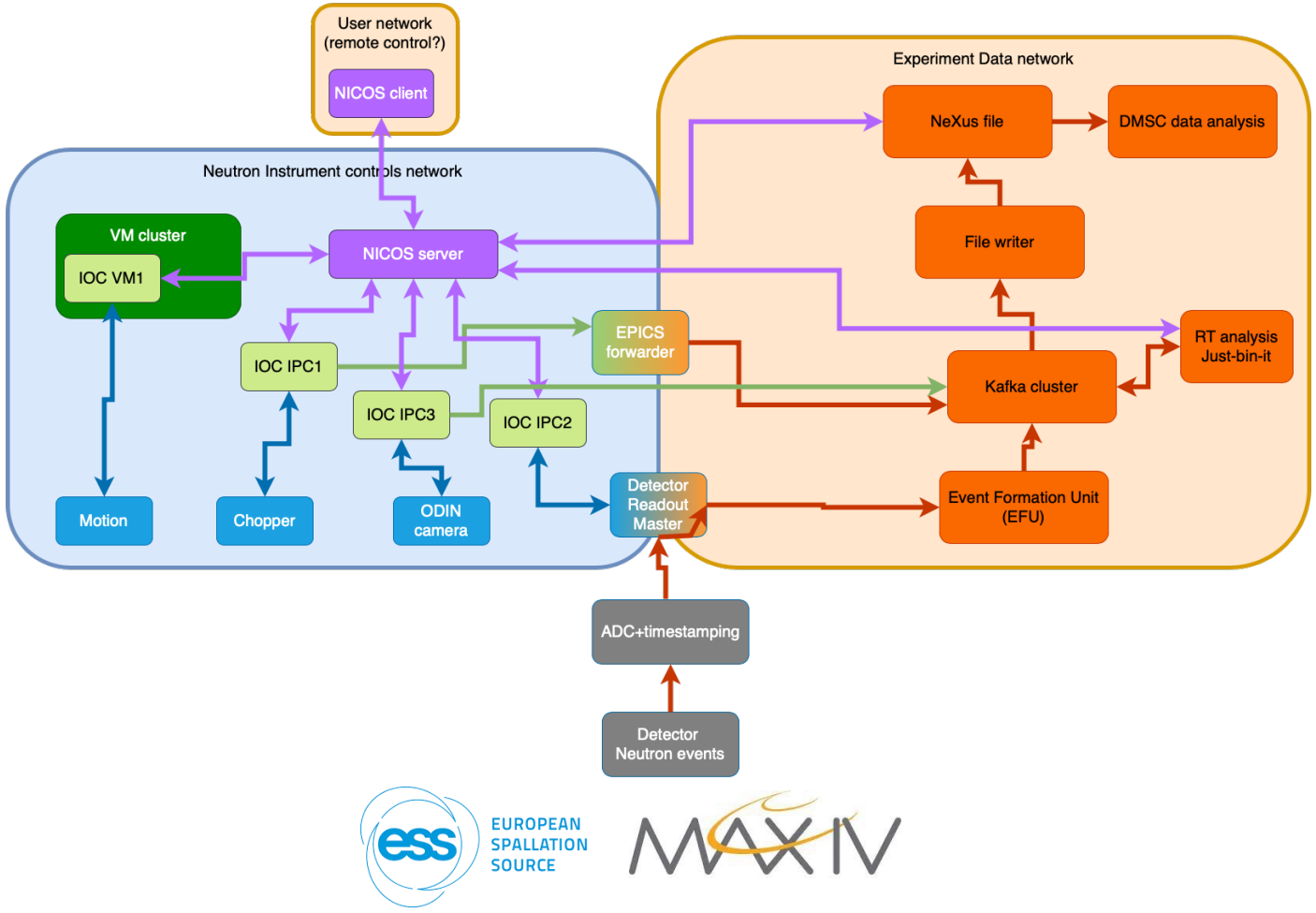


PV Name	Status	Appliance	Connected?	Monitored?	Sampling period	Last event	Details	Quick chart
DTL-010:Vac-VGC-10000:PrsR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-010:Vac-VGC-10000:PrsStatR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-010:Vac-VGC-20000:PrsR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-010:Vac-VGC-20000:PrsStatR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-010:Vac-VGC-50000:PrsR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-010:Vac-VGC-50000:PrsStatR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-020:Vac-VGC-10000:PrsR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-020:Vac-VGC-10000:PrsStatR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-020:Vac-VGC-20000:PrsR	Paused	archiver-01	N/A	N/A	N/A	N/A		
DTL-020:Vac-VGC-20000:PrsStatR	Paused	archiver-01	N/A	N/A	N/A	N/A		

EUROPEAN SPALLATION SOURCE

MAX IV

Neutron Experimental Data - Appendix



Observations about archiving at ESS…

- Well defined procedure to include new PVs to the archiving cluster; 👍
  - Git repositories with lists of the PVs to be archived, inclusions are done via merge request approved by a group of people;
  - Still requires some level of manual and repetitive tasks as more systems are deployed; Work on going on a method for the IOCs to present themselves the PVs to be archived;
- Web interface provide simple and useful features for most of the clients; 👍
- Excellent integration with the main EPICS client, the CS-Studio application; 👍
  - "Data browser" app allows easy visualization of trends: just set the PV name and the period to be plotted;
- Which PVs should be added to the Archiver? Who defines? Integrator or System Owner? 👎
  - No clear definition of who is responsible to configure de Archiving of a particular system once it is deployed;
- Dimensioning, planning and defining data retention policies – Infrastructure and Software teams are left "on their on"; 👎
- Very difficult to balance the users (System Owners) requirements* with the practical implementation;
  - "We don't know upfront what we need! We need all the data, all the time in order to find out" 👎

ess EUROPEAN SPALLATION SOURCE

MAX IV

Observations about archiving at ESS…

- Very poor management of the active PVs included in the Archiver; 👎
  - Hundreds of PVs that are inactive for many reasons (PV name has changed, system was redesigned, or even for unknown reasons);
  - Long shutdown periods with systems filling the storage with noise data;
- How to handle arrays? 👍👎
  - The EPICS Archiver Appliance can archive arrays, however the native data visualization tools do not support; User needs to fetch data using API;
  - ESS is a pulsed machine (14 Hz) with some diagnostics systems that are scaled to hundreds: BPM, BCM, BLM, LLRF, etc.;
  - Most of these systems are comprised of ADCs with sampling rates ranging from 1 MHz to 250 MHz; Since the window of events is ~4ms every 71ms, there is a desire to archive as much as possible of the raw data from the digitizers;
  - We have a separate cluster just for arrays, but the scale of the machine makes it unfeasible to keep adding new PVs;
  - Specific solutions for arrays are being developed by ICS;

MAX-IV Presentation


MAX IV

MAX IV/ESS Workshop
Data archiving

Johan.forsberg@maxiv.lu.se

Nov 17 2023

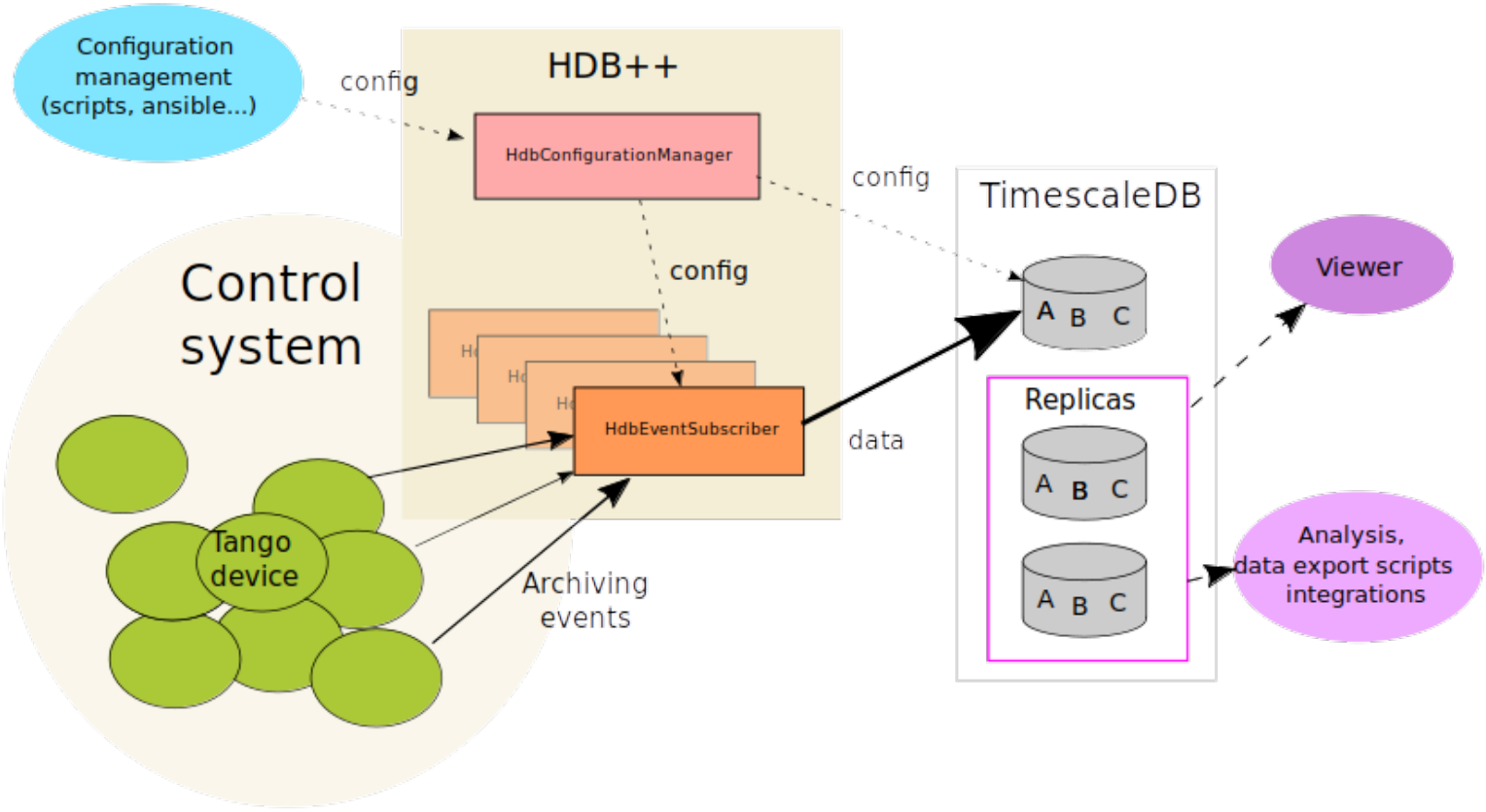
Archiving

- Long term storage of scalar (and some array) data
- “Reasonable” rate; e.g. up to a few Hz, but mostly lower (e.g. 1 per minute)
- For monitoring, detecting long term trends, statistics, troubleshooting, etc.
- Not “experimental” data, or data acquisition
- Mainly for personnel, not end users

HDB++ Tango archiving system

- General system for storing historical values of Tango attributes, “non-experimental” data.
- Developed mainly at ESRF (Grenoble) and Elettra (Trieste).
- Consists of Tango devices written in C++
- In use at MAX IV since 2016 or so
- Database backend originally Cassandra, moving to TimescaleDB

MAX IV Archiving system



Arrow direction shows data flow
Arrow size: amount of traffic
Dotted arrow: sporadic traffic

Some statistics

- One separate database per control system (all in the same DB cluster).
- ~16000 attributes archived in the accellerator, some hundreds for each of 16 beamlines
- Receiving around 3000 events per second in the machine, much less on beamlines (~10/s)
- ~100B events per year stored
- Right now DB is ~7TB on disk, but there is an ongoing migration, and also compression is not fully enabled yet.

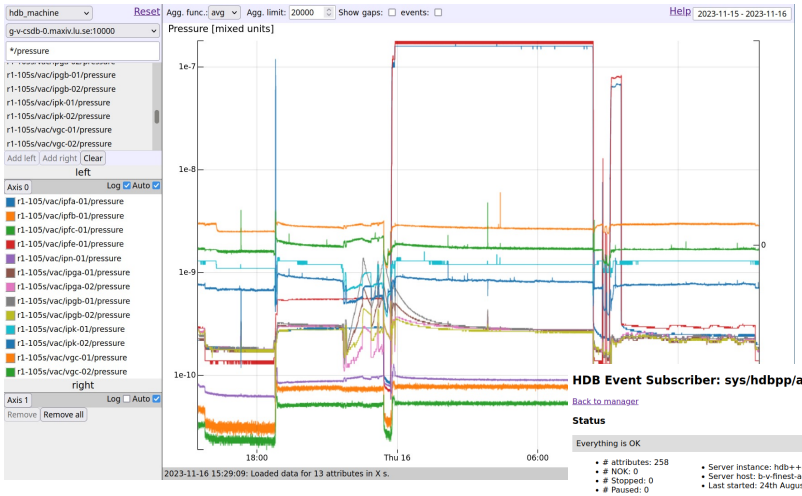
Configuration tool

- HDB++ configures each **attribute** individually
- Need for a “higher level” configuration system
- Tango attributes are grouped into “**devices**”, which in turn each belong to a “**class**”
- Configuration on the **class** level, using default configurations that can be automatically applied to all or some devices of that class.
- Possible to **filter** on device name, etc
- Inspired by e.g. Ansible; show differences, apply only needed changes.

yaml2archiving

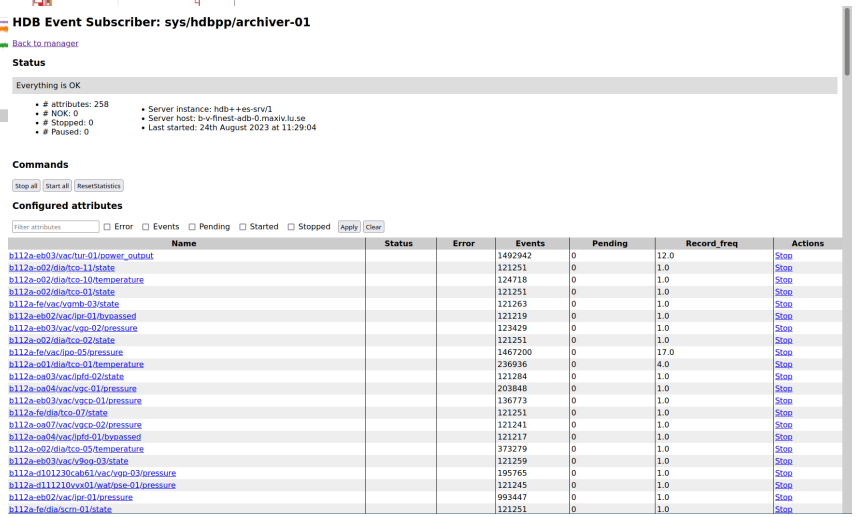
```
# Header about the archiving system
db: "g-v-csdb-0.maxiv.lu.se:10000"
manager: hdb/config/1
archiver: AUX/KITS/ARCHIVER-01

configuration:
  TangoTest:  # class
    filtering:
      # further limit to specific things
      device: "sys/tg_test/.*"
    attributes:
      # archiving config for matching attributes
      short_scalar_ro:
        archive_period: 6000
        polling_period: 3000
        archive_strategy: SERVICE
      short_scalar:
        archive_period: 7000
        polling_period: 3000
        archive_rel_change: 5
```

Some other tools



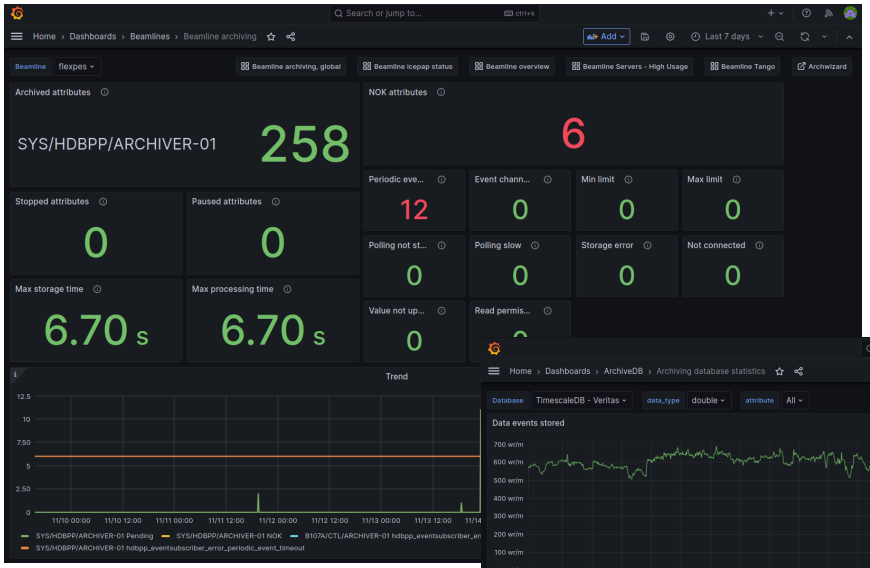
“Archviewer”
Web based archive data viewer

Loads data directly from DB using aggregated queries.



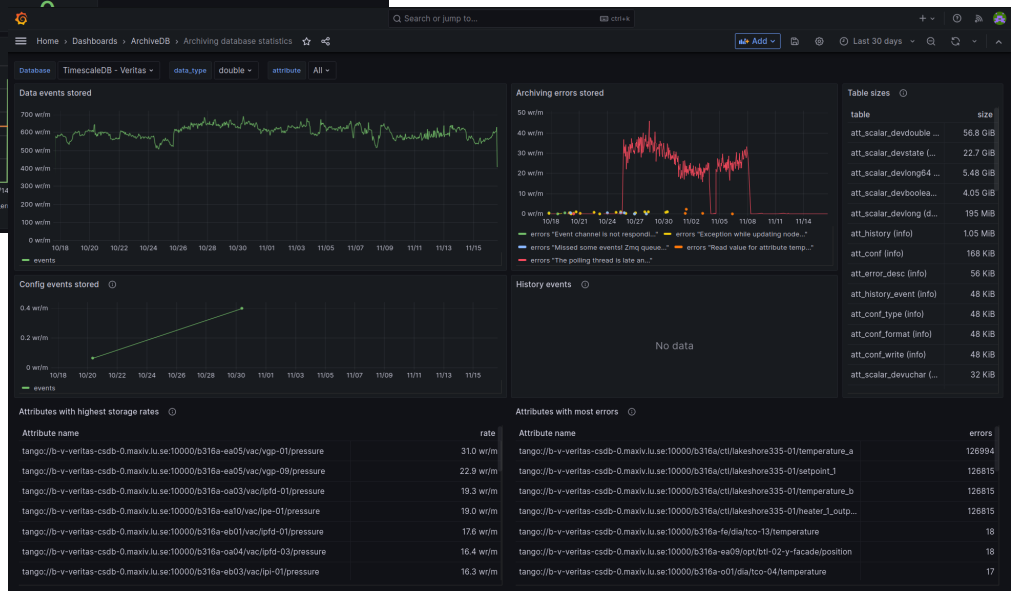
“Archwizard”
Web based monitoring tool
Shows current configuration, statistics and errors.

Monitoring the archivers



Data collection using Prometheus
(Victoria metrics)

Dashboards using Grafana



Just like with archiving;
You don't need it all the time, but when something happens you're very happy you have it...

Selection criterion

Tango has the following archiving related settings for an attribute:

- How often to check the value (a.k.a ”polling”)
- How often to archive the value, at least
- Filters that archive whenever the checked value has changed more than X (absolute or in percent) since last check.

We tend to use both period (e.g. 1 minute) and some combination of filters, to pick up quick changes.

Selection criterion

Most Tango devices at MAX IV are of standardized types. E.g. vacuum, diagnostics, cooling, etc. Here, the subsystem responsible choose what to archive and at what rate. These are used as defaults everywhere.

For more specialized devices it is usually up to the responsible party (e.g. beamline scientist) to choose archiving settings

Archiving settings are reviewed by ICT before deployment, to prevent bad settings, overloading etc.

We try to keep settings reasonable and standardized

Git + Gitlab merge requests

Looking to automate the process more

Retention policy

- Right now, all data is kept indefinitely (since ~2017)
- No plan to change this generally
- Future higher rate archiving will likely make it necessary to use a ”time to live” setting for some attributes. This is supported by HDB++.
- Continous aggregates in TimescaleDB can be used to keep “low res” data for TTL attributes.

General observations

- Archiving is messy. It has to handle many things, and scale up.
- Users only notice things missing/broken in archiving when they need historical data, and then it's too late. The rest of the time, most people don't care and therefore don't prioritize it.
- Structure the configuration and use versioning. This makes it more clear when, why and by who a change was made, easy to update defaults, etc.
- Apply the configurations often, to keep it in sync with reality.
- Review configuration changes carefully. Bad configuration can affect performance, waste resources, etc.
- Be conservative. The archiving system will live for a long time, and keep growing. Do it gradually.
- Monitor everything. The first step to fixing problems is to know about them! (We know about lots of problems...)
- Databases are complex to set up, maintain and tune. Especially large ones.

MAX IV

That’s all folks!

Group Activity

- **What does this group consider to be “best practises” regarding selection of data for archiving for control systems? What about data retention policy?**
- **For each facility (MAX IV and ESS) discuss and recommend at least two actions to improve the handling of archived data, data retention and performance/stability of archiver backend.**
- **Describe a scenario where incorrect or suboptimal handling of archiver data selection and retention can cause loss of beam time. How is the loss of beam time caused? What can we do to reduce the risk for loss of beam time? If beam time is lost despite our best effort how do we recover quickly?**

Discussion Topics (suggestion)

- Technology: frameworks, software packages, file formats, Big Data
- Configuration management
- Archiving large datasets (arrays)
- Handling the lack of well defined requirements from stakeholder
- Infrastructure dimensioning (storage, network)
- More..?

Group Activity Results – Group 1

1) What is considered as "best practices"?
- Scripts to update the config regularly (for device servers/IOCs, archiving). Regular review.
- Versioning of the config scripts.
- Naming conventions (document and tools to check, dbformat).

2) Recommended actions?
- Renaming policies as this happens and is not trivial. Changing types also.
- Injection of default values in the IOC
- Decimation of data.

3) Loss of beam time?
- Make device servers more robust for fetching data.
- Overload of network with too much data requests. We can lose control of a system.

How to avoid?
- Better tests of the IOCs.
- Reduce frequency of publishing of PV. + Decimation of the PV for vis.
- LTTB (Largest triangle three buckets)
- Keep raw data in SDS; Reduced PV in the Archiver (vis)

How to recover?
- Stop one of the archiver instances

Group Activity Results – Group 2

1) What is considered as "best practices"?
Data management work group
Archiving, monitoring, cycling
Think first about How we're supposed to use the data
The idea of class is very useful
Try to find the real purpose of the archiver
Establish a culture. psychological cost
Representative from accelerator physicist for accelerator control system

2) Recommended actions?
Data management group that includes accelerator and controls
Decide on policy
Archiving, monitoring, cycling: Do temperatures really need to be acquired in >20Hz or so?
Each may have different needs. It has to be considered separately and should not enforce each others need.
Raw cycling data should not be protected from GUI clients or Archivers.
Define guideline for everyone who is involved in reading/using the data/service.
Study of the bandwidth of archiving
Supervisory systems

3) Loss of beam time?
Issue at ESS 14Hz, stopped some high demanding PVs which was a good recovery action.

Additional Notes

- @MaxIV they usually need the data for at least 1y.
- time series DBs use postgress engine
- Compression is extremely useful for reducing the data size.

@MaxIV the experimental data is not archived in Archiver.
Rather the study owners retrieve and archive data on their MATLAB scripts.

@ESS scientists are using archiver as accelerator experiments which is not the case for MaxIV

@MaxIV there is a system to archive the waveform of BPMs but that's still on demand only during studies.

Psychological effect: Put a price value on archival of data -> it's not for free

Setpoint and readback. In Tango they're tied together. It's usual that we miss archiving setpoint in EPICS

Failover policy at ESS. What happens if the instance goes down?

Big data management: SDS HDF?

Experience with SDS-like @MaxIV: They had similar system where they gathered the data in HDF format distributed among different systems. At the end it wasn't useful as the amount of data was too big to be useful.

It has to be thoroughly studied if we need the data or not. **Data management work group**

Data policy: Do temperatures really need to be acquired in >20Hz or so?

Archiving, monitoring, cycling may have different needs. It has to be considered separately and should not enforce each others need.
For instance the raw cycling data should not be protected from GUI clients or Archivers.