

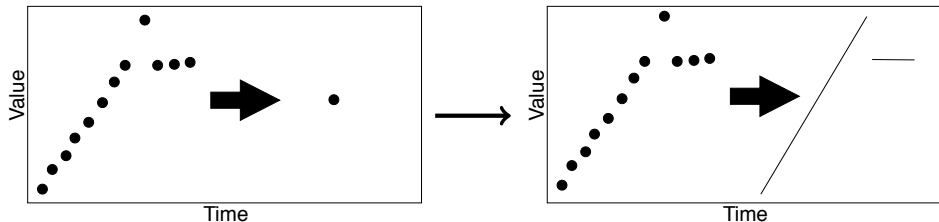
ModelarDB: Analytics of High-Frequency Time Series Across Edge, Cloud, and Client

Christian Thomsen, AAU
Søren Kejser Jensen, AAU

D3A — 2024-10-23

Wind Turbine Data Management

- ▶ Wind turbines have 100s of sensors that produce gigabytes of data each day
- ▶ Lossless compression like LZ4, zstd, and Gorilla does not compress enough!



- ▶ Thus, the time series are aggregated to reduce storage and bandwidth (left)
 - ▶ For example, 10 minutes averages, but important fluctuations and outliers are lost!
- ▶ ModelarDB represents time series with models instead of simple aggregates (right)
- ▶ In ModelarDB *models are any representation from which the values of time series can be reconstructed within a user-defined per value error bound (possibly zero)*

ModelarDB — Overview

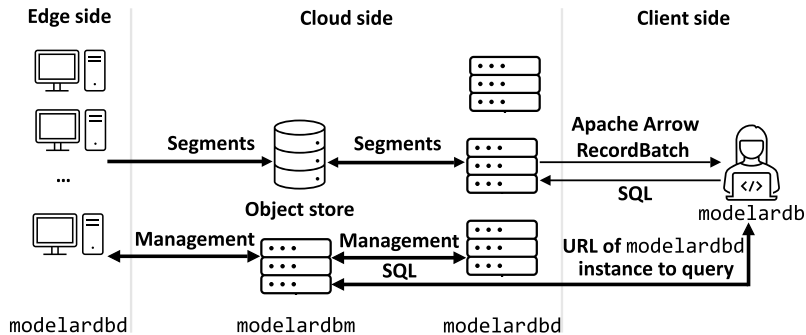


Figure: Deployment of ModelarDB across edge and cloud

- ▶ Open-source implementation in Rust using Apache Arrow-based components
- ▶ Server (`modelardbd`) on edge and in cloud with manager in cloud (`modelardbm`)
- ▶ Push-based ingestion of binary Apache Arrow RecordBatches (DataFrame-like)
- ▶ Time series are split by signal, compressed, and stored in Parquet in Delta Lake
- ▶ Queries data on local disk on the edge and data in the object store in the cloud

ModelarDB — Query Processing

- ▶ Supports normal *tables* for any relational data and *model tables* for time series
 - ▶ CREATE TABLE(wind_turbine_id VARCHAR, vendor VARCHAR, model VARCHAR)
 - ▶ CREATE MODEL TABLE(ts TIMESTAMP, wind FIELD(1), prod FIELD, id TAG)
- ▶ ModelarDB can execute arbitrary SQL queries on tables and on model tables
- ▶ Model tables add operators to query engine, e.g., to reconstruct data points
- ▶ Optimizer rewrites some aggregates to use models to reduce query processing time

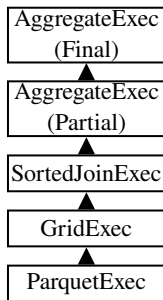


Figure: SUM computed from reconstructed data points

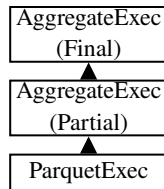


Figure: SUM computed directly from models

ModelarDB — Results

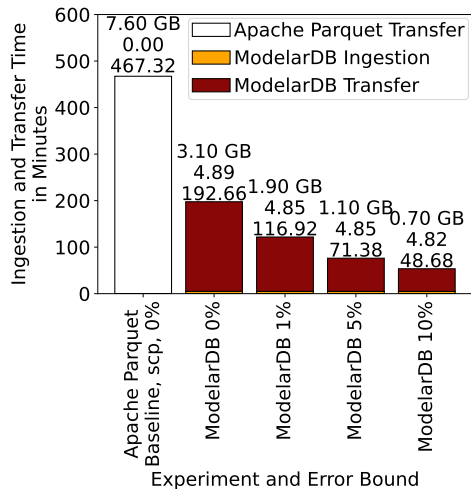


Figure: Size, ingestion, and transfer

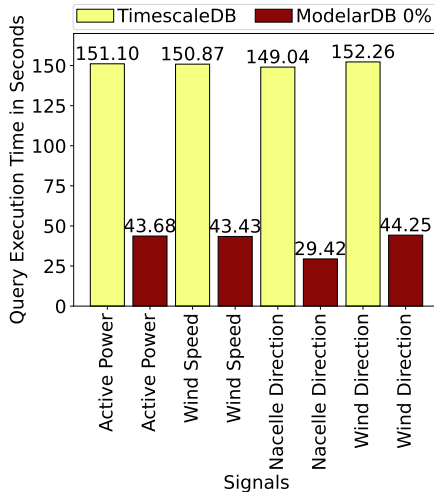


Figure: Aggregates queries

ModelarDB — Future Work

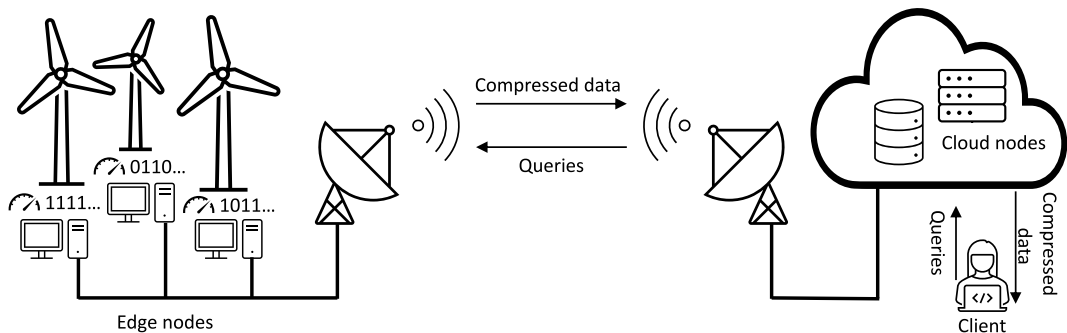


Figure: Efficient management of wind turbine data by integrating edge, cloud, and client

- ▶ Better performance with less hardware by integrating edge, cloud, and client
 - ▶ E.g., transfer compressed data if bandwidth is limited, but client's CPU is powerful
- ▶ Increase the types of operations that can be performed directly on the models
 - ▶ E.g., training and inference of machine learning models and optimize models for it

References

- ▶ S. K. Jensen et al., *Time Series Management Systems: A Survey*, TKDE 29(11), 2017
- ▶ S. K. Jensen et al. *ModelarDB: Modular Model-Based Time Series Management with Spark and Cassandra*, PVLDB 11(11), 2018
- ▶ S. K. Jensen et al., *Demonstration of ModelarDB: Model-Based Management of Dimensional Time Series*," SIGMOD, 2019
- ▶ S. K. Jensen et al., *Scalable Model-Based Management of Correlated Dimensional Time Series in ModelarDB+*, ICDE, 2021
- ▶ S. Tirupathi et al., *Machine Learning Platform for Extreme Scale Computing on Compressed IoT Data*, IEEE BigData, 2022
- ▶ S. K. Jensen et al. *ModelarDB: Integrated Model-Based Management of Time Series from Edge to Cloud*, TLDKS (53), 2023
- ▶ S. K. Jensen et al, *Holistic Analytics of Sensor Data from Renewable Energy Sources: A Vision Paper*, ADBIS (Short Papers), 2023
- ▶ C. E. Muniz-Cuza et al. *Evaluating the Impact of Error-Bounded Lossy Compression on Time Series Forecasting*, EDBT, 2024
- ▶ S. K. Jensen et al., *Time Series Management Systems: A 2022 Survey*, in Data Series Management and Analytics (forthcoming), T. Palpanas and K. Zoumpatianos (Eds). Preprint:
<https://vbn.aau.dk/da/publications/time-series-management-systems-a-2022-survey>
- ▶ A. Abduvakhobov et al. *Scalable Model-Based Management of Massive High Frequency Wind Turbine Data with ModelarDB*, PVLDB 17(13), 2025
- ▶ Official ModelarDB Legacy Repository. [Online]: <https://github.com/ModelarData/ModelarDB>
- ▶ Official ModelarDB Repository. [Online]: <https://github.com/ModelarData/ModelarDB-RS>