Evaluation of Time Series Database for IPS HDSR of MMIS

SeongJin Kim[a], Geunyoung Yoon[a], Hyeokjae Yoon[a]
[a] *Soosan ENS, Hyundai Venture Ville 3F 304, 10, Bamgogae-ro 1-gil Gananm-gu, Seoul, Korea*
[*] *Corresponding author: ksj1330@soosan.co.kr*

1. Introduction

The importance of data is increasing day by day in preparation for the approach of the fourth industry. In the field of nuclear MMIS (Man-Machine Interface System), a representative data management system is HDSR (Historical Data Store & Retrieve) of IPS (Information Processing System). The HDSR is one of the most important systems for operation and maintenance activities because it stores operation historical data and record the overall status of the event when an event occurs.

Generally, the data format of historical data for HDSR is configured with 'time', 'point id', 'value' and 'status of signal' and data is stored at every store cycle with that format. This time-oriented data set is called 'time series data', and the database that optimized for this data type is called 'Time Series Database (TSDB)'.

We selected InfluxDB which has high ranking in TSDB benchmark and evaluate for HDSR.

InfluxDB is a TSDB written in Google's GO language and can run on multiple operating systems. InfluxDB is available 'InfluxDB OSS' version and 'InfluxDB Enterprise' version. Enterprise version supports clustering function.

2. Evaluation of Time Series Database for IPS HDSR

2.1 Test Environment

The test environment consisted of one InfluxDB **server** and one signal stimulate computer. The operating system of the test server computer is Windows 7, the CPU is Intel Core i7 3770 (3.4 GHz, 4Core), and the RAM is 8 Gbytes.

2.1 Performance test for the store function

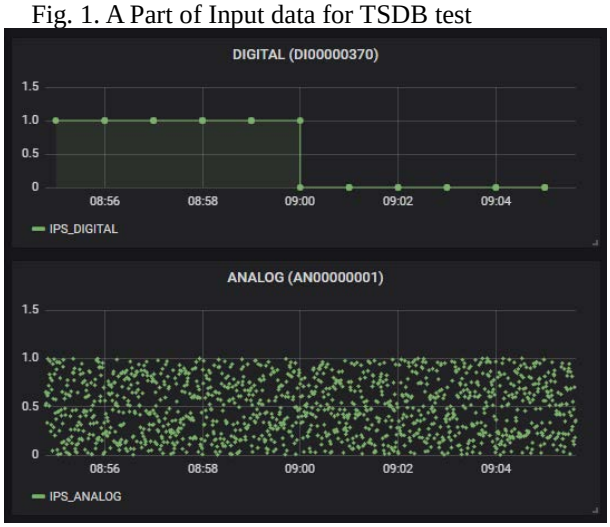
Time series data set for HDSR consisted of 'Time stamp', 'Point ID (TAG)', 'Value', 'Status of signal'. The number of 'Point ID' was estimated about twice for IPS of JRTR (Jordan research and training reactor).

Table I: Size of Test Data

Signal Type	Test Point ID	Number of Point
Analog	AN00000001 ~ AN00001000	1000
Digital	DI00000001 ~ DI00004000	4000
Alarm	AL00000001 ~ AL00002000	2000

First, we measured write throughput. This test measure that how much data can be store per unit time in TSDB. In this test, we put data to TSDB server continuously from a remote stimulate computer.

Fig. 1. A Part of Input data for TSDB test



Stimulated signal for alarms and digital was generated continually. That switched as 1 and 0 continually for each 1 hour. The analog signal was continuously changed for each 500ms using a random number generator for prevents internal data compression. (Fig.1)
The result of write throughput test is shown in Table II.

Table II: Write Throughput Test result

Item	Performance (Full Load)
TSDB Write Throughput	avg 66,000 points/sec
CPU Load	about 40~50 %
Disk Load	about 3.3 MByte/sec
Network Load	about 37.2 MBPS
Storage	1.2 GByte / 1 day, (7000 Points)

The measurement result for write throughput is 66,000 points per second (average). It means that system can be write 33,000 points data per 500ms interval.

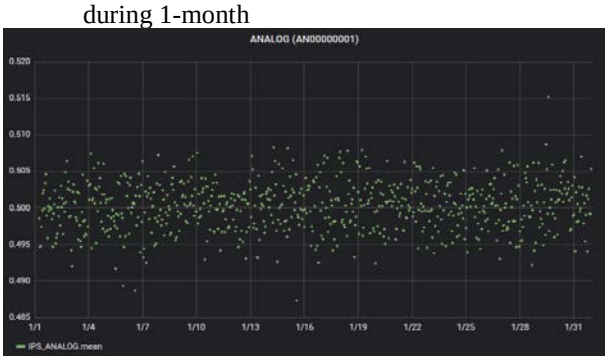
If system have 7000 point id and data stored 500ms interval, it uses 21.2% of maximum write throughput. Thus, it guaranteed 78.8% free load. This free load can be used for data retrieve or other purpose such as backup and maintenance.

The storage usage except for temporary files in the database consumed about 1.2Gbytes per day. The change of input value in actual plant is smaller than random value input, it takes up less capacity by internal data compression algorithm.

2.2 Performance test for the retrieve function

To be used as a HDSR, it is need to be able to retrieve quickly a large amount data. In particular, it should be possible to extract data with multiple resolutions. For example, it should be possible to quickly retrieve long-term data such as 1-minute interval average, 10-minute interval maximum value, and 60-minute interval minimum values. InfluxDB provides MEAN, MAX, MIN with GROUP BY time() query functions to quickly retrieve data in the desired format.

Fig. 2. Retrieve result for 1-Hour Interval Average data during 1-month



For the retrieve test, we measured response time for each point with various forms. The measured result shown in Table III.

Table III: Performance for Data Retrieve

Retrieve Form	Number of Point	Average Retrieve Time
500 ms interval RAW data (1-day)	172800	2.456 s
1 minute interval Average data (1-day)	1440	57 ms
1 hour interval Average data (1-day)	24	41 ms
1 hour interval Average data (1-Month, Fig.2)	744	1.715 s

2.3 Redundancy

InfluxDB Enterprise version supports DB Clustering. DB Clustering requires multiple meta servers and data servers. It is suitable for large scale clustering, but it is difficult to use clustering function for general dual redundancy. However, it is possible to give dual redundancy using external data collection application and DB synchronization application.

2.4 Backup and Recovery

InfluxDB provide the backup function that can back up or restore data from a specific time to a specific time. Therefore, periodic backups are possible with backups for the required periods at the required intervals. Daily backup time consume about 1 minute 15 seconds and consume about 650MB (490MB compressed) storage. Considering that the total amount of data of the analog signal (2 point/s * 1000 points* 3600 seconds * 24 hours * 4 bytes = 659.2 Mbytes), the data space is effectively used.

2.5 Fault Tolerance

When power of server was lost at the full load state, data was lost less than 1-second and there are data loss was not detected at the non-full load (about 3000 points / 100 ms) status.

3. Conclusions

The write throughput of the InfluxDB is measured as about 66,000 points per second. The retrieve performance is measured as 2.456 seconds for the daily data with 0.5 second intervals (172800 points).

It shows the InfluxDB has enough performance for IPS HDSR that handle less than 20,000 points per second at a HDSR server.

But, InfluxDB is not a full package such as Historian of SCADA System (Proficy, Ovation). In order to apply to the actual HDSR for IPS, it is necessary to design and implement additional collector that collects data from IPS and input it to TSDB and the redundancy control application such as database synchronization.

REFERENCES

[1] InfluxData, InfluxDB 1.7 documentation, 2018.
[2] JRTR MMIS I/O List for IPS, .2018.01
[3] SJ.Kim, JRTR Software Design Specification for IPS, Soosan ENS, 2014.07