OAK RIDGE National Laboratory

TimescaleDB for Archived Data

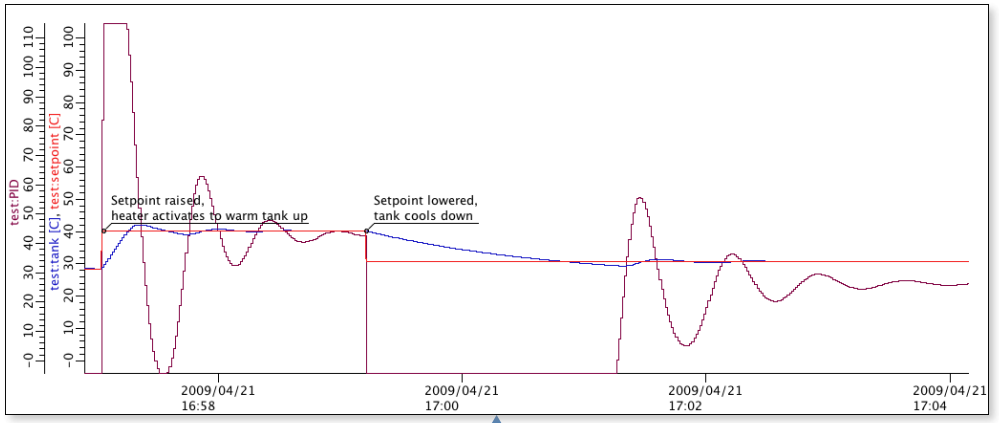
EPICS Collaboration Meeting

Slovenia, Sept. 2022

Kay Kasemir

U.S. DEPARTMENT OF ENERGY

Archive System



Configuration:

Which channels?
How to sample?

Sample Engine (Service)

Read PVs,
write to storage

CA, PVA

IOC

Storage

Vast and fast

✓Faster

Depends on implementation:

- Long-term track record?
- What queries are supported?
- Patch selected samples?

- Slower

Convenient:

✓Robust for decades

✓Arbitrary queries

✓Data management

✓Pre-aggregation for long-term reports

CS-Studio RDB Archive Engine

- ✓Supports Oracle, PostgreSQL, MySQL
- ✓Simple to start with one “sample” table

Channels 1-19999

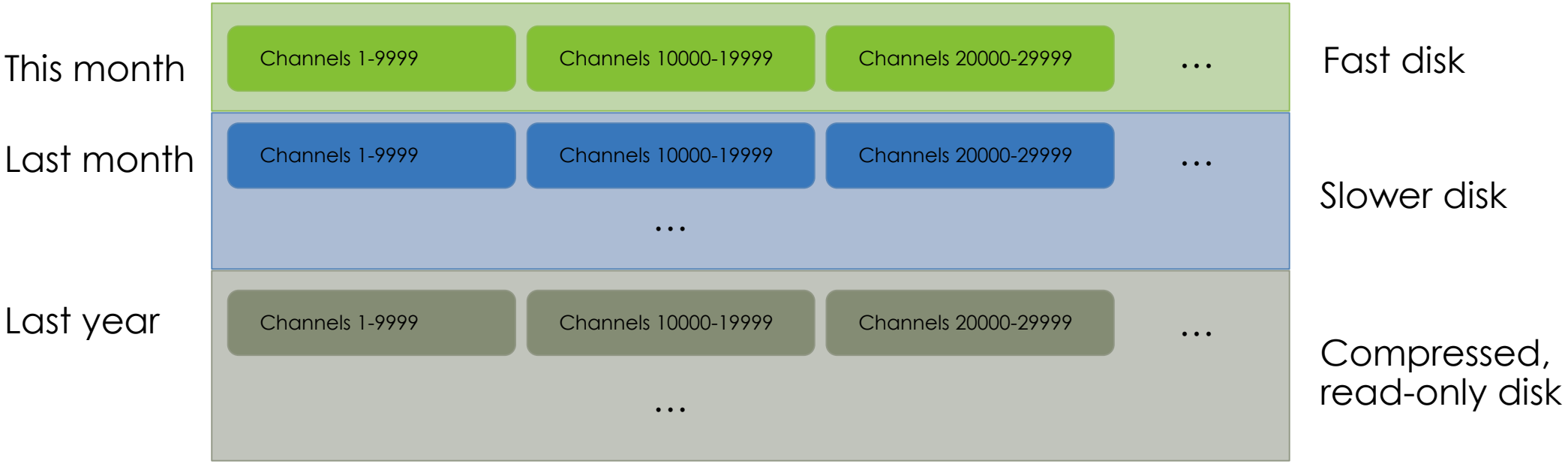
This month, last month, last year, …

Data maintenance looks easy, but operations like these can be slow, or fail to recover actual disk space:

- DELETE FROM sample WHERE channel_id = 42;
- DELETE FROM sample WHERE timestamp BETWEEN … AND …;

Oracle @ SNS, PostgreSQL @ ITER

Partitioned “sample” table



DIY set of scripts to automate partitioning and support optimized read-out

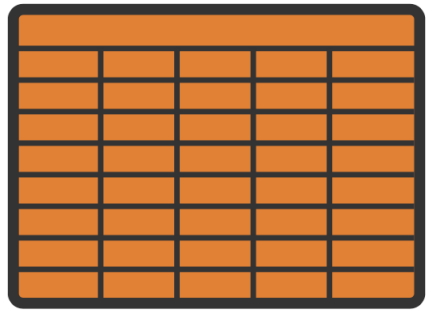
TimescaleDB

https://www.timescale.com

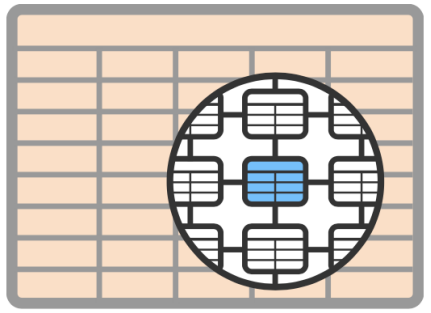
Free, open-source PostgreSQL extension

+ automated partitioning into ‘chunks’

+ additional time-based queries



Hypertable



Chunk

- ✓ Chunk by time and channel name/ID
- ✓ Automated indexing by time, fast access based on chunks
- ✓ ‘time_bucket’ query for retrieval of min/max/average
- ✓ Compression of older chunks
  - Size reduced to ~10%
  - Read-only
  - Re-arranged by channel for fast access to time slice for channel

How do I use that?

- RDB Archive Engine can write to TimescaleDB without any changes!
- CS-Studio Data Browser can read via stored procedure that takes advantage of TimescaleDB chunking
- See https://github.com/ControlSystemStudio/phoebus/tree/master/app/databrowser-timescale

TimescaleDB Archive

Allows RDB archive engine and Data Browser data retrieval to use TimescaleDB.

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.	...
...	...	...	

Refer to the following documents for details:

- 1 Install TimescaleDB
- 2 Configure Database
- 3 Archive Engine
- 4 Data Retrieval
- 5 TimescaleDB Details

Initial Experience

- ✓ Everything worked as described

- ✓ Hard to compare with SNS Oracle cluster (different networks, hardware and amount of available data), but performance seems at least as good as our custom partitioning setup

- ✓ TimescaleDB can be configured to 'chunk' by time range and channel ID
  - ❑ but unclear what the "best" configuration would be
  - ✓ Time range easy to adjusted at runtime for new data
  - ❑ Chunking by channel ID cannot easily be changed at runtime

- ✓ Compression works well, quite easy to automate

- ✓ Chunks can be placed in dedicated Table Space (disk)
  - ❑ But lacks Oracle BIGFILE approach where tablespace is a file of up to 128 TB which can be unlinked, backed up, restored, and linked back into database.

TimescaleDB

If you're about to start an RDB-archive, consider TimescaleDB instead of creating your own partitioning scheme

- ✓ RDB Archive Engine can write to it just like Oracle, MySQL, plain PostgreSQL
- ✓ CS-Studio Data Browser can benefit from stored procedure

Details: https://github.com/ControlSystemStudio/phoebus/tree/master/app/databrowser-timescale

What about InfluxDB for archived data?

Built for time series data

- Implemented in ‘go’
- Since 2013
- Started support in CS-Studio Archive Engine and Data Browser, but..
  - ‘NoSQL’, yet we do have fixed data types suitable for schema…
  - Issue with ‘long’ vs. ‘double’ distinction because of JSON serialization https://github.com/influxdata/influxdb-java/issues/276
  - Retention (aging) means different names for current vs. old data
  - Clustering is closed-source

Comparison with TimescaleDB:
https://blog.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877