THE LHC LOGGING SERVICE: HANDLING TERABYTES OF ON-LINE DATA

C. Roderick, R. Billen, R. D. Gaspar Aparicio, E. Grancher, A. Khodabandeh, N. Segura Chinchilla,
CERN, Geneva, Switzerland

Abstract

Based on previous experience with LEP, a long-term data logging service for the LHC was developed and put in place in 2003, several years before beam operation. The scope of the logging service covers the evolution over time of data acquisitions on accelerator equipment and beam related parameters. The intention is to keep all this time-series data on-line for the lifetime of LHC, allowing easy data comparisons with previous years. The LHC hardware commissioning has used this service extensively prior to the first beams in 2008 and even more so in 2009. Current data writing rates exceed 15TB/year and continue to increase. The high data volumes and throughput rates, in writing as well as in reading, require special arrangements on data organization and data access methods.

INTRODUCTION

The *LEP Logging System* [1] was put in place in 1992, three years *after* the commissioning of LEP. The raison-d'être was that at that time, the LEP behaviour was not yet understood. Systematic logging of LEP hardware and physics parameters over time, and being able to correlate this data, enabled a better understanding and performance increase of LEP operation.

For LHC, the project for a logging system was started in 2001, several years before beam operation. In 2004, the first logged data concerned vacuum pressure acquisitions of the injector machine SPS. Since then, practically all LHC sub-systems make use of the Logging Service to store and retrieve data acquisitions.

LOGGING EXPERIENCE

The LHC Logging project was based on experience with previous efforts, namely for LEP and for the LHC magnet test stand. The experience also showed the usefulness to keep all historic data readily available, during and beyond the lifetime of the installation. Several years after the dismantling of LEP, the logged data continued to be scrutinized to calibrate the centre-of-mass energies.

With the technical know-how available, the design and implementation of the LHC Logging Service was performed in-house.

ARCHITECTURE AND DESIGN

A variety of data acquisition systems on custom front-end computers and industrial supervision systems provide the time-stamped data. This data is captured and stored in a database management system (DBMS), along with configuration meta-data, user profiles and audit data statistics. End-users have interactive and applicative interfaces to retrieve data into standard file formats or to visualize the trends graphically.

The initial functional requirements of this central Logging Service were straightforward, while the challenge was to be found in the scalability and performance that needed to be satisfactory over the years.

IMPLEMENTATION

The implementation is in line with the overall data management of the accelerator complex [2]. For the in-house developments, the underlying repository is the Oracle® Database in a RAC setup (Real Application Cluster). The most important aspects are summarized below, whereas more technical details are described elsewhere [3].

Database Objects

Each individual data channel (*variable*) is uniquely identified and registered in advance in the database. This makes up the meta-data, necessary for its common understanding. Only a limited number of data types are supported, the numeric data type being the most generic one. One table per data type holds the data trend, with each record being composed of:

{variable_id, UTC-timestamp, value}

These skinny 3-column tables become huge in size with the ever-growing number of rows (10^{11}-10^{13} records). Instead of maintaining two separate storage structures, one for the ordinary (heap-organized) table and one for the B-tree index, *index-organized-tables* (IOT) are used – physically storing key and non-key column values in a single B-tree index. The database handles all manipulations via the B-tree index, which is more costly at insert time, but more effective for reading.

The data tables are compressed and range-partitioned over time (hourly and daily).

Data Access

For all read and write access to the database objects, Application Programming Interfaces (API) have been developed. This enforces a tight control on how the applications access the data and ensures optimal performance. The implementations have different flavors:

- XML-file based for bulk loading from industrial data acquisition systems
- JDBC based for data via Java applications
- PL/SQL based for database-to-database transfer

Space Management

For effective space management, the partitioned tables and indexes are also spread across daily *tablespaces* (i.e. a logical storage unit consisting of an auto-extensible *datafile* which is physically located in the file system of the server). Every night, new table partitions and corresponding tablespaces are created for future data (five days ahead) and tablespaces older than 14-days are made read-only (RO). The correct execution of this programmed procedure is also automatically checked and notified to the administrators in case of failure.

Backup Strategy

Establishing the appropriate backup strategy for such a large and I/O intensive database is a real head breaker. Table 1 shows the three types of backups that are scheduled repeatedly, with their current frequency and duration.

Table 1: Current Backup Schedule

DBMS Backup Type	Frequency	Duration
Incremental	1 per day	± 1 hour
Full, without RO data	1 per week	± 8 hours
Full, all data	1 per 3 weeks	± 6 days

In addition, backing up of the latest database transactions in the redo log files take place every 15 minutes. Since the backup to tape processes are demanding on I/O resources as well, they have a non-negligible impact on the user processes. However, the strategy does ensure a complete recovery after a database crash. In case of such an event, an immediate but partial recovery is anticipated to minimize the database downtime. Such a partial restoring of the database onto an independent server is tested on a regular basis in order to validate the tape backup sets.

INSTRUMENTATION

Monitoring of what is going on and investigating what is going wrong is quite difficult in three-tier architecture. The importance of instrumentation and adequate monitoring tools at all levels is primordial.

Database and Application Servers

The Oracle Enterprise Manager (OEM) is a powerful tool to monitor complete vertical deployments across database and application server platforms. Detailed performance metrics can show the detailed activity of all sessions and which resources are stressed (e.g. CPU, I/O, concurrency, network, etc.). A configurable 7-day history of this information is extremely useful to identify offending processes.

Applications

All applications that access the database are declared in advance and need to identify themselves before being granted a connection. From that moment on, all application transactions and requests are monitored and registered in terms of number of variables, data volume and elapsed time. This instrumentation is mainly implemented using Java *managed beans* (MBeans) which are exposed for real-time consultation and configuration via any JMX (Java Management Extensions) interface – such as the one included in the OEM.

Server Hosts and Storage

All CERN-IT managed server hosts are monitored on performance metrics, exceptions and status information by means of sensor agents installed on the hosts. For our servers and clusters, the statistics are presented graphically in a CERN-IT accessible web tool called *Lemon* (LHC-Era Monitoring).

The monitoring of the NAS (Network-Attached Storage, i.e. the disks and their dedicated controllers) is done via the tools of the storage provider NetApp®. This detailed information is complimentary to the performance metrics available through OEM.

THE CHALLENGES AND PARADOXES

Competing requirements lead to careful decision taking and consensus for a number of subjects.

Data Volumes

In the scientific and engineering environment of the LHC, the clients of the Logging Service would like to keep all data acquisitions at the highest possible frequency and have it immediately available for analysis. Technical (and budgetary) limitations do apply on a centralized system serving a wide user community. A compromise for data reduction is therefore mandatory. The *Measurement Service*, acting as a high-throughput, short-term front-end data store of the Logging Service, aims to reduce the data volumes by appropriate filtering.

Database Flavor

The data loading and data processing part requires typically an OLTP-oriented system (on-line transaction processing). On the other hand, the data retrieval and analysis part tends towards an (off-line) data warehouse solution. The need for rapid availability of the data imposes the OLTP implementation. This puts the pressure on finding a scalable solution. Again, the controlled access via API, used by registered applications is a necessity.

Database Versioning

The newest functionality and features of the technology stack cannot be used as soon as they become available. Stability is fundamental in our production environment, especially for a mission-critical service. For example, the *log errors* clause in the bulk data insert statement – only available as from Oracle Database 10*g* – was eventually used to catch and diagnose data loading errors. Also, it would be beneficial to make use of the *Database Resident Connection Pool* in Oracle Database 11*g* for efficiently handling a large number of independent connections.

Hardware

The anticipated lifetime of the LHC Logging Service (20-25 years) exceeds the lifetime of the technological components of the supporting hardware technology. Typically after 3 years the maintenance cost start rising for components that are becoming as rapidly out of date. Server processing power can be added or replaced, but for the multi-terabyte disk storage, the issue is more delicate. This has brought us in a situation whereby the data is spread over several systems from different manufacturers. Again, the API is in charge of handling the retrieval from the correct platform. A regular migration of the data to new storage is not yet a financially viable solution.

Clustering

High-availability (HA) is an essential requirement for the on-line accelerator databases, including the Logging Service. Some of the data is used for decision making after an unforeseen beam dump event or cryogenics quench. Database deployment across a cluster of servers addresses both HA and scalability. Database activity on a failing server is transparently taken over by another server in the cluster. Adding additional servers to the cluster increases the processing and the traffic over the server interconnects. That is the reason that today, 70% of these HA-setups are done with 2-node clusters. As a side-effect, half of the server's resources are practically unused.

STATUS AND OUTLOOK

The LEP Logging database, spanning a period of almost a decade, contains 20GB of data, equivalent of what is currently logged in less than a day. The initial figures in 2003 for LHC were estimated at 1TB/year during normal operation. Presently, data rates are already one order of magnitude higher than first estimated, and LHC is not operating with beam yet.

As far as data extraction is concerned, and despite the fact that this activity is monitored, it is extremely difficult to make any estimation or assumption for the future. Exploiting at full the dynamic capabilities of the instrumentation in the applications, will surely help to understand the patterns in client behavior.

SCALABILITY

The strategy was already outlined in 2001 to have scalability built-in with an optimum "relational" design. The correctness of the application design is confirmed by the monitored metrics. A quite stable database load, mainly I/O bound, with very little statement parsing and little traffic on the cluster nodes interconnect indicates a wise usage of the RAC. Consequently, the scalability issue then needs to be addressed by adding more hardware components in the architecture.

New disk arrays are being purchased just-in-time. Adding nodes to the cluster can be done whenever necessary, combined with a careful organization of the service distribution between the nodes in order to minimize interconnect activity.

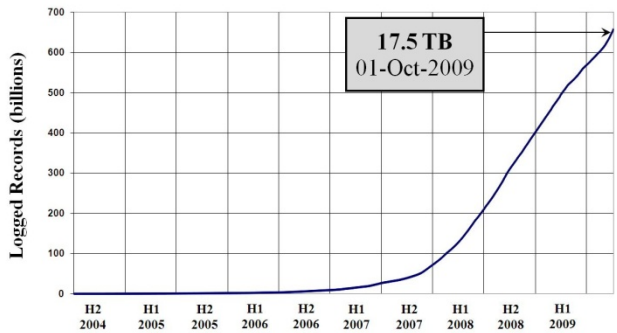

Figure 1: Evolution of logged data.

The logging user community and its requirements have grown substantially, even more so since the unfortunate LHC incident on 19 September 2008. More than 650 billion records have been logged i.e. 17.5TB worth of data; currently over 1.5 billion records are stored for the long term every day.

The backup strategy will need to be scrutinized as well. At the end of September 2009, 65TB disk storage has been attached to the accelerator databases, with a backup to disk solution in mind. Fine tuning of the backup parameters and a specialised protocol to transport data between NAS and backup devices will be explored.

CONCLUSION

The LHC Logging Service has proved its usefulness since several years. The demanding requirements for a centralized, long-term multi-terabyte system have imposed thoroughness in technical choices and compromises. Continuous performance monitoring is a necessity to ensure quality of service. Nevertheless, the challenge to keep up with the increasing expectations remains. A wide client community is looking forward to capturing, analyzing and exploiting LHC beam data.

REFERENCES

[1] R. Billen *et al*., "LEP Accelerator Logging System Using On-line Database", ICALEPCS'93, Berlin, Germany, October 1993, NIMA 352 (1994) 296-299.

[2] R. Billen *et al*., "Accelerator Data Foundation: How It All Fits Together", ICALEPCS'09, Kobe, Japan, October 2009, TUB001.

[3] C. Roderick and R. Billen, "Capturing, Storing and Using Time-Series Data for the World's Largest Scientific Instrument", November 2006, CERN-AB-Note-2006-046 (CO).