PNNL-33393

Evaluation of Use Cases and Types of Databases for Hosting Wind Power Data

September 2022

Matt Macduff
Ekman Kaur
Kefei Mo
Heng Wang
Chitra Sivaraman

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

Available to DOE and DOE contractors from
the Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062
www.osti.gov
ph: (865) 576-8401
fox: (865) 576-5728
email: reports@osti.gov

Available to the public from the National Technical Information Service
5301 Shawnee Rd., Alexandria, VA 22312
ph: (800) 553-NTIS (6847)
or (703) 605-6000
email: info@ntis.gov
Online ordering: http://www.ntis.gov

PNNL-33393

Evaluation of Use Cases and Types of Databases for Hosting Wind Power Data

September 2022

Matt Macduff
Ekman Kaur
Kefei Mo
Heng Wang
Chitra Sivaraman

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99354

Contents

1.0 Introduction

The goal of the National Wind Power Production Data Dashboard project is to develop a publicly available platform to model, process, and share wind power with uncertainty quantification for the current and future onshore and offshore wind plants across the continental United States. As of 2021, more than 70,000 utility-scale wind turbines have been installed across the United States and this number is expected to grow in the next few years. A large-scale wind power production database is needed for stakeholders (policy makers, operators, and researchers) to access easily for their decision-making. Currently, meeting that need is a challenge. Most wind power operators and system operators focus on regions of their own interest, and the data are usually inaccessible to the public.

This project aims to address this challenge; its objectives are:

1. Develop super-resolved, grid-cell meteorological and power datasets.
2. Develop a database of plant-level power time series.
3. Quantify plant-level power uncertainty and identify the uncertainty sources and driving factors.
4. Assess physical accuracy of super-resolved meteorological data, power data, and uncertainty quantification by comparison with observations (e.g., second Wind Forecast Improvement Project [WFIP2]) and historical actuals.
5. Integrate the developed datasets into existing U.S. Department of Energy (DOE) Wind Energy Technologies Office (WETO) datasets such as the National Renewable Energy Laboratory (NREL) Wind Integration National Dataset (WIND) toolkit and the DOE Atmosphere to Electrons (a2e) Wind Data Hub hosted on https://a2e.energy.gov.
6. Engage with industry stakeholders for data dissemination and feedback.

As part of objective 2 (develop a database of plant-level power time series data) and objective 5 (integrate the dataset with the Wind Data Hub hosted on https://a2e.energy.gov), the first task and milestone are to determine use cases and evaluate the database for optimal performance, efficiency, and cost to host 1 terabyte (TB) of information at different resolutions and different time scales. To build a performant database, a few factors need to be taken into consideration, namely identifying use cases and users, how information will be retrieved or downloaded, kinds of applications envisioned for the database, and how data will be normalized for efficiency.

1.1 Identifying Use Cases

As part of identifying use cases, the project team asked 10 stakeholders with different domain expertise to participate in a discussion. Eight of the 10 stakeholders agreed to meet with the team to discuss the current challenges, gaps, and their needs. Each of the stakeholders responded to the following topics:

- What is your stakeholder interest in this proposed database work?
- What do you want to achieve by using the work and what value does this bring to your work?
- What are your pain points and challenges?
- How would you interact with the database?

- What are your needs for data and formats?

The team categorized the answers as feature requests or capabilities and tabulated the impact of each feature request as seen in Table 1.

Table 1. Features gleaned from the discussion with stakeholders

Data Needs	Formats	Resolution	Queries	Time	Graphical Needs
Observed and model wind and power	CSV	Hourly data	Based on Interconnection, BA, geographical, lat/lon, polygon	Time is represented in UTC with an option to convert it to local time	Provide a map of transmission line, substation, bus stations, capacity and ownership
Historical data and forecasted data	netCDF	Coarsest resolution	Based on wind farm or wind turbine	Provide documentation on time conversion	Preview of data
Compare one location to another and compare one turbine to another for intercomparison		1-min data		Temporally align and spatially align across all datasets	Location of the wind turbines/wind farm and for future wind turbines/wind farm and their capacities.
Notification if data are changed or updated		15-min data		Instantaneous and nearest neighbor	
Data should be unitized*		Raw data		UTC and average to end	
Provide temperature and moisture data				Missing value should be NaN	
An API to connect to WIND Toolkit and query wind data for a particular location				Synchronize with WIND Toolkit	

API = application programming interface; BA = balancing authority; CSV = comma-separated values; NaN = not a number; netCDF = network common data form; WIND = Wind Integration National Dataset; UTC = Coordinated Universal Time

1.2 Stakeholder’s interest

Stakeholders included a private consultant, contractor to a utility company, employee of a utility or balancing authority, modeler, consultant, and researcher. The team tabulated and summarized each stakeholder’s pain points, challenges, and desires.

1.3 Current Pain Points and Challenges

Based on the team's synthesis of the discussion, there is a strong desire for a central platform to provide methods to filter and download all the stakeholder data needs in the right format and resolution.

1.4 User Requirements

The following user requirements were gleaned during the discussion with stakeholders.

1.4.1 Data Needs

Data needs included:

- Historical observed wind power, temperature, and moisture data
- Forecasted and modeled wind and wind power time-series data
- Access to NREL's WIND Toolkit wind data.

1.4.2 Data Formats

Users suggested downloading data in comma-separated values (CSV) and network common data form (netCDF) data formats. They also communicated the desire to preview data before downloading.

1.4.3 Data Queries

Users had several queries in mind that included filtering data by interconnections, balancing authorities, geographical locations (latitude/longitude or polygonal), or wind plants or type of wind turbines (i.e., technology types).

1.4.4 Representing Time in Data

Most users were familiar with Coordinated Universal Time (UTC) but requested options to convert to local time or provide documentation about time conversion when data are downloaded. Users also preferred that data were aligned temporally and spatially across datasets and aligned with WIND Toolkit. Guidance was also provided about averaging to the end of the interval. Missing values could be represented as not a number, or NaN.

1.4.5 Applications

Users communicated a strong need for a graphical user interface (GUI) that can be used to query the database. Methods to query based on technology using the GUI were suggested. Users requested a visual representation of not only the location of wind turbines and farms on a map, but also information about capacity and ownership of transmission lines and substations.

1.5 Determining Queries

By understanding the above requirements, each stakeholder's potential interaction with the database was formulated along with the reason for such an interaction.

1.5.1 Private Consultant/Contractor to Utility/Planner

To determine costs of new plants, this stakeholder requested hourly observed and modeled power and wind data along with the capacity of wind plants and ownership of the transmission lines, substation, and bus stations, and the distance to the transmission lines.

1.5.2 Modeler

This stakeholder requested hourly wind profiles based on a polygon (interconnection, balancing authority [BA] footprint) be provided, the time be aligned with the WIND Toolkit, and notification be provided when data are changed or updated.

1.5.3 Western Electricity Coordinating Council (WECC) Employee

This stakeholder requested information about load, resources, and location of the wind turbines and wind plants so location of future wind turbines/wind farms and their capacities can be determined.

1.5.4 Consultant Planner

This stakeholder requested that the project provide raw, 15-minute and 1-hour historical and forecasted data for a particular location using a shape file (latitude and longitude) that could be filtered by long- or mid-range wind profile for different design or technologies. This information could be used to compare the cost of one location to another.

1.5.5 Electric Reliability Council of Texas (ERCOT) Employee

To understand the ramping supply when load varies, this stakeholder requested information be provided as 1-hour power data in CSV format with timestamps represented in UTC, averaged to end of the hour with documentation on converting to local time.

1.5.6 Researcher

This stakeholder requested information on the observed power curve per turbine or per farm in the coarsest resolution for all years in netCDF so wind speed and power curve can be modeled.

1.5.7 Professor

This stakeholder requested that one year of processed, validated power data in local time for a particular location selected via a GUI be provided, so students can use these data for class projects.

1.6 Data Availability and Data Sources

The team built a SQLite database with five static tables to assimilate data from various sources to assess the kinds of data that can be acquired publicly and to understand the complexities of the data.

The team extracted data from the following sources and populated static tables:

- https://www.eia.gov/electricity/data/eia860/
- https://eerscmap.usgs.gov/uswtdb/viewer/#3/37.25/-96.25
- https://en.wind-turbine-models.com/turbines
- https://sam.nrel.gov/download.html

The team used the Plant, Generator, and Wind tables from EIA-860 to extract data about plants and generators that have wind as energy source. These formed the plants and generators tables in the SQLite database.

Turbine geographic information system (GIS) data were extracted from the U.S. Geological Survey (USGS) website to put in the eerscmap table.

Turbine model information and power curve data were extracted from the Wind turbines database and the SAM database to build the models and power curve tables.

2.0 Evaluating Database for Optimal Performance, Efficiency and Cost

The consideration of a database for storing the large amount of data is based on some assumptions of the details. The focus of this evaluation is for the storage of the data records, which may be 1-minute, 15-minute or hourly data across many locations. The data to be stored for each time and location may vary, but will include power, wind speed, and other meteorological values. Other information about the locations can be stored in a separate relational database like MySQL.

For this evaluation, the schema can be expressed as a single table with values for each time and location. At least initially, the database will be updated infrequently. Thus, the primary factor is the query performance. Using a dataset size of 1 TB covering a span of 20 years gives a reference for evaluating cost.

While alternate hosting platforms could be considered for implementing this database, Amazon Web Services (AWS) provides a large selection of database services along with explicit pricing to enable some comparison. The current Wind Data Hub platform and framework is also currently hosted on AWS, which would make the integration of this dashboard much easier and cost-efficient. Thus, for this evaluation we focus on the options available through AWS. Although a wide variety of databases exist, the team compared five databases for use for these data in Table 2.

In comparing cost, a key question is how many queries would be performed and how much data would be read. This required some interpretation of how the queries would be formed and how the underlying database would be configured. In general, queries will be bounded by time and location, so not all of the data would be scanned. Further, the queries would often be for hourly data (as opposed to 1-minute data). The query and access pattern will be intermittent, based on the individual researchers' needs. As an extreme case, the entire database is scanned through the course of a month, which would result in the equivalent of 1 TB of reads for each database option. These assumptions are used in considering cost in Table 2.

While it is likely that any of the databases could be made to work, two of the options stand out based on their features and low cost for further evaluation: Timestream and Athena.

Table 2. Comparison of the Databases – AWS

	Timestream	Redshift	Dynamo	Athena	MySQL
Benefits	• Serverless with auto-scaling • Quickly analyze time series data using SQL	• Concurrent scaling • High-performance query processing	• Serverless • Key-value No SQL database	• Serverless • Runs standard SQL • Executes queries in parallel	• On-demand scalability • Good for general purpose OLTP database
Limitations		Does not support result cache for cross-database queries	• Querying data is extremely limited • Table joins are impossible		• Table maximum size is 16 TB
Additional Information			Good for high read/write rate, auto-sharding, auto-scaling and high durability	Primarily a query engine	
Price Format	Pay per transaction	Always on cluster	On-demand capacity mode and provisioned capacity mode	Pay per transaction	Always on instances
Pricing	• (Writes) 1 million write of 1 KB size - $0.50 • (Queries) Per GB scanned - $0.01 • (Memory Store) price per GB stored per hour - $0.036 • (Magnetic store) Price per GB stored per month - $0.03	On-Demand: ra3-xlplus - $1.08 per hour	https://aws.amazon.com/dynamodb/pricing/ • On-demand (standard): First 25 GB stored per month is free and $0.25 per GB • Read Request Units (RRU):$0.25 per million RRU	• $5.00 per TB of data scanned • Plus AWS bucket cost - First 50 TB / Month	• T4.xlarge = $0.258/hour • General purpose SSD storage = $0.115 per GB-month
Plug-in Numbers	• For queries, $10 for 1 TB scanned • Magnetic store for 1 TB = $30 • Memory of 1 GB is $26	$800/month if the ra3-xlplus node is to run 24*7 with 1 TB of data	$244 for storing 1 TB of data + $0.25 for read request (given we read the entire database about four times)	$5 per TB queried+ $23 storage = $28	t4g.xlarge and storage = roughly $300 per month
Analysis	• Serverless database for time series data known for high performance with adaptive query engine • Multiple tables for different time resolutions	Complex pricing system, generally best to use for real-time analytics and combining multiple data sources	• High price for large amount of data as it is often used for low-scale operations because of its simplicity • Likely multiple tables for different time resolutions	• Reasonable pricing and executes queries in parallel resulting in good performance • Requires a design of schema and shards	Good for general purpose OLTP database but may not be efficient for our use case, but works directly with existing structure

AWS = Amazon Web Services; OLTP = online transaction processing; SQL = Structured Query Language; SSD = solid state drive

2.1 Timestream

Timestream automatically scales up or down to adjust to capacity and performance as it offers virtually infinite scale. Table schema for this database are dynamically created based on the attributes of the incoming (time series) data, which allows flexible and incremental schema definition. When data are stored, Timestream partitions those data based on time and attributes, which accelerates data access. For storage, both memory storage and magnetic stores are used. The memory store is designed for high-throughput data writes, and quick point-in-time queries. The magnetic store is for lower-throughput, late-arrival data writes; long-term data storage; and fast analytical queries. Timestream also enables configuration of memory and magnetic storage. Data are queried using SQL statement(s). For performance optimization, scheduled queries can help by precomputing some fleet-wide aggregate statistics. For cost optimization, using multi-measure records, setting data retention for memory/magnetic store, and batching multiple events per write can help with cost efficiency. The limitation for this database is that maximum data size for a query result allowed is 5 GB, and the maximum number of measures per multi-measure record is 256. Timestream supports a flat model and time series models for queries. Common query patterns are last values queries (asset tracking, latest location, latest sensor reading), n-event correlation (looking for patterns in events), aggregates, derivatives, and rate of change.

2.2 Athena

Athena is serverless, which automatically scales based on datasets or number of users. Athena uses a managed data catalog to store information and schemas about the databases and tables in Amazon S3. It also uses schema-on-read technology, which requires no data loading or transformation, and tables definitions/schema can be deleted without affecting the underlying stored data. Data are queried using SQL statement(s). Athena carries out queries in parallel on extremely large datasets within seconds. It can support a large number of queries, making it a read-heavy efficient database. Query optimization can be done by using ORDER BY, JOINs, GROUP BY and approximation functions. For performance and cost optimization, it is recommended to partition data and compress and split files. This database is limited in that it cannot: build custom user-defined functions (UDFs), write back to S3, or schedule and automate jobs. Athena also requires a separate bucket to log results.

3.0 Conclusion

For the purpose of this work, the Timestream database appears to be very well suited. It provides automatic optimization around time, the primary facet of these data. Timestream should provide excellent performance and future scalability with minimal labor. The estimated price is an excellent value. However, without running specific tests, it is difficult to make a final recommendation. Other options may need to be considered in practice. While Athena appears to be slightly cheaper, it is not as clear how much labor would be required to optimize its structure for best cost/performance.

In addition to the general use of case of database queries, the project team must take into consideration the relatively common cost of requests for very large sets of data. While the selected database may handle many of these requests, all of the databases considered have different limits that would affect very large queries. To accommodate these queries, it is recommended that they be run as batch jobs, or as some type of database dump, so that the user actually just downloads these as files and not as a typical database query response.

Pacific Northwest National Laboratory

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99354

1-888-375-PNNL (7665)

www.pnnl.gov