ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP036

RESEARCH ON HALF HISTORICAL DATA ARCHIVER TECHNOLOGY

Xiaokang Sun*, DaDi Zhang, Huhang Chen
University of Science and Technology of China, HeFei, China

Abstract

The Hefei Advanced Light Facility (HALF) is a 2.2GeV fourth-generation synchrotron radiation light source. The HALF control system is a distributed control system based on Experimental Physics and Industrial Control System (EPICS). As the essential part of the HALF control system, the Historical Data Archiving System (HDAS) is responsible to store operational data for the entire facility including the accelerator and beamlines, and provides the functions for data query and analysis. According to the estimation based on the HALF scale, approximately 25,000 EPICS PVs will be stored in HDAS, and these accumulated massive data require a dedicated database for persistent storage and management. Under the EPICS PV data scenario of HALF, a fair database test platform is designed and built to test the read-write performance of databases commonly used in the particle accelerator field. The tested objects include EPICS Archiver Appliance and the five databases MongoDB, HBase, InfluxDB, TimescaleDB, and Cassandra.The test results indicate that TimescaleDB has the fastest read performance, and 1.4×10^6 items of data can be read per second. In the future, a TimescaleDB distributed cluster will be designed and deployed, and an HDAS prototype system will be developed based on this cluster.

INTRODUCTION

The Hefei Advanced Light Facility (HALF) is a 2.2GeV fourth-generation synchrotron radiation light source, which is scheduled to start construction in Hefei, China in 2023. The HALF consists of an injector, a 480 meters diffraction-limited storage ring, and ten beamlines for phase I [1, 2]. The HALF control system is a distributed control system based on Experimental Physics and Industrial Control System (EPICS). As the essential part of the HALF control system, the Historical Data Archiving System (HDAS) is responsible to store operational data for the entire facility including the accelerator and beamlines, and provides the functions for data query and analysis.

The HDAS archives the EPICS PV data from the control system. PV is a structured kind of data, which includes PV ID, PV value, timestamp, severity, and other metadata. According to the estimation based on the HALF scale, the HALF control system will generate approximately 250,000 PVs in total.According to the some accelerator project cases [3–5], in order to simplify the research metrics of archived PVs, the archived PVs percentage of the HALF control system is chosen as 10%. Approximately 25,000 PVs will be stored in HDAS, which will generate tens of TB of data per year. These massive amounts of data generally require a dedicated database for persistent storage and management. The architecture of the HDAS is shown in Fig. 1, the data archiving engine is responsible for collecting PV from the IOCs through the CA/PVA protocol, and storing the data into the database cluster. Based on the database cluster, the Web applications for data querying and analysis are developed. The database performance directly influences the speed of data processing and analysis. Therefore, it is necessary to choose a high-performance database for HDAS.

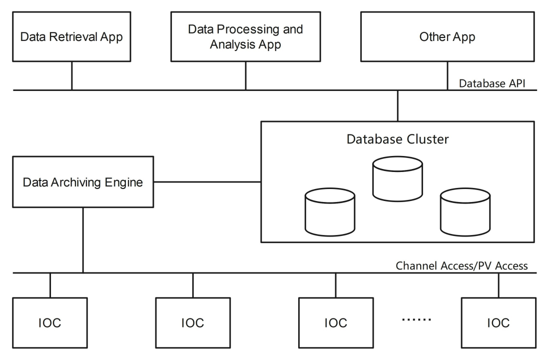

Figure 1: Architecture of the HDAS.

In order to choose a high-performance database for HDAS, in the EPICS data archiving scenario, an environment-fair test platform is designed and built to test the read-write performance of databases commonly used in the accelerator field. Additionally, research on the database software ecosystem is conducted for the development of database applications.

Different databases are adopted by the particle accelerator facilities around the world, the popular and commonly used databases are as follows: MongoDB, HBase, InfluxDB, TimescaleDB, and Cassandra. Besides, the EPICS Archiver Appliance (AA) as a dedicated EPICS data archiving tool is widely used in many particle accelerator facilities [3]. The AA is developed by a collaboration of SLAC, BNL, and MSU in 2015. The AA stores data in the form of Google Protocol Buffer files, and the multiple stages mechanism is adopted to achieve high data retrieval performance. The Beijing Electron Positron Collider II (BEPC-II) proposed a data archiving system based on MongoDB [6]. MongoDB is a database written in C++, based on distributed document storage [7]. The Hefei Light Source (HLS-II), the China ADS Front-end Demo Linac (CAFe), and the Japanese Proton Accelerator Research Complex (J-PARC) all use the HBase database for storing historical operation data [8,9]. HBase is a column-oriented non-relational database that runs on top of the Hadoop Distributed File System (HDFS) [10]. The accel-

* sunxk@ustc.edu.cn

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP036

erator facilities at ISIS and the Los Alamos National Laboratory have developed a historical data archiving system based on InfluxDB [11, 12]. InfluxDB is a time-series database developed using the Go language [13]. At the EPICS Collaboration Meeting in September 2022, Kay Kasemir proposed an archiving solution based on TimescaleDB, which is directly compatible with the RDB Archive Engine [14]. The RDB Archive Engine is an EPICS official archive engine that supports relational databases. TimescaleDB is an open-source time-series database based on PostgreSQL, which supports standard SQL queries [15]. S. Marsching and his team developed a Cassandra-based archiving system, the Cassandra Archiver, for storing EPICS PVs [16].Cassandra is a distributed non-relational database. It achieves high availability by setting replication factors and cross-cluster replication strategies [17].

In summary, the five databases—MongoDB, HBase, InfluxDB, TimescaleDB, and Cassandra—have use cases in the particle accelerator field. The above five databases cover the relational, non-relational, document-based, and time-series database types. They not only have horizontal expansion capabilities, but also have mechanisms to ensure high availability. However, there is currently a lack of complete testing and comparison of these databases in the EPICS data archiving scenario. Therefore, the comprehensive test is carried out for the above databases. The test content includes read-write performance tests and a comparison of database software ecosystem. It should be noted that AA is not a database, but as a commonly used EPICS archiving tool, it also is tested as a comparison. These research results can provide a reference for HALF database selection.

EXPERIMENT DESIGN AND DEVELOPMENT

In the scenario of massive EPICS PV data, an experiment is designed to conduct multi-dimensional read-write performance tests on the aforementioned five databases and AA. For HDAS databases, the data is usually written in batches every few tens of seconds, and the data will not be altered or deleted after writing. Data queries are generally executed multiple times, and may involve long-term range data. In response to the above-mentioned scenario, the corresponding read-write performance metrics have been designed. The read performance metric is the volume of data queried per second, denoted as Queried Data Per Second (QDPS). The higher QDPS means less data query latency, which indicates that the database read performance is higher. Similarly, the write performance metric is the volume of data successfully inserted per second, denoted as Inserted Data Per Second (IDPS). The higher IDPS means higher data insertion efficiency.

Hardware Platform Design

As shown in Fig. 2, the experimental platform is designed and built using virtual machines (VM). The experimental platform consists of eight VMs, all deployed within the same local area network. An IOC application runs on the VM7, which is responsible for generating PV data. The VM1 to VM6 all have the same hardware configuration, and run the AA and the following databases respectively: MongoDB, HBase, InfluxDB, TimescaleDB, and Cassandra. The VM8 runs a data archiving engine developed based on Python, which is responsible for data collection and writing into databases automatically. Additionally, this VM also runs the performance testing tool Apache JMeter. The specific hardware configurations and software version information for the VMs are presented in Table 1. The CPU model used by all VMs is Intel Xeon Silver 4216, the storage is ME4084 disk array, and the operating system is Debian 11 (Bullseye).

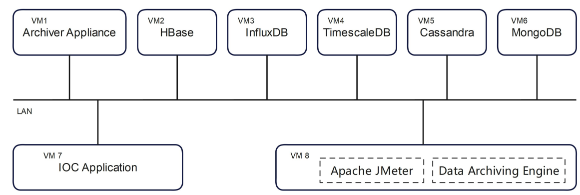

Figure 2: Database performance testing experimental platform.

Table 1: The Virtual Machines' Configuration

VM NO.	Software	Memory (GB)	Disk (GB)
VM1	Archiver Appliance 1.1.0	8	500
VM2	MongoDB 5.0.9	8	500
VM3	HBase 2.4.2	8	500
VM4	InfluxDB 2.2.0	8	500
VM5	TimescaleDB 2.7.0	8	500
VM6	Cassandra 4.0.4	8	500
VM7	EPICS 7.0.6 IOC Application	8	50
VM8	Apache Jmeter 5.5 Data Archiving Engine	8	500

Software Development

IOC Application As mentioned in the Introduction section, there are about 25,000 PVs stored in HDAS. Due to the limited hardware resources of the experimental environment, the IOC Application is designed based on 25,000 PVs which reduces the number of PVs by about 10 times. There are a total of 2,000 PVs, and each PV is updated at a rate of 10Hz to generate about 4 days of data, with a total of about 7 billion items of PV data. This data volume is enough to test the performance of each database.

Database Design There are significant differences in the data models of various databases. To ensure the relative

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP036

fairness of the test, the design of the database data models observes the following principles:

- The database adopts the default or officially recommended configuration.
- Ensure the consistency of information stored in the database. Based on the PV information collected by the RDB Archive Engine, the fields that need to be stored include a nanosecond-level timestamp, pv id, value, severity id, and status id. The meanings of these fields are presented in Table 2.
- To maximize read performance, the indexes or primary keys of the database tables use the combination of PV name and timestamp. The partition parameter of the database tables uses the data volume of the same time span.
- Each database adopts the fastest data compression mechanism respectively.

Table 2: The Field Description of PV Data

Field	Data type	Field meaning
pv_id	String	Unique identification of PV.
timestamp	Timestamp	Second portion.
nanos	Integer	Nanosecond portion.
value	Integer	PV value.
severity_id	Integer	PV severity.
status_id	Integer	PV status.

Performance Testing Tool The Apache JMeter application is an open-source Java application designed to load test functional behavior and measure performance. The Jmeter can support the read-write test of relational databases and various NoSQL databases such as HBase and MongoDB. By utilizing the data interfaces of different databases and customizing the corresponding load, the performance test of different types of databases can be achieved. Furthermore, there are many examples of database performance tests using Apache JMeter.

In our case, the specific usage is to import the official drivers of each database into JMeter and write the corresponding Java program to test the read-write performance. The PerfMon Metrics Collector plugin is used to monitor the consumption of CPU and memory resources on the VMs during the testing process.

EXPERIMENTS AND RESULT ANALYSIS

According to different test parameters of the data volume, the test can be divided into the following four schemes:

- Reading Test: Test QDPS with different data volumes under a single thread.
- Writing Test: Test IDPS with different data volumes under a single thread.

To eliminate the test accidental errors, the reading test conducts 500 experiments under the same parameters. Each experiment randomly selects PV and time range, the test results take the average QDPS of 500 experiments. Before each writing test, a state restoration operation is performed on the related VMs to ensure that the database is in initial state before each test. The test details are described below.

Reading Test

This test simulates a scenario where a single client queries data over a period, which is the most basic and common data reading scenario in HDAS. The data volume generated by a single PV within the time range of 0.4 hours to 10 hours is designed to do this test, the corresponding data volume ranges from 1.44×10^4 items to 3.6×10^5 items. Figure 3 (a) and Fig. 3 (b) respectively represent the test results of the read performance for the latest data and historical data. As the volume of query data increases, whether it is the latest data or historical data, the QDPS of each database presents an initial increase and subsequent stable trend. It can be concluded that when the query data volume reaches 1.0×10^5 items, the QDPS of each database gradually tends to be stable.

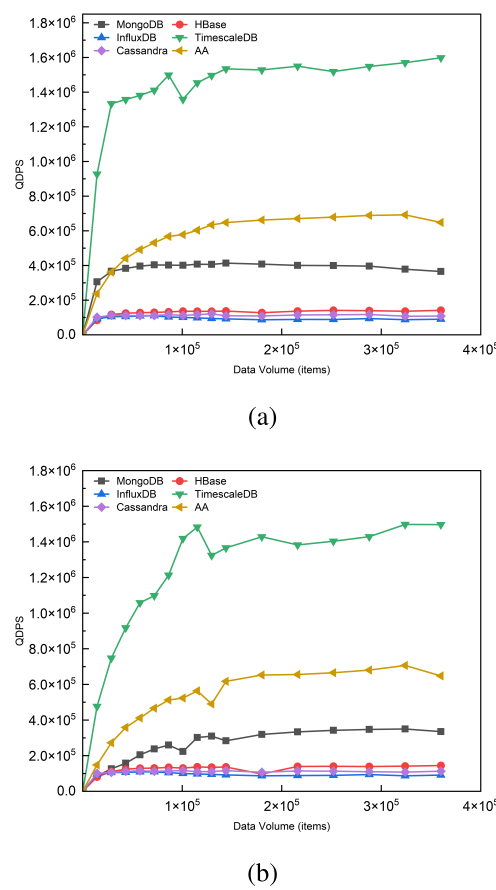

Figure 3: Test results of Reading Test. Figure (a) shows the test results of querying the latest data, and Figure (b) shows the test results of querying historical data.

The QDPS of Cassandra, HBase, and InfluxDB has almost no difference, the QDPS of historical data and latest data are all stable at around 1.0×10^5 items. The AA and MongoDB

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP036

have better read performance, the QDPS of AA is stable at 6.0×10^5, and the QDPS of MongoDB is stable between 3.5×10^5 and 4.0×10^5. The highest QDPS is achieved by TimescaleDB, which can reach 1.4×10^6.

Writing Test

This test simulates a scenario where the data archiving engine writes different data volumes to the database in a single thread. The test uses the accumulated data volume of 2000 PVs in the range of 0.1 minutes to 1 minute for batch insertion. The results are shown in Fig. 4. When the written data volume exceeds 4.8×10^5 items, the IDPS of InfluxDB can reach 1.5×10^5, which shows that it has the highest insertion speed. TimescaleDB also has good write performance, and its IDPS can be stabilized at 1.2×10^5.

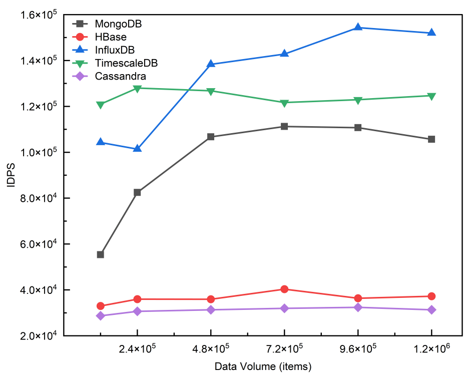

Figure 4: Test results of Writing Test. The IDPS results for each database under different data volumes.

Summary of the Test

When the single-threaded is used for data query, TimescaleDB always has the fastest performance. Its QDPS can reach more than 1.4×10^6, which exceeds the QDPS limit of other databases. Therefore, this read performance test conclusion is also applicable to the accumulated data volum over a longer period. InfluxDB and TimescaleDB have superior performance in single-threaded data writing. Considering the HDAS is required to store 25,000 PVs, and each PV updates at an average frequency of 1Hz. Therefore, databases with IDPS greater than 25,000 can theoretically meet the data writing requirements of HDAS. According to the results of the Writing Test, each database in a single node can meet the writing requirements of HDAS. Considering HDAS's "once-write, multiple-reads" scenario, QDPS is a more important parameter than IDPS. Therefore, TimescaleDB emerges as the most suitable database for HDAS.

The reason why TimescaleDB has excellent read performance is as follows. TimescaleDB is an extension of PostgreSQL with storage optimized for time-series data, supporting partitioning and adding time-aware retrieval methods to improve performance over plain RDB methods. TimescaleDB primarily optimizes the storage of time-series data through the following internal mechanisms:

- TimescaleDB utilizes hypertables to store and manage time-series data. Hypertables enable data to be partitioned into chunks based on both time and space dimensions. This allows queries to access only the relevant chunks of data instead of scanning the entire dataset, reducing the number of IO operations and improving query speed.
- TimescaleDB incorporates an efficient compression mechanism for hypertable data, reducing chunk sizes by over 90%. This not only saves storage space but also enhances read performance.
- TimescaleDB provides an automatic aggregation function, which can form a new materialized view according to the aggregation strategy, which can further improve the query speed..

CONCLUSION

The database is a core component of the HDAS, directly determining the query speed of massive data volume. To select a suitable HDAS database, an environment-fair test platform is designed and built under the HDAS data scenario to test the read-write performance of AA and the five databases, which are MongoDB, HBase, InfluxDB, TimescaleDB, and Cassandra. Test results indicate that under various data volumes, TimescaleDB always has the fastest read performance, and its QDPS upper limit can reach more than 1.4×10^6. The write performance of TimescaleDB is also excellent, and its IDPS upper limit can reach more than 1.2×10^5, which can fully meet the data writing requirements of HDAS. Considering HDAS's scenario of "once-write, multiple-reads", TimescaleDB is more suitable as the HDAS database. In the future, a TimescaleDB distributed cluster will be designed and deployed, and an HDAS prototype system will be developed based on this cluster.

REFERENCES

[1] Z. H. Bai, W. Li, Y. Li, G. Liu, and L. Wang, "Super-Period Locally Symmetric Lattices for Designing Diffraction-Limited Storage Rings", in *Proc. IPAC'19*, Melbourne, Australia, May 2019, pp. 1498–1500. `doi:10.18429/JACoW-IPAC2019-TUPGW041`

[2] Z. H. Bai *et al.*, "A Modified Hybrid 6BA Lattice for the HALF Storage Ring", in *Proc. IPAC'21*, Campinas, Brazil, May 2021, pp. 407–409. `doi:10.18429/JACoW-IPAC2021-MOPAB112`

[3] M. V. Shankar, L. F. Li, M. A. Davidsaver, and M. G. Konrad, "The EPICS Archiver Appliance", in *Proc. ICALEPCS'15*, Melbourne, Australia, Oct. 2015, pp. 761–764. `doi:10.18429/JACoW-ICALEPCS2015-WEPGF030`

[4] L. R. Dalesio, "NSLS II Control System Overview", in *Proc. ICALEPCS'07*, Oak Ridge, TN, USA, Oct. 2007, paper TPPB41, pp. 253–255.

[5] A. Wallander *et al.*, "News from ITER Controls - A Status Report", in *Proc. ICALEPCS'11*, Grenoble, France, Oct. 2011, paper MOBAUST01, pp. 1–4.

ISBN: 978-3-95450-238-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS2023-THPDP036

[6] Y. S. Qiao, G. Lei, and Z. Zhao, "The Study of Accelerator Data Archiving and Retrieving Software", in *Proc. IPAC'17*, Copenhagen, Denmark, May 2017, pp. 4007–4009. doi:10.18429/JACoW-IPAC2017-THPAB121

[7] MongoDB official website, https://www.mongodb.com/

[8] W.W. Shen *et al.*, "A new EPICS time-series data archiver using HBase", *Radiat. Detect. Technol. Methods*, vol. 513, pp. 550–557, 2021. doi:10.1007/s41605-021-00277-2

[9] N. Kikuzawa, Y. Kato, A. Yoshii, H. Ikeda, and N. Ouchi, "Status of Operation Data Archiving System Using Hadoop/HBase for J-PARC", in *Proc. PCaPAC'14*, Karlsruhe, Germany, Oct. 2014, paper FPO016, pp. 193–195.

[10] Apache HBase Reference Guide, https://hbase.apache.org/

[11] I. D. Finch *et al.*, "Vsystem to EPICS Control System Transition at the ISIS Accelerators", in *Proc. IPAC'22*, Bangkok, Thailand, Jun. 2022, pp. 1156–1158. doi:10.18429/JACoW-IPAC2022-TUPOPT063

[12] K. Xu, "CAFlux: A New EPICS Channel Archiver System", in *Proc. ICALEPCS'19*, New York, NY, USA, Oct. 2019, pp. 1470–1472. doi:10.18429/JACoW-ICALEPCS2019-WEPHA164

[13] InfluxDB Documentation, https://docs.influxdata.com/influxdb/v2.2

[14] K. Kasemir *et al.*, "TimescaleDB for Archived Data", presented at EPICS Collaboration Meeting, 2022, unpublished, https://indico.cern.ch/event/1173788/contributions/5012987/

[15] TimescaleDB Documents, https://docs.timescale.com

[16] S. Marsching, "Scalable Archiving with the Cassandra Archiver for CSS", in *Proc. ICALEPCS'13*, San Francisco, CA, USA, Oct. 2013, paper TUPPC004, pp. 554–557.

[17] Cassandra Documentation, https://cassandra.apache.org/doc