OAK RIDGE National Laboratory

Archive

Oct. 2023

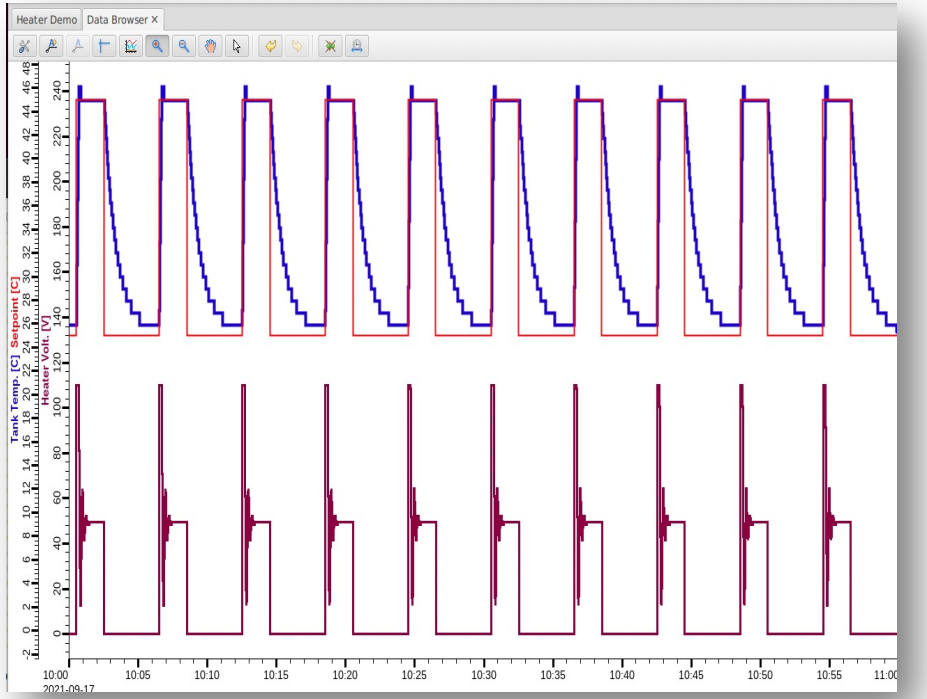
Kay Kasemir

ORNL is managed by UT-Battelle, LLC for the US Department of Energy

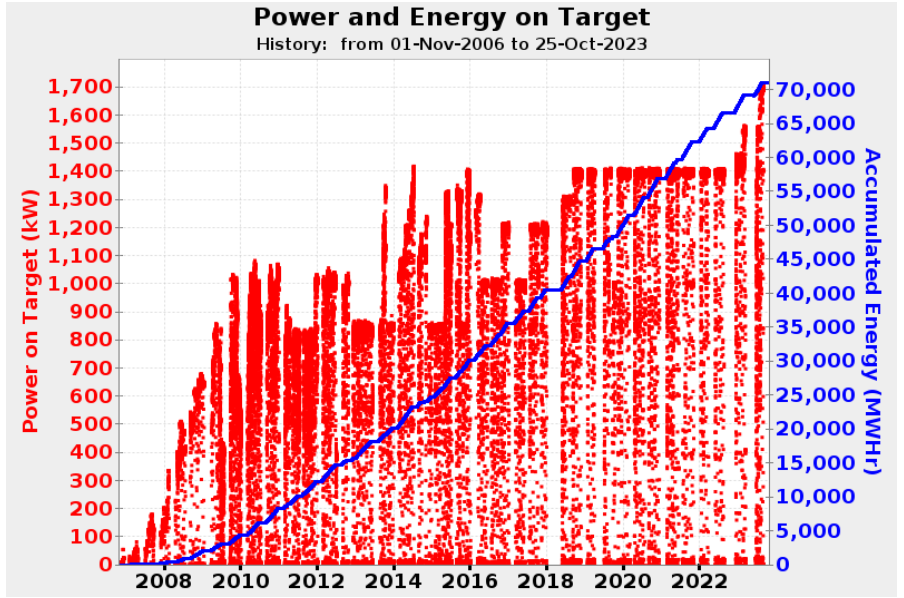
U.S. DEPARTMENT OF ENERGY

End-User View of Archive

- Databrowser can show history
  - Easy, "Scroll back" on time axis



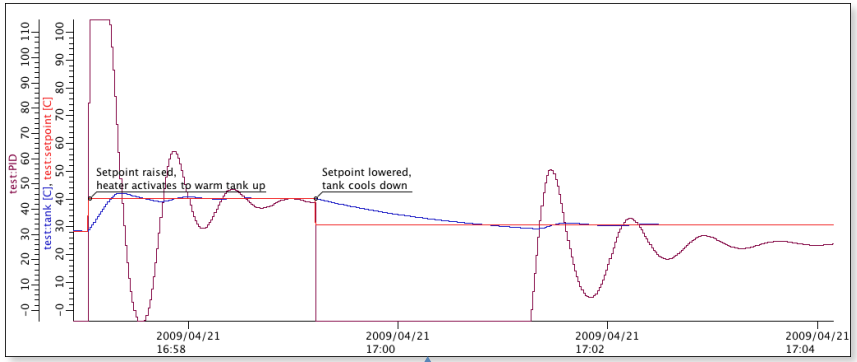
- May access history from Python, Java, … to perform your own analysis
  - Write a program to do that

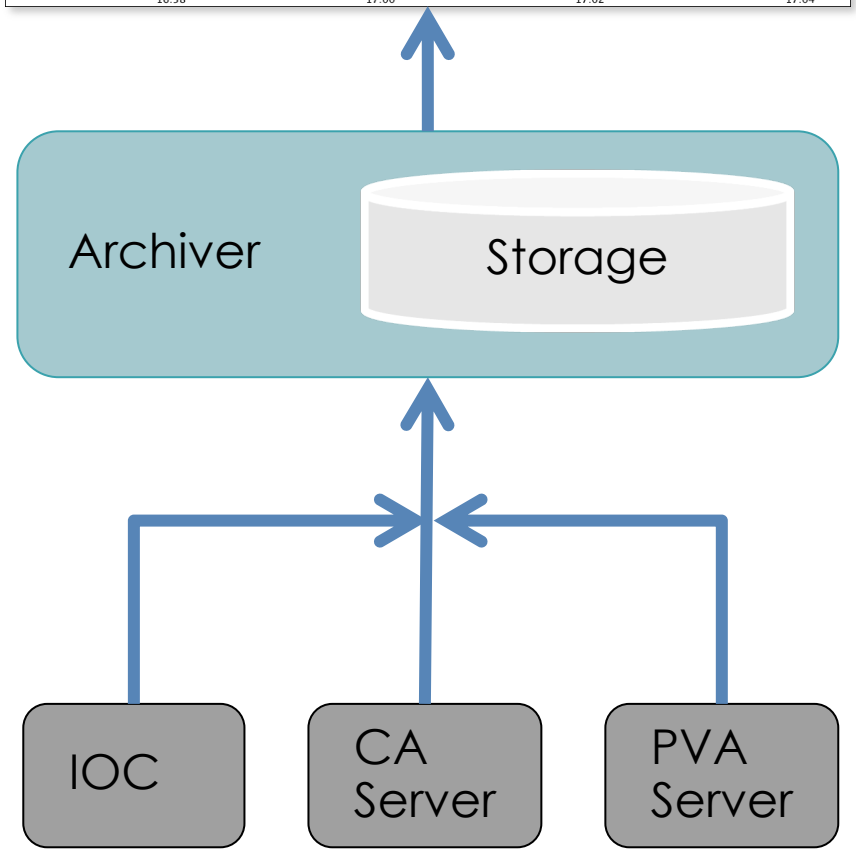


Basic Technical View of Archive

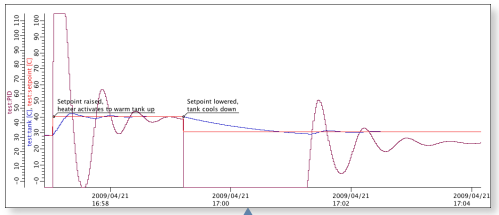
Archiver

- Reads PVs via Channel- or PVAccess into Storage
- Provides history from Storage





More Detail



Configuration:

Which channels?
How?

Archive Engine (Service)

Read PVs,
write samples to storage

CA, PVA

IOC

Storage

Theory:
Tons of fast storage

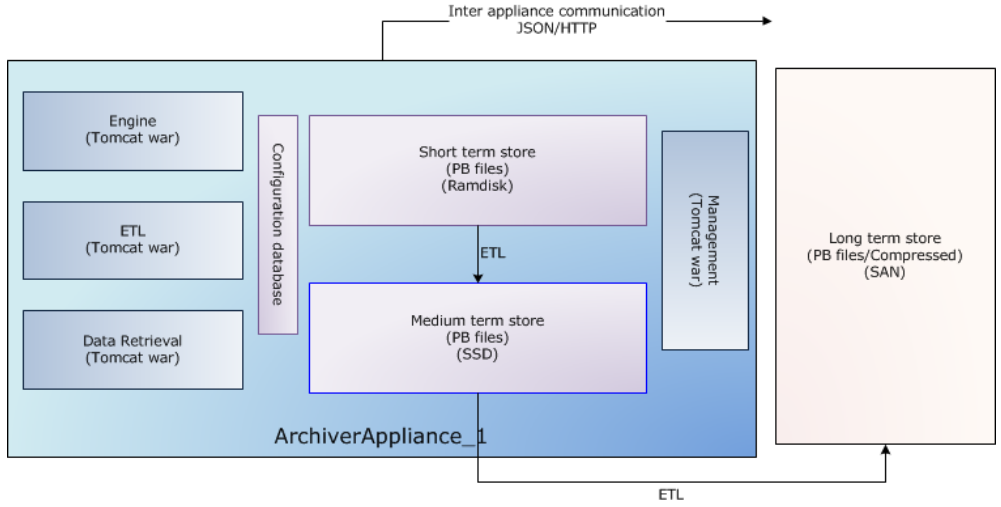
Practice:
Slow,
limited space

Choices…

	Channel Archiver	Relational Database	TimescaleDB	Archiver Applicance
Storage	Custom "data" and "index" files	MySQL, Postgres, Oracle	Postgres optimized for time series	Custom
Storage Speed	Fast	Slow	Better than Slow	Fast
Storage Maintainability	Eventually impossible	Trivial (for RDB admin)	Trivial (for RDB admin)	A few files per channel
Configuration	XML file	RDB with XML file import/export	RDB with XML file import/export	Web interface, XML file import/export
Readout	DataBrowser, custom C++ lib	DataBrowser, any RDB client	DataBrowser, any RDB client	DataBrowser, web interface, web service

Archiver Appliance Detail

- Storage: Files with "Protobuf"-encoded samples
  - One file per Channel and Stage
- Example Stages:
  - RAM disk for "today"
  - Solid-state disk for "this month"
  - NFS-mounted folder for "older"
- May create cluster of applicances



- Web interface to add channels, monitor performance, fetch data
- See https://slacmshankar.github.io/epicsarchiver_docs/index.html for "Quickstart" and more

Relational Database Detail

- Storage: MySQL, Postgres, Oracle
  - Ideal if you can leverage existing RDB cluster and admin support
  - RDBs have been reliable for decades
- Archive Engine, run as Linux service, writes samples to RDB
  - May run one for "Vacuum", one for "Cryogenics", one for "Beamlines" etc.
- Data accessible by Data Browser and pretty much any programming language

- See https://github.com/ControlSystemStudio/phoebus/blob/master/services/archive-engine/doc/index.rst

TimescaleDB Detail

- RDB Admins likely develop site-specific ways to “partition” the data and provide stored procedures for optimized data readout
- TimescaleDB = Postgres with extensions to automatically partition time series data

Instead of storing all samples into one RDB table like this:

All Samples

TimescaleDB allows storing the data in "chunks", for example creating one chunk per month:

January	February	March	...

Chunks may additionally split the data by channel IDs:

Ch 0-9999 Jan.	Ch 0-9999 Feb.	Ch 0-9999 Mar.	...
Ch 10000-19999 Jan.	Ch 10000-19999 Feb.	Ch 10000-19999 Mar.	...
Ch 20000-29999 Jan.	Ch 20000-29999 Feb.	Ch 20000-29999 Mar.	...
...	...	...	

- https://github.com/ControlSystemStudio/phoebus/blob/master/app/data browser-timescale/README.md

Decisions…

- Archiver appliance
  - By now a safe option, used by many in EPICS community
  - You'll need somebody to maintain the data
- RDB
  - Very dependable, ideal if you already have RDB admin support and want to access the data in various ways
- TimescaleDB
  - New, no operational experience
  - Promising option for new RDB setup, headstart over site-specific partinioning solutions

IOC Configuration

Archive engines subscribe to “archive” events (DBE_ARCHIVE)

```
camonitor -m l the_pv_name
```

For analog records, configure the ADEL field*

Unfortunately, not perfect for ‘log’ type values like vacuum pressures.

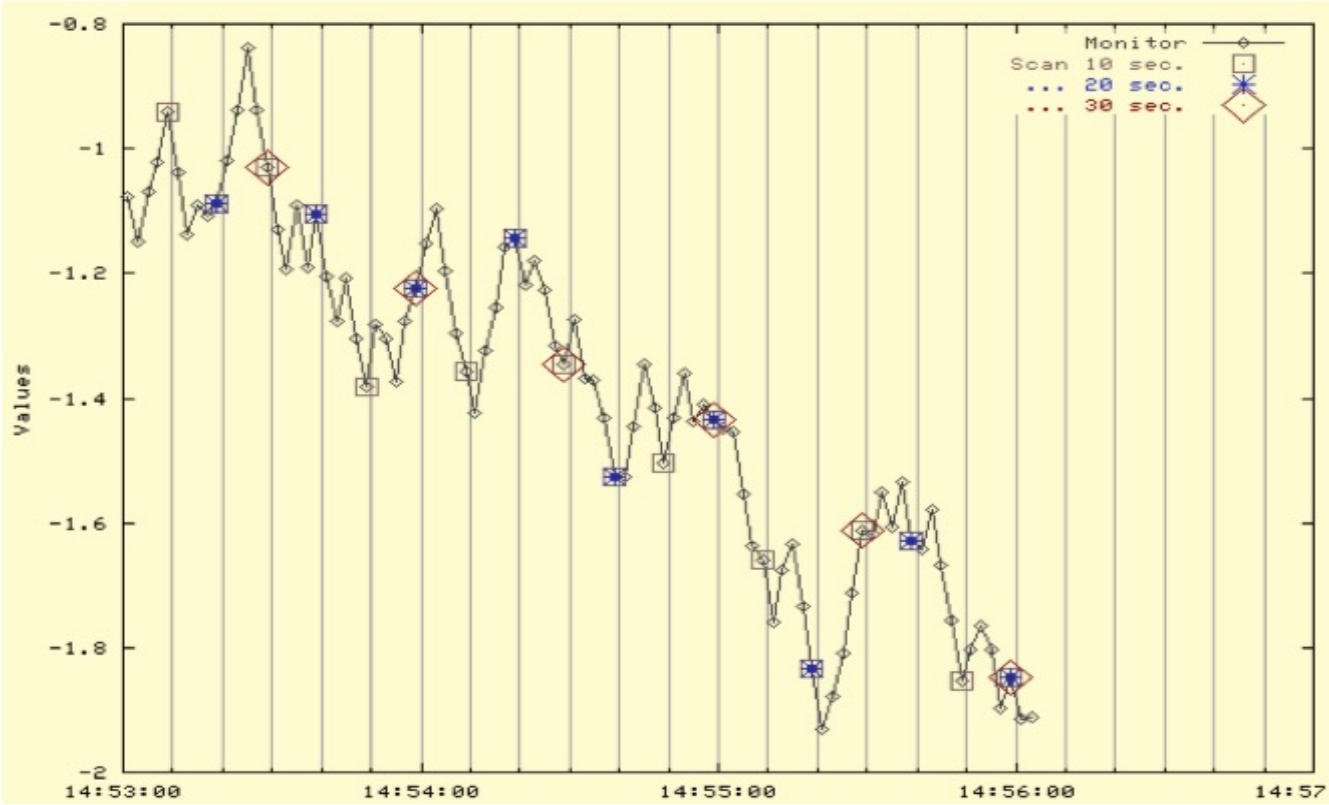
Assert that your IOC knows the time

Check time stamps reported by `camonitor`

* Also: EGU, PREC, ZNAM, ONAM, …
If it’s worth archiving, it should be properly configured.

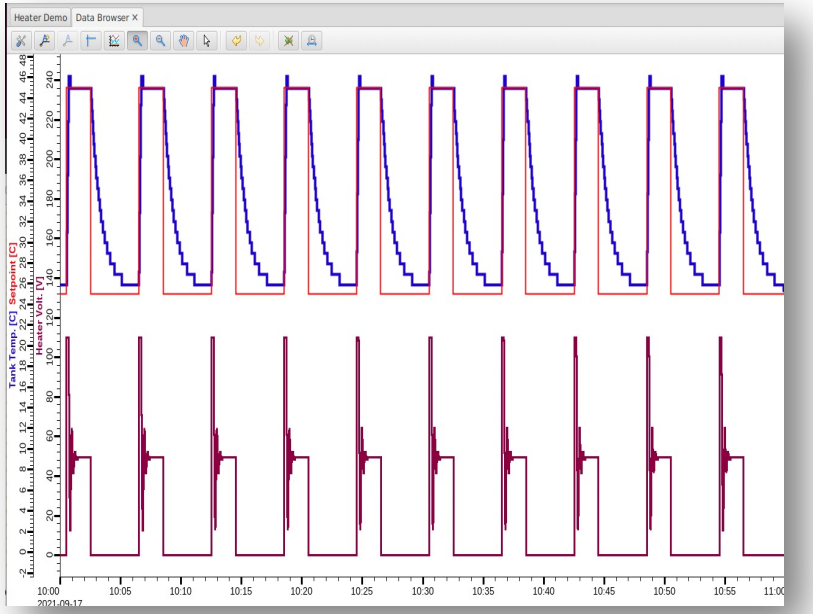
Archive Engine Options

- Monitor
  - Tries to save every received update
  - Based on ADEL
  - Might need an expected-update-rate to allocate buffers, skipping samples if there are too many
- Scan
  - Writes the most recent value every N seconds
  - Stores original time stamps



Viewing Archived Data

Open Data Browser, add PV, zoom/pan/set time range



Relies on Data Browser Preferences:

```
org.csstudio.trends.databrowser3/urls=jdbc:mysql://localhost/archive|RDB
org.csstudio.trends.databrowser3/archives=jdbc:mysql://localhost/archive|RDB
org.csstudio.trends.databrowser3/use_default_archives=true
org.phoebus.archive.reader.rdb/user=report
org.phoebus.archive.reader.rdb/password=$report
```

Archive

- Fundamentally simple: Store values of PVs
- There is no perfect implementation
  - Can't store everything at high rate forever
- Looking at data in Data Browser is easy

Examples specific to RDB in USPAS setup

Initial RDB Archive Installation

See https://github.com/ControlSystemStudio/phoebus/blob/master/services/archive-engine/doc/index.rst

1. Install MySQL or MariaDB, PostgreSQL, Oracle
2. Setup archive tables

Example archive for ‘fishtank’

Create configuration (based on existing one)

```
cd /ics/examples/19_archive/

archive-engine -help

archive-engine -list

archive-engine -engine Demo -export `pwd`/Demo.xml
```

… read, compare with fishtank.xml.

Import configuration and start sample engine:

```
archive-engine -engine fishtank -import `pwd`/fishtank.xml -replace_engine

archive-engine -engine fishtank
```

Check http://localhost:4812 in web browser

Fishtank Example

- Run IOC

```
cd /ics/examples/02_fishtank;
./st.cmd
```

- In CS-Studio, open `/ics/examples/02_fishtank/heater.bob` ➔ Current data
- Right-click on plot, "Open Data Browser"
  - Right-click to Show Toolbar
  - Change time range to 2021-09-17 10:00 … 2021-09-17 11:00 ➔ Archived data!