TECH NOTES

ci-compass.org

Evaluation of Scalable Solutions for Time Series Database Streaming

Valerio Pascucci, Giorgio Scorzelli, Steve Petruzza, Erik Scott, Jarek Nabrzyski, Anirban Mandal, Jameson Rollins, Patrick Godwin, Jonathan Hanks, Martin Beroiz

Date Published: June 10, 2025
DOI: 10.5281/zenodo.11087445

Data Distribution and Data Streaming for Event Databases

Fast access to time databases (databases specialized for time series data) is an important component of many national research facilities and large-scale experiments. They allow continuous information collection about the phenomena to be recorded by the detectors of the facility. This information needs to be analyzed in real-time to trigger the immediate execution of workflows associated with key events of interest. In addition, more detailed analyses are typically driven by the selection of longer, more complex sequences of events retrieved from a historical database to reveal hidden patterns and create derived quantities for further inspection.

It is therefore critical for the U.S. National Science Foundation (NSF) Major Facilities' (MF) cyberinfrastructure (CI) to incorporate efficient, scalable services that allow storing large databases of time events, let the related scientific communities query them in real-time, and augment them with derived quantities of interest. In this TechNote, we document a study and a possible solution that is both efficient and scalable.

Time Databases

As a part of a study conducted by NSF CI Compass during its engagement with the NSF Laser Interferometer Gravitational-Wave Observatory (LIGO), we evaluated a number of time databases to assess their performance in terms of execution time for different queries (light and heavy) as well as responsiveness of the database from scratch (loading or updating data times). In particular, we focused on circumstances where the database recording the events should be accessed immediately, which requires fast updates of the table with the new information provided. Moreover, we wanted to guarantee high scalability over time to make sure that growing with a large number of historical records would not reduce its performance.

With this goal in mind, we considered a number of solutions [1] and compared their performance and scalability to assess possible solutions with good overall characteristics. The resulting analysis allowed for focusing the evaluation activities on two communication protocols (Kafka [2] and Arrow Flight [3]) for data transfer and three databases (ClickHouse [4], InfluxDB, and TimescaleDB) for storing the data and allowing fast queries. The practical use cases of reference are based on the experiences developed at the NSF IceCube Neutrino Observatory (IceCube) [5], the A Toroidal LHC ApparatuS (ATLAS) experiment at the Large Hadron Collider at the Conseil Européen pour la Recherche Nucléaire (CERN) [6], and the LIGO Next Generation Data Delivery system [7].

Use Case Evaluation

The reference use case is based on the LIGO environment [7], as illustrated in Figure 1

This project is supported by the National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.

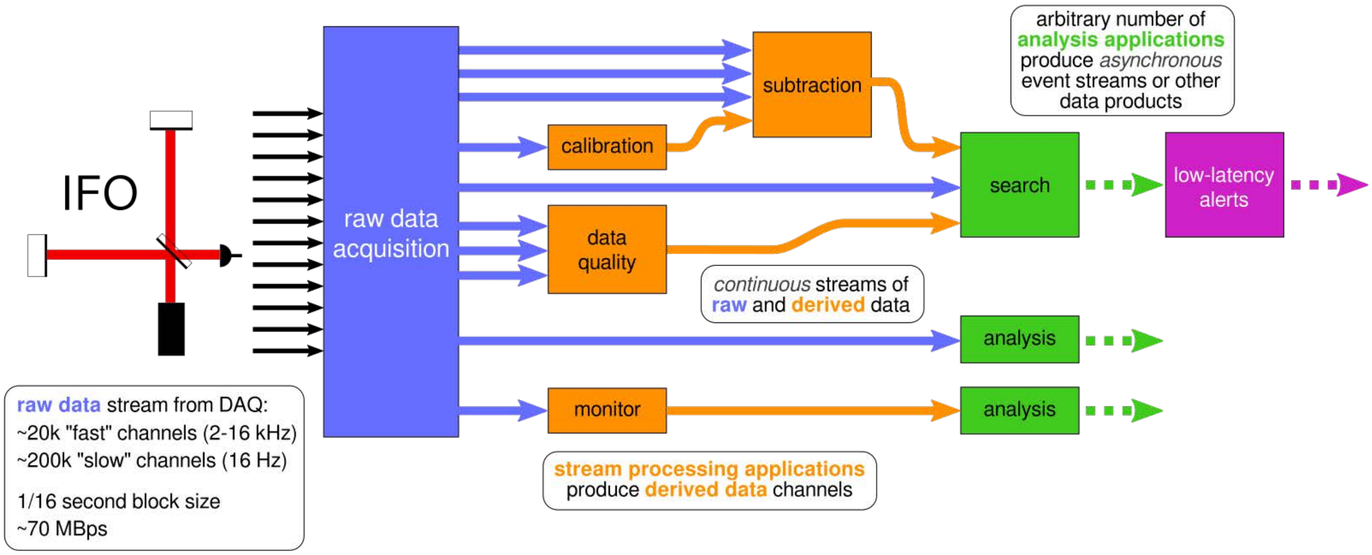


Figure 1. High level diagram of the Laser Interferometer Gravitational-Wave Observatory's Next Generation Data Delivery (NGDD) system. Original diagram available at [7].

below, which shows the stream processing/distribution framework and associated applications that process and access the raw and derived data streams from their interferometers (IFO). The stream processing applications perform various tasks like calibration, subtraction, and data quality checks. The framework needs to support hundreds of data stream producers and thousands of data stream consumers. The objective is to unify the low-latency stream processing, data distribution, and access to online and offline channels into a single integrated system.

For the choice of communication platform, both Kafka and Arrow Flight have been tested and deemed appropriate since they both provide the necessary set of communication primitives and performance characteristics. Kafka is considered somewhat more preferable because of its simplicity of use and overall flexibility. Generally, if a system already has a communication layer based on either technology, though, we would recommend to continue using it since switching may be cost-prohibitive to the research facilities.

Regarding the database technologies, the differences are more substantial. We summarize here the key findings from the detailed performance provided in the blog post [8], which involves a variety of tests, including queries with heavy and light workloads, database startup time, and average data rates (data size on memory vs. disk). The source code for the benchmarks (https://github.com/timescale/tsbs) was used for verification of the results. The following figures (taken from [8]) show the key performance characteristics of interest. In particular, the load time for a relatively small database is shown in Figure 2. Note that ClickHouse is optimized for larger datasets, with a better-than-average responsiveness gap at startup is expected to be even better for larger datasets.

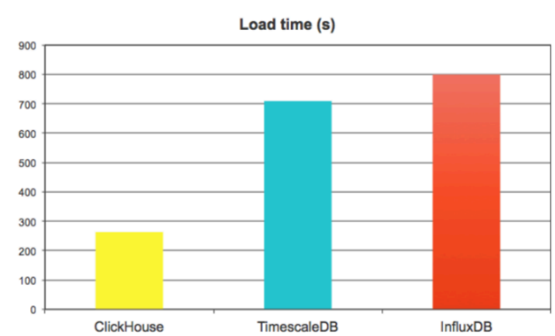


Figure 2 Database load time for a 10K row dataset.

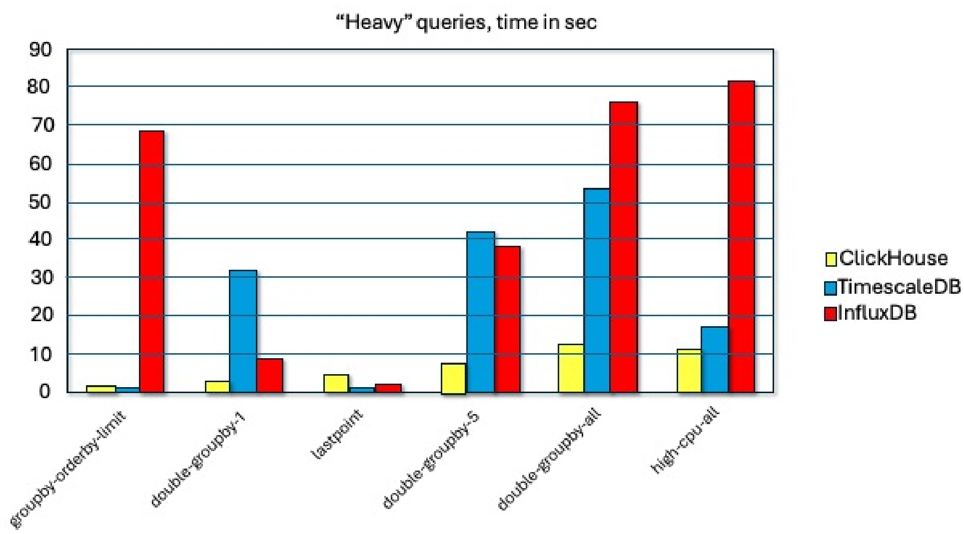
The most important parameter is the time execution of heavy (order of seconds execution) and light (order of milliseconds execution) queries of the database, which are reported in Figure 3.

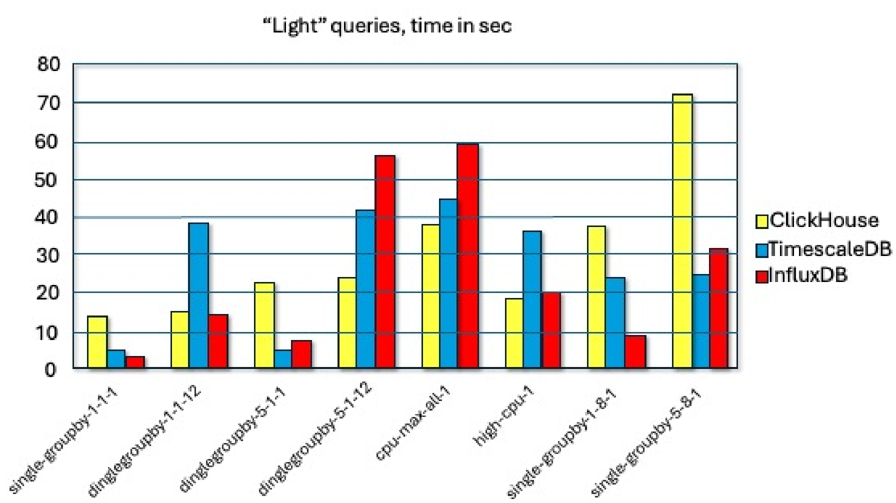
As discussed before, ClickHouse is optimized for heavier loads and, therefore, shows better scalability when the data grows in size. In particular, light queries are executed mostly in the same range as for the alternative databases, while heavier queries show a clear

This project is supported by the U.S. National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.

and consistent advantage, which makes this the approach of choice.

The results provided in the Atlas project assessment reported in Figure 4 confirm the performance assessment.





Comparison of different databases (ClickHouse, TimescaleDB, InfluxDB) in the Time Series domain. The "Time Series Benchmark Suite" (https://github.com/timescale/tsbs) was used to test performance. Queries were run type by type, 1000 times in 8 parallel workers. Results are divided into two groups for easier presentation: "heavy inquiries" (top) that take single-digit seconds or more; "light Queries" (bottom) that typically take milliseconds to complete. Results show that for the fastest millisecond queries, ClickHouse is generally behind competitors. But as soon as the query becomes more complicated (using, for example, aggregators functions or sliding windows), ClickHouse outperforms TimescaleDB and InfluxDB.

This project is supported by the U.S. National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.

Additional hands-on experience

We conducted a direct empirical evaluation of the ClickHouse solution, focusing on data ingestion for the following practical use cases of community data commons.

- Material Commons https://materialscommons.org/
- Amazon Web Services (AWS) Open Data https://aws.amazon.com/it/opendata/
- Digital Rocks Portal https://www.digitalrocksportal.org/
- Globus Material Data Facility https://www.materialsdatafacility.org/
- Texas Advanced Computing Center (TACC) Ranch https://tacc.utexas.edu/systems/ranch/

We scraped scientific repositories to gather dataset records, then transformed them into a set of Comma-Separated Values (CSV) files. We then loaded these records into our ClickHouse database, which was running on an AWS EC2 cloud instance. Table 1 shows how long it took to inject data into our ClickHouse database; the "time" column includes (i) CSV download time from Wasabi object storage to the cloud machine and (ii) ClickHouse ingestion time. Numbers show how quickly this procedure could performing in a real-world scenario: e.g., it took 25 seconds to process 129K files, from the 5TB Material Commons repository [9]); 291 seconds to handle 49M records from the 10 PB AWS Open Data repository [10]; 115 seconds to inject 3M files from the TACC repository[11].

The overall performance has been impressive, with a local injection of 339 million records stored in a CSV table completed in under 3 minutes on a single AWS EC2 instance. To the best of our knowledge, performance of this level can only be achieved on GPU-based deployments.

This approach handles different dataset sizes efficiently, making it ideal for managing large-scale scientific data catalogs.

Table 1: TIme to inject data into ClickHouse

Repository	Number of files	total size	time
Materials Commons 2.0	129K	5TB	25s
Open Data on AWS	49M	10PB	291s
Digital Rocks	8k	3TB	42s
globus labs	1M	5TB	159s
TACC	3M	34PB	115s

In our final assessment, we also considered the maturity and size of each solution's user base. The two plots in Figure 5 display the GitHub star history (from https://star-history.com/) for both components of the solution that were tested. GitHub star history is a tool that provides a graphical representation of a repository's popularity over time. This shows how all the tools are quite mature, with ClickHouse and Kafka having larger communities.

This project is supported by the U.S. National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.

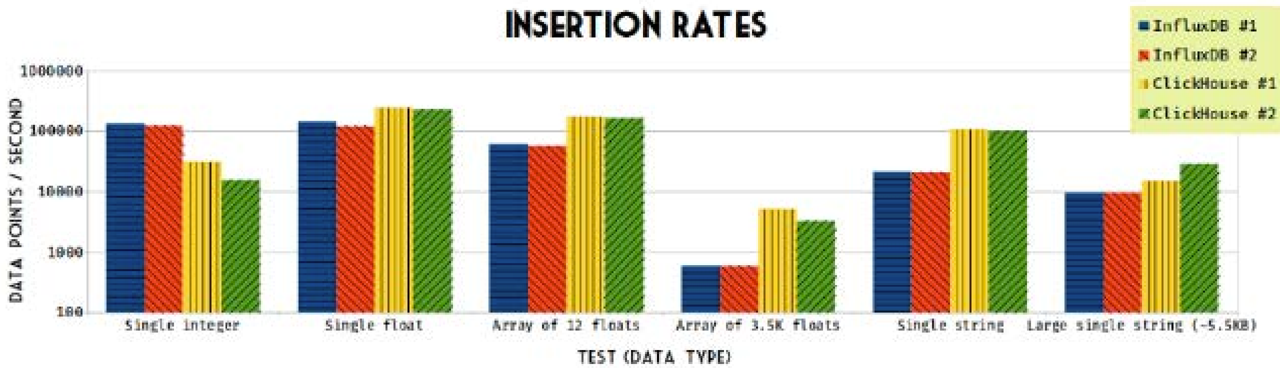


Figure 4. Performance assessment from [6].

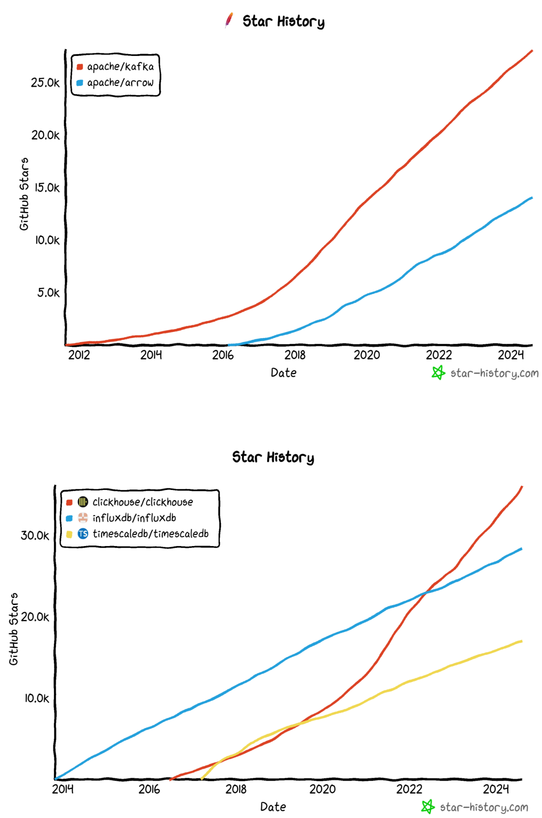


Figure 5: GitHub star history for the solutions tested. (top) Comparison among ClickHouse, InfluxDB, and TimescaleDB. (bottom) Comparison between Kafka and Arrow Flight.

Conclusions

Overall, the assessment has resulted in a design featuring a core database capability built on ClickHouse and a Kafka communication layer. These two frameworks provide the performance and scalability required in a typical deployment at a Major Facility supported by the U.S. National Science Foundation. This includes the robustness and stability required in a production environment that supports daily operations.

This project is supported by the U.S. National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.

Bibliography

[1] V. Rudakov, M. Timur, and A. Yedilkhan, "Comparison of Time Series Databases," in *2023 17th International Conference on Electronics Computer and Computation (ICECCO)*, IEEE, Jun. 2023, pp. 1–4.

[2] G. Shapira, T. Palino, R. Sivaram, and K. Petty, *Kafka: The Definitive Guide : Real-time Data and Stream Processing at Scale*. O'Reilly, 2021.

[3] T. Ahmad, Z. Al Ars, and H. Peter Hofstee, "Benchmarking Apache Arrow Flight -- A wire-speed protocol for data transfer, querying and microservices," *arXiv [cs.DC]*, Apr. 06, 2022. Available: http://arxiv.org/abs/2204.03032

[4] A. Wickramasekara, M. P. P. Liyanage, and U. Kumarasinghe, "A comparative study between the capabilities of MySQL and ClickHouse in low-performance Linux environment," in *2020 20th International Conference on Advances in ICT for Emerging Regions (ICTer)*, IEEE, Nov. 2020, pp. 276–277.

[5] M. G. Aartsen *et al.*, "The IceCube Neutrino Observatory: instrumentation and online systems," *J. Instrum.*, vol. 12, no. 03, p. P03012, Mar. 2017.

[6] M.-E. Vasile, G. Avolio, and I. Soloviev, "Evaluating InfluxDB and ClickHouse database technologies for improvements of the ATLAS operational monitoring data archiving," *J. Phys. Conf. Ser.*, vol. 1525, no. 1, p. 012027, Apr. 2020.

[7] J. Rollins, "Next Generation Data Delivery, Wiki Page," *GitLab*. Available: https://git.ligo.org/groups/ngdd/-/wikis/home. [Accessed: May 03, 2024]

[8] "ClickHouse crushing time series," *Altinity | The trusted partner for ClickHouse®*, Nov. 15, 2018. Available: https://altinity.com/blog/clickhouse-for-time-series. [Accessed: May 03, 2024]

[9] G. Tarcea *et al.*, "The materials commons data repository," in *2022 IEEE 18th International Conference on e-Science (e-Science)*, IEEE, Oct. 2022. doi: 10.1109/escience55777.2022.00060. Available: https://ieeexplore.ieee.org/document/9973700/

[10] "Registry of Open Data on AWS." Available: https://registry.opendata.aws/. [Accessed: May 08, 2025]

[11] "TACC Research," Oct. 27, 2022. Available: http://tacc.utexas.edu/research/tacc-research/. [Accessed: May 08, 2025]

Acknowledgments

We acknowledge the valuable input from all the staff members and leadership of the Laser Interferometer Gravitational-Wave Observatory, which created the impetus for this evaluation of the data streaming and time databases technologies.

For more information, please visit ci-compass.org or email us at contact@ci-compass.org.

This project is supported by the U.S. National Science Foundation Office of Advanced Cyberinfrastructure in the Directorate for Computer Information Science under Grant #2127548.