ibaHD-Server Benchmark

Comparison between ibaHD-Server and time-based database systems

White Paper

Issue 1.0

Measurement Systems for Industry and Energy

www.iba-ag.com

Manufacturer

iba AG

Koenigswarterstr. 44

90762 Fuerth

Germany

Contacts

Main office	+49 911 97282-0
Fax	+49 911 97282-33
Support	+49 911 97282-14
Engineering	+49 911 97282-13
E-mail	iba@iba-ag.com
Web	www.iba-ag.com

Author

Fabian Mielke

Issue	Date	Author	Revision
1.0	02-2021	FM	First issue

Content

1 Introduction

There is a wide variety of database systems on the market with special architectures for different tasks. Some database systems are specialized in storage management and management of time-related data. They are generally characterized by the fact that timestamps are used as primary keys of the data tables.

iba AG has carried out a comparison between ibaHD-Server and the two time-based database systems InfluxDB and TimescaleDB within the scope of an internal project. These two systems are well-known representatives of time series databases [1]. The objective was a transparent comparison of the available features and performance. The results and especially the main differences between the systems are presented in this white paper.

Benchmarks are always subject to certain boundary conditions and are therefore difficult to compare with each other. Especially if the compared systems differ in their ideal application scenario. In order to perform an objective benchmark, these scenarios must be taken into account. This white paper represents a summary of the results of the benchmark.

2 Comparison of ibaHD-Server with InfluxDB and TimescaleDB

The three database systems are based on different technical approaches. InfluxDB is a schema-free time series database, i. e. no table schema has to be created in advance Instead, InfluxDB groups records based on *tags* that can be added to the record. Multiple records that have exactly the same combination of tags belong to a time series. This approach is also known as the tagset model. The data is sent to InfluxDB using the Line Protocol as a plain text format. In contrast, TimescaleDB follows a hybrid concept of relational and time-based databases [2, 3]. It can be installed as an extension to PostgreSQL. The time series in TimescaleDB are almost indistinguishable from normal relational tables. Basically they are abstract hypertables that store time series as so-called *chunks*. Relational and time series data can be combined using TimescaleDB.

In ibaHD-Server, a different strategy is pursued, which can store measurement data much faster and more efficiently compared to InfluxDB and TimescaleDB. The efficiency is achieved by a fast and lossless encoding algorithm. Storage efficiency has the highest priority especially for high-frequency data acquisition. ibaHD-Server offers a seamless integration of iba data into the iba system and does not depend on powerful hardware.

2.1 Technical background

ibaHD-Server in comparison with InfluxDB and TimescaleDB.

Interfaces	ibaHD-Server	InfluxDB	TimescaleDB
Write	ibaPDA ibaDatCoordinator ibaHD-Server import	HTTP API UDP (JSON) Client libraries	ADO.NET JDBC Native C Library ODBC Streaming API
Read access	ibaAnalyzer ibaDatCoordinator ibaDaVIS gRPC-API	HTTP API UDP (JSON) Client libraries (Wrapper)	ADO.NET JDBC Native C Library ODBC Streaming API
Query Language	-	InfluxQL FluxQL	SQL

2.2 Orientation

The intention of TimescaleDB and InfluxDB is different from ibaHD-Server. An essential component of the two systems is the retention policy, which is kind of a clean-up strategy. For example, it can be specified that obsolete data is to be deleted or that downsampling is to be applied. One of TimescaleDB's arguments for the performed *downsampling* is that the relevance of older data decreases over time because it is no longer updated [4]. However, the information content in detail is also lost with downsampling.

ibaHD-Server follows a different strategy in this respect. The measurement data is compressed immediately and lossless compressed in order to save measurement data also for documentation and analysis purposes. Nevertheless, a limitation of the used storage space is configurable. Aggregation is also performed, so that long-term analyses can be carried out if required which do not relay on the original data acquisition resolution. In this way, an overview of the recorded data can be obtained. The raw data can then be queried for detailed analysis.

	ibaHD-Server	**InfluxDB**	**TimescaleDB**
Strategy	High writing speed Secure transmission through *layered storage* concept with intermediate storage	Keeping current data sets in RAM Storing previous data sets on the disk	Holding current time data chunks in RAM Storing previous data sets on the disk
Aggregation	Aggregation levels in addition to the original data Retention of the original resolution in terms of documentation *Retention policy* with storage limitation	*Retention policy* deleting old data *Downsampling* [5]	*Retention policy* deleting old data Continuous aggregation policy (*Downsampling*) [4]
Resources	Automatic management of computer resources Optional limitation of the maximum amount of managed data	Custom configuration of resource consumption possible [6]	Custom configuration of resource consumption possible [7]
Time series related metadata	Metadata is contained entirely in the measurement data segments	No relational concept for metadata [8]	Relational linking between time series and metadata possible [9]

	ibaHD-Server	**InfluxDB**	**TimescaleDB**
Non-equidistant data	Only equidistant time series data *Event Store* supports non-equidistant data	Non-equidistant time series are natively supported	Non-equidistant time series are natively supported
Scalability	Limited	Scalable on a local cluster with *Influx Enterprise* *InfluxDB Cloud* as a service on *Amazon Web Services*, *Google Cloud Platform* and *Microsoft Azure* [10]	Scalable cluster as a service on *Amazon Web Services*, *Google Cloud Platform* and *Microsoft Azure* [11]
Further features	Management and storage of events in a separate *Event Store* Demo clients available for ibaHD-Server API with gRPC (Python, C++, C#, Java)	*Telegraph* agent for collecting and reporting metrics Client libraries available	*PostgreSQL* functionalities compatible with TimescaleDB
Operating system	Windows	Linux OS X Windows (no official support)	Linux OS X Windows

2.3 Performance

Besides the features, an important part of the benchmark was the examination of the write and read performance. For this purpose, data was generated on a dedicated computer, which corresponded to typical application in data acquisition and measurement environments. The database systems were tested with the support of specially developed software. The test software generates data sets that are transferred to a client software. This client software unified the data and prepares it for the database interface. The data records are then inserted. In this way, a unified initial condition is created.

In advance, all three systems were optimally configured. The unit *metrics per second* was chosen as the comparative metric for read and write speed. The structure of such a *metric* is based on typical measurement data, i. e. it consists of a time stamp, a signal name and a floating point number symbolizing the measured value. The measured time also includes the necessary conversions for the interface of the respective systems. This is e.g. encoding for ibaHD-Server, the insertion into SQL commands for TimescaleDB and formatting into the Line Protocol for InfluxDB.

A desktop computer with an Intel Xeon E-2146G 3.50GHz processor with 6 cores and 16 GB RAM was used as the test computer. The versions of the database systems in details: ibaHD-Server 2.5.0; InfluxDB 1.8.3; TimescaleDB 2.0 with PostgreSQL 12 64-bit. All systems were tested on Windows 10.

In both write and read speed, ibaHD-Server outperforms the other two time series databases by far. The performance of InfluxDB and TimescaleDB is reduced by the interface and the internal database mechanisms while the data can be encoded continuously with ibaHD-Server. The following results are average values of several test series.

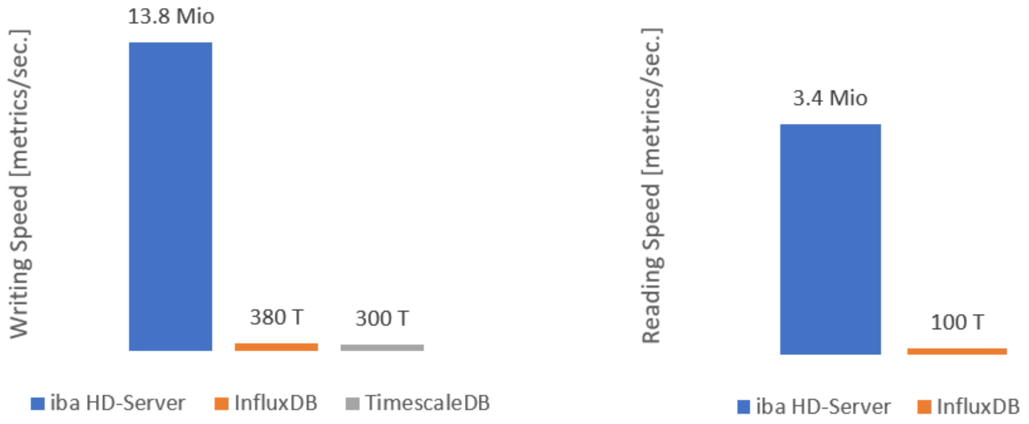

Fig. 1: Mean values of write and read speed. The results for TimescaleDB in the right part were unfortunately no longer available.

The execution of comparable benchmarks is a complex procedure that is subject to many influencing parameters. In order to exclude errors and mistakes, the results have been validated with other benchmarks and publications [12, 13, 14, 15]. In these publications, similar metrics were determined for InfluxDB and TimescaleDB.

3 Application scenarios

All tested data systems including the ibaHD-Server are best suited for certain applications, respectively. The following table summarizes these scenarios.

ibaHD-Server	InfluxDB	TimescaleDB
High speed recording of measurement data Lossless data acquisition Long-term availability of process and production data for later analysis and product documentation Integration into the iba system *Real-time* visualization with minimal delay	Monitoring applications Industrial Internet of Things applications	Applications requiring relational structures Applications with high time series cardinalities [16]

4 Future developments

4.1 Scalability

ibaHD-Server is suitable to manage a vast amount of data of the same structure. In measurement technology it is common to organize analog and digital signals in a tree structure with modules. The assignment between the signals of the input sources must be carried out once before the running process of the software and is saved as configuration. A high variance of the data structure, e. g. by changing signal configurations, reduces the performance strongly. Solutions, in which a large number of clients simultaneously send data to ibaHD-Server for storage, also pose a great challenge for the infrastructure. In the future, iba AG will also offer a scalable concept, e. g. for IoT applications.

4.2 Availability

The availability of the measurement data can also be increased with a scalable ibaHD-Server system as cluster- or cloud-capable service. For example, if one *data node* fails, other instances can return the requested data. This capability is achieved by replication. In addition to redundancy, such a system can also be used to distribute the load of read and write accesses between the nodes.

4.3 Performance optimization

ibaHD-Server already achieves high data speeds on simple desktop computers and works very efficiently due to the ongoing encoding. Nevertheless, iba AG continuously improves the systems.

5 Glossary

Aggregation	Compression of information, e. g. by averaging or *downsampling* methods.
Chunk	A defined set of data records or information that is part of the total data set.
Cluster	Networking of computers, e. g. to guarantee the reliability of a system or to increase the overall computing power of a system.
Data node	Part of a network of several computers (*cluster*) that stores subset of the total data.
Downsampling	Systematic reduction of data sets, e. g. by deleting every tenth data set.
Real-time capability	Ability of a system to respond within a specified time period for sure. Real-time capability indicates nothing about the performance or speed of a system.
Metrics	Structure and information defining a data set. For example, a metric consists of a timestamp, a *tag* and a number.
Relational database	Database concept that contains tables that are linked to each other. The relations of the columns are created by so-called keys.
Tags (InfluxDB)	Additional information about a data set, which can also be used for filtering. Tags also determine the relation of a single data set to a time series.
Time series	Series of data sets, each of which is assigned to specific points in time. In a time series, each point in time is unique.
Time series database	Database concept and subset of relational databases with optimized functions for time-related data. The timestamp of a data set always represents a unique primary key within a time series.

6 References

[1] DB-Engines. Information about relational and NoSQL database management systems.
URL: https://db-engines.com/de/ranking/time+series+dbms. Last accessed 03.02.2021.

[2] TimescaleDB Blog. TimescaleDB vs. InfluxDB: Purpose built differently for time-series data. 2020.s
URL: https://blog.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/

[3] InfluxDB. Key Concepts.
URL: https://docs.influxdata.com/influxdb/v1.8/concepts/key_concepts/ . Last accesssed 08.02.2021.

[4] TimescaleDB Documentation. Data retention.
URL: https://docs.timescale.com/latest/using-timescaledb/data-retention. Last accessed 03.02.2021.

[5] InfluxDB. Downsampling and data retention.
URL: https://docs.influxdata.com/influxdb/v1.7/guides/downsampling_and_retention/. Last accessed 03.02.2021.

[6] InfluxDB. Configure InfluxDB OSS.
URL: https://docs.influxdata.com/influxdb/v1.8/administration/config/. Last accessed 03.02.2021.

[7] TimescaleDB. Configuring TimescaleDB.
URL: https://docs.timescale.com/latest/getting-started/configuring. Last accessed 03.02.2021.

[8] InfluxDB Documentation. InfluxDB schema design and data layout.
URL: https://docs.influxdata.com/influxdb/v1.8/concepts/schema_and_data_layout/. Last accessed 03.02.2021.

[9] TimescaleDB Documentation. Data model.
URL: https://docs.timescale.com/latest/introduction/data-model. Last accessed 03.02.2021.

[10] InfluxDB. Cloud.
URL: https://www.influxdata.com/products/influxdb-cloud/. Last accessed 08.02.2021

[11] TimescaleDB. Cloud.
URL: https://www.timescale.com/cloud. Last accessed 08.02.2021

[12] InfluxDB. OSS guidelines.
URL: https://docs.influxdata.com/influxdb/v1.8/guides/hardware_sizing/. Last accessed 08.02.2021

[13] TimescaleDB Blog. Insert Performance.
URL: https://blog.timescale.com/blog/timescaledb-vs-influxdb-for-time-series-data-timescale-influx-sql-nosql-36489299877/ Last accessed 08.02.2021

[14] Brussels University. Time Series Databases and InfluxDB. 2017.
URL: https://cs.ulb.ac.be/public/_media/teaching/influxdb_2017.pdf. Last accessed 08.02.2021.

[15] Ghent University. Towards a representative benchmark for time series databases. 2019.
URL: https://lib.ugent.be/fulltxt/RUG01/002/786/062/RUG01-002786062_2019_0001_AC.pdf

[16] InfluxData Blog. Use Cases.
URL: https://www.influxdata.com/blog/category/usecase/. Last accessed 08.02.2021

7 Support and contact

Support

Phone: +49 911 97282-14

Fax: +49 911 97282-33

Email: support@iba-ag.com

Note

If you require support, indicate the serial number (iba-S/N) of the product or the license number.

Contact

Headquarters

iba AG
Koenigswarterstrasse 44
90762 Fuerth
Germany

Phone: +49 911 97282-0

Fax: +49 911 97282-33

Email: iba@iba-ag.com

Mailing address

iba AG
Postbox 1828
D-90708 Fuerth, Germany

Delivery address

iba AG
Gebhardtstrasse 10
90762 Fuerth, Germany

Regional and Worldwide

For contact data of your regional iba office or representative
please refer to our web site

www.iba-ag.com.