21, rue d'Artois, F-75008 PARIS
http : //www.cigre.org

CIGRE US National Committee
2015 Grid of the Future Symposium

Investigation of PMU data loading performance
- Hurdles and potential improvements

N. NISHIKAWA[1], M. TAKATA[2], J. YAMAZAKI[1], B. YANG[1], P. KANSAL[3]
[1]Hitachi America, Ltd.
USA
[2]Hitachi, Ltd.
Japan
[3]American Electric Power
USA

SUMMARY

With the technical benefits that Phasor Measurement Units (PMUs) have to offer, they have become increasingly popular in the North American electrical grid. Due to grants from the American Recovery and Reinvestment Act (ARRA), the installation of PMUs has doubled in the past few years and reached 100% coverage of the U.S. transmission grid. A significant amount of R&D effort has been dedicated to enhance grid awareness and control with information from synchrophasor data. Many outstanding PMU data analysis tools have been developed, providing power engineers and operators abundant analytic resources to excel on grid monitoring and operations.

PMU measurements have been widely adopted for wide area monitoring and also seen as a valuable compliment to SCADA data for operations and planning. For example, operators can closely monitor grid dynamics beyond their own control area and between SCADA intervals. Such granular grid dynamics can efficiently visualize potential stability challenges, i.e. oscillations or unexpected tripping events. Planning and generation departments can also benefit from PMU data through offline model validation and post event analysis. As PMU coverage increases, synchrophasor data is expected to bring more benefits to power companies and change the way the power grid is planned and operated. To achieve this goal, the ideal synchrophasor infrastructure must be able to provide fast data loading, efficient data cleansing, and effective data analytics, which is not an easy task when the size of the databases increases. For a database of hundreds of terabytes in size, data loading can take hours for some queries. This prolongs not only the overall process for offline applications, but it also slows the potential adoption of synchrophasor data for control room applications. This paper will deeply dive into the data loading process, identify potential bottlenecks and explore potential strategies for performance improvement.

KEYWORDS

Phasor measurement unit - data ingestion - exporting - database technologies - power system applications

norifumi.nishikawa@hal.hitachi.com

Introduction

With the technical benefits that Phasor Measurement Units (PMUs) have to offer, they are increasingly popular in the North American power grid. Due to grants from the American Recovery and Reinvestment Act (ARRA) [1], the installation of PMUs has doubled in the past few years and reached 100% coverage of the US transmission grid. However, IT infrastructure has not grown at the pace of PMU deployment and it can be limiting as we continue to grow PMUs in the system. Among other IT challenges, database loading performance in terms of data ingestion and exporting can become an issue if database size grows from 10 to 100 times bigger than existing databases, regularly sized from 1 to 10s of terabytes. For example, it could take several minutes just to export 1 month of data for a few PMUs. The inefficient data loading processes prolong most offline analysis and poses challenges to integrating decision support tools, based on historic data analysis, into future online applications where data loading speed is critical. It is therefore important to understand the data loading process, identify bottlenecks, and advance technologies for performance improvement.

PMU applications

In the initial years of PMU deployment, the focus was mainly on the installation of the PMU device itself. Significant improvements have been made on PMU data quality in the last couple of years and the industry is more focused on translating PMU measurements to actionable information. The real-time applications interesting to power companies include oscillation detection and source identification, angle difference monitoring, voltage sensitivity monitoring, frequency response monitoring, linear state estimation, and wide area visualisation. Most of these applications are commercially available now and depending on the size of an individual utility/RTO's system, the number of PMUs feeding these applications can vary from tens to hundreds. For these online applications, only a short time window of PMU data is needed, hence data loading can be fast and performance is generally not a concern. However, with the rapid advances in data mining techniques, operational information embedded in historical data can become more useful for concurrent grid operations. This would necessitate faster data ingestion performance for online applications using decision support based on historical data.

Also, most offline applications like event analysis, system modal benchmark analysis, model parameter estimation, and system trending analysis, require either access to historical data or an understanding of previous events of a similar nature. Power engineers usually need to identify time frames of interest and retrieve data from PMU historian into the machine hosting the offline analysis tools for further process. The volume of data can become large when the number of PMUs, phasors or the time frame of interest increases [2]. It is estimated that total PMU historian will be more than 100 terabyte for one-third of the current installation and data loading will take more time, e.g. hours for some cases. This prolongs not only the overall process for offline application, but it also hinders the potential adoption of synchrophasor data for future online applications. The electric power industry will benefit from a high-speed data retrieval infrastructure, including a database engine and a retrieval mechanism, both of which can improve PMU data accessibility and strengthen data analytics. Faster data access can also be helpful in the future if decision support tools based on historic data analysis become an integral part of real-time applications in the control room.

Decoding PMU data loading process

The typical PMU data loading process is depicted in Figure 1. It can be conceptually broken down into five sub-processes. In the **Data Acquisition** process, PMU measurements are collected from sensors and sent through the communication infrastructure to a Phasor Data Concentrator (PDC) or Super PDC, often in the standard IEEE C37.118 format. The data transfer speed during this step depends on the communication protocol and communication infrastructure. The PDC or Super PDC then sends the data to the back-office, where it is consolidated (**Data Consolidation** Process) and sometimes compressed in a write cache. This process is fast because data is processed as it arrives in small batches. If there is any application requiring online visualization or analysis, the data batch is passed on to the application. The consolidated data is then indexed and stored in the database or file storage system (**Storage** Process). It includes raw measurements, time stamps and other derived labels, where the speed is highly dependent on database technologies or file storage systems.

When the offline PMU data application submits a query to the database, the **Data Retrieval** Process will read the data from the database and/or file system, decompress it (if needed), sort it and write it to data files (**Data Exporting** Process) for customer programs to use. If a large amount of data records are returned, the disk writing can take hours before a user can access the data. The file storage system, i.e. Hadoop, usually retrieves files, i.e. PMU data, more efficiently than a relational database. But it usually doesn't support business intelligence analysis such as data scan, customized query, and similarity search and so on. On the other hand, the Database server, usually based on a file system, can provide more complex functionalities, i.e. integration of user defined models etc. This can potentially reduce the overall data analysis time. For example, some trending and data driven analysis can be easily implemented in the database, which avoid data transfer between hard disk drives and reduce total analysis time.

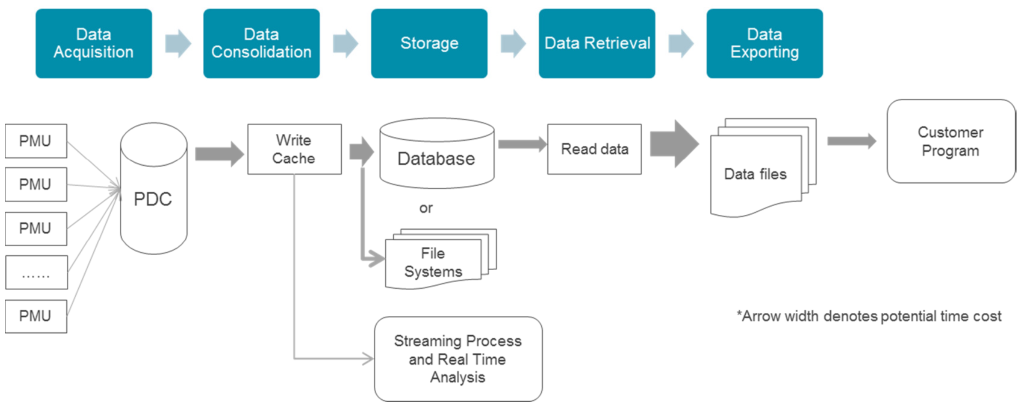

Figure 1 -- Sub-processes of PMU data loading

Most leading vendors of PMU data management tools adopt the process in Figure 1 more or less, while some of them use data compression to reduce the processing time. For example, the Operational Data Historian (ODH) process by a leading vendor [3] receives data from devices and sensors through an API, compresses the data, stores the compressed data into a write cache, then loads the data into a database on disks. For data exporting, the ODH process reads required data from the database on the disks, decompresses the data, then sends the decompressed data to a client program through the API. The ODH compress PMU data, thus it is suitable for handling large PMU data. In another PMU data platform [4], PMU data are sent to the PDC server through a wide area network using various data formats such as IEEE C37.118. Then the PMU data is transferred to a historian and PMU data analysis tools get the data from the historian. The PDC server sorts and adds timestamps to the data, executes some actions such as real-time event detection, and then sends the data to the historian. The historian stores the data in flat binary archive files. The power client offline tools retrieve data from the historian and execute off-line analysis. This platform contains online and offline analysis tools. Grid operators may be able to monitor and control power grid easily.

Performance investigation

Since many of the PMU data management systems are based on relational databases, the overall process depends on the performance of the database engine. This paper will focus only on Data Retrieval and Data Exporting processes (Figure 1), as they represent typical data requests for offline PMU analysis. Data Acquisition and Consolidation processes will not be discussed here. The Hitachi

Advanced Database (HADB)[*1] [8] is used in this experiment. The impacts of different database technologies are also discussed conceptually through performance a comparison between HADB and a conventional database management system.

As in Figure 2, the PDC sends a data packet to the Write Cache every 2 seconds, which is then written to temporary files without compression. The Data Loading tool (DLT) reads those files and loads them into the DB every 5 minutes. The process continues until no more data comes in from PDC. Then the Relational Database Management System (RDBMS), in this case HADB, reads PMU data from the database on the hard disk drive and sends them to server memory. Here, HADB is one implementation of the RDBMS. The HADB then sorts data by PMU ID and timestamp. Data Export Tools (DET) transfer data and write them into files in parallel, which is ready to be loaded and analyzed by offline analysis tools. The HADB contains DLT, DB, RDBMS server and DET.

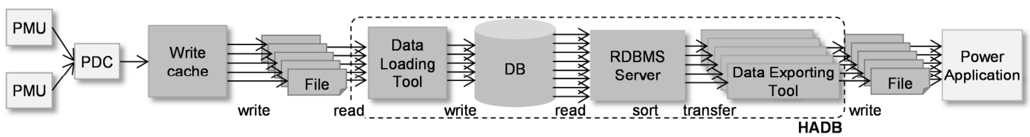

Figure 2 -- Testing setup

The **testing environment** is set up as in Figure 3, which consists of a server with 2 CPUs and 126 GB memory. The storage has 24 600GB SAS (Serial Attached SCSI) hard disks and 4GB cache memory. The server and storage are connected with four 8 Gbps Fibre channel cables to minimize potential latency. Our testing environment also contains RDBMS software, where the HADB is used.

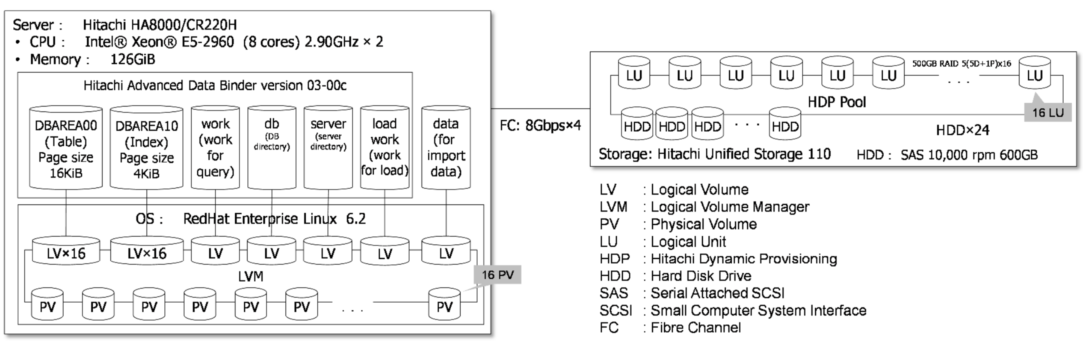

Figure 3 -- Testing system architecture

Testing dataset: Simulated PMU data was adopted for this test, where 24 PMUs from different vendors were simulated in a Hardware-in-the-Loop environment for 60 days. Each PMU collects 17 signals. The total number of records is approximately 63 billion, which is around 6 terabytes. Data is ingested into the HADB without compression, including raw PMU measurements, time stamps and asset information. The data complies with the Time Series Data Access (TSDA) [5] format.

Table 1 -- Tested queries and results

Query ID	# of PMUs	Time Duration (day)	# of data channel	# of records	Response time in second
1	1	1	1	2,592,000	5
2	1	1	9	77,760,000	56
3	1	30	1	20,736,000	10
4	1	30	9	622,080,000	271
5	5	1	1	12,960,000	9
6	5	1	9	388,800,000	173
7	5	30	1	103,680,000	41
8	5	30	9	3,110,400,000	916

*1: Out-of-Order Execution Principle developed by Dr. Kitsuregawa, the Director General of the National Institute of Informatics and Professor of The University of Tokyo, and Dr. Goda, Project Associate Professor of The University of Tokyo.

Tested queries: Eight example queries are tested, which retrieve one or multiple data channels from multiple PMUs at different time durations. Table 1 has details of the queries, the total number of records that are returned, and their performance. The returned data records-range from 234 MB for Query 1 to 274 GB for Query 8.

Performance: The data retrieval speed ranges from 0.5 Million rec./s to 3.39 Million rec./s. The data throughput increases almost linearly when the retrieved data volume increases. The query time ranges from 4.9 seconds to 15.3 minutes. Figure 4 shows an example of the performance breakdown for the Data Retrieval Process and Data Exporting Process. The former includes Read (from DB on disks) and Sort (according to PMU ID and time stamp), and the latter contains Transfer (from HADB to DET) and Write (to data files on disks). The Read requires intensive disk I/O utilization (I/O wait rate is approximately 50%) and Sort process requires intensive CPU utilization (approximately 100%), while the Transfer & Write process consumes over 50% of the overall time. In HADB, data reading is governed by the availability of data rather than sequence of query orders in order to minimize idle time for the processors. Hence the data throughput and CPU efficiency are maximized and total time cost for the Read process is reduced.

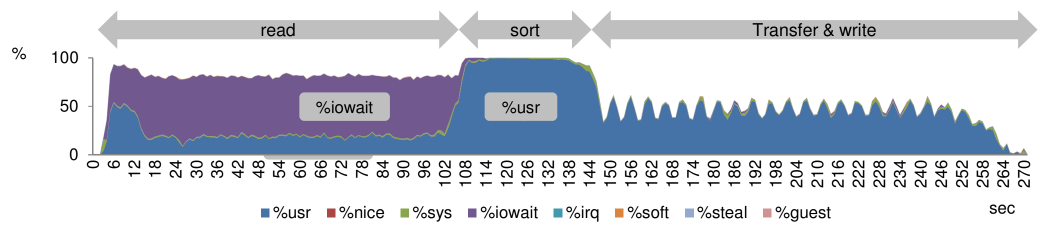

Figure 4 -- Query performance of each sub-process

Potential impacts of different database technologies

It is also evaluated how database technologies may impact overall performance. Two typical queries are tested on both HADB and a widely used conventional database (CDB) under a similar testing environment, as in Figure 3. First query requests for 4 PMUs' data for 10 minutes, which are often used to analyse temporal development of local dynamics, such as frequency baseline or system trend analysis and so on. A second query searches for a snapshot of all PMU data (500 in total). The system snapshot can be used for linear state estimation, voltage contour, etc. The execution time of both queries in HADB and CDB are shown in Figure 5. The blue bar indicates data search time with CDB while the red one shows HADB performance, which consumes significantly less time.

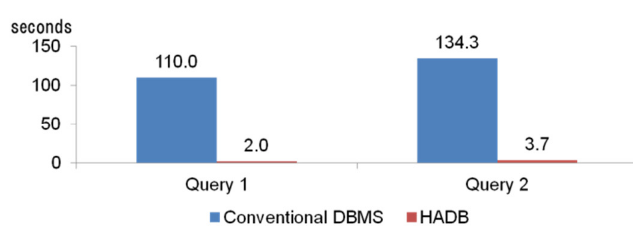

Figure 5 -- Data search performance comparison

Performance improvement

The performance investigation illustrates the breakdown of process time and server workload for the typical data loading process, which doesn't represent the optimal performance for HADB. For example, there are a few ways that the data loading performance can be further improved.

- HADB features novel asynchronous I/O processing techniques and consumes less processing time as the number of storage disk increases [6]. Multiple nodes can also release the CPU bottlenecks as in Figure 4 and further improve performance.
- Applying data compression techniques can significantly reduce data volume and thus reduce data loading time. For example, compressed data can be one tens or less of the original size, which can then be processed in memory and is much faster.
- Data pre-processing/mining before archiving can export useful data signatures/patterns and serve as an informative index with much lower dimension. For example, oscillatory stability and voltage stability of power systems can be used as signatures of impending system events [7]. Thus previous oscillation or voltage instability can be easily retrieved and used as reference.

Conclusions

As shown in Figure 4, the data loading process is broken down into roughly three processes: 1) Read (from DB), 2) Sort (according to PMU ID and time stamp), and 3) Transfer & Write (from HADB to data files). The first and third processes take more than 80% of the total time, which shall be the future target for improvement. The first process reads data from storage governed by the database software. Its performance is highly dependent on the efficiency of the database technology and can potentially be improved through advances in database software, optimizing data table structure, and strengthening the computation capability of the server. The third process can be executed in a distributed file system, i.e. Hadoop, to achieve better performance. Another possible way to improve the third process is to minimize or even avoid the need for data transfer between hard disk drives. For example, one potential strategy is to integrate commonly used data analysis functions into the database software. Future research will target optimizing and benchmarking the performance of HADB versus commonly used conventional database management systems.

BIBLIOGRAPHY

[1] ARRA: https://www.smartgrid.gov/files/PMU-cost-study-final-10162014.pdf

[2] "Designing IT and Communications for PMU applications," (Synchrophasor fundamental tutorials at PES GM 2013)

[3] OSIsoft PI System. http://www.osisoft.com

[4] M. Chenine, L. Vanfretti, S. Bengtsson and L. Nordstrom, "Implementation of an Experimental Wide-Area Monitoring Platform for Development of Synchronized Phasor Measurement Application," (International Implementation Experience and Prospective Applications of Synchrophasors and their Supporting Infrastructures Panel Session, IEEE PES GM 2011)

[5] "Energy management system application program interface (EMSAPI) Part 407: Time Series Data Access (TSDA)," (International Electrotechnical Commission, 2007)

[6] Yutaka Kokai, "Realization of Big Data Platform for PMU to Improve Control Center - Real-Time Data Processing & Historical Data Analysis," (North American Synchrophasor Initiative Working Group Meeting, 2014)

[7] Vijay Vittal, Trevor Werho, Mladen Kezunovic, Ce Zheng, Vuk Malbasa, Junshan Zhang, Miao He, "Data Mining to Characterize Signatures of Impending System Events or Performance from PMU Measurements," (Power Systems Engineering Research Center final project report, 2013)

[8] Masaru Kitsuregawa, Kazuo Goda, "Vision and Preliminary Experiments for Out-of-Order Database Engine (OoODE)," (DBSJ Journal, Vol.8, No.1, 2009)