DATA ARCHIVING AND RETRIEVAL FOR SPRING-8 ACCELERATOR COMPLEX

A. Yamashita, T. Fukui, M. Kodera, T. Masuda and R.Tanaka, SPring-8, Hyogo, Japan

Abstract

The data archiving and retrieving system based on a relational database management system has successfully managed SPring-8 accelerator complex control system for two and half years. The scalability of the database system is flexible enough to require no major modification on the database for the addition of accelerator subsystems. We describe the database system focused on data access methods in data archival and retrieval.

1 INTRODUCTION

The database of the SPring-8 control systems, which was initially designed to control the storage ring, has been growing to manage SPring-8 complex including beamlines [1], the injector synchrotron [2] and the New SUBARU [3].

We designed the system to store every data, store that data forever and to retrieve them from everywhere.

"Every data" means logging of large numbers of data from equipment like vacuum gauge, klystron voltage, magnet current etc. in an uniform way. Data acquisition programs running on workstations constantly collect the data from VME systems distributed around the accelerator complex.

We archive every data without selection. Because at a third generation light source, where quality of beams is essential, a small fluctuation of a component may perturb the beam quality. For critical study of the beam, accelerator scientists need data from every component of the machine to find the source of the perturbation.

The data are stored forever. For example, whenever one wants to compare any data to that of the data taken one year ago, one can retrieve them as easy as retrieving the latest data.

"Everywhere" means database users can obtain real-time and archived data independent of their client computers and their location on the SPring-8 site. Here, the users of the database are accelerator physicist, experimentalist of the beamline and accelerator operators.

For example, experimental users can get the beam current by accessing the database at their beamline. Not only operators in the central control room, but also beamline experimentalist can monitor and control equipment by relying on the database.

Besides the requirements above, reliability, scalability and ease of maintenance are essential. To satisfy those demands, we manage the data with a commercial relational database management system (RDBMS), Sybase Adaptive Server Enterprise 11.

2 DATABASE

The structure of the database for data archiving has two layers, on-line for logging real-time data and archive database for long time storage. Another database, the parameter database, manages the configuration of the equipment and the structure of the data.

2.1 On-line Database

On-line databases are built to store and share real-time data of equipment. At the beginning of the run in fall 1999, it contains 11168 data points of real-time data from 8343 equipment. The on-line database consists of 108 active tables in 400M bytes including indices. The number of signals, equipment and tables are summarized in Table1.

Table 1: Number of signals and tables of each component.

Components	Number of equipment	Number of signals	Number of tables
Storage ring	2598	4703	34
Beamline	4375	4549	57
Synchrotron	1173	1524	11
New SUBARU	197	392	6

We designed the on-line database to be fast and sure data entry. To reduce the I/O time, the number of SQL commands to enter the data should be minimized. Data acquisition processes issue one SQL command to one table per one data acquisition cycle. One row of a table contains as many columns as possible. The number of columns is limited to 250 by the database management systems. The tables of the on-line database have a fixed number of rows (<4000) and a small size to fit into the data cache of the RDBMS. The RDBMS allocates a large amount of data cache (1.4GB), to reduce the amount of physical I/O to the disks and the time of data query. SQL commands *update* columns instead of *delete/insert* to minimize machine cycle inside the RDBMS.

The structure of a table is very simple. As shown in Figure 1, a table in the parameter database manages the location of each signal. The first column of an on-line table has a sequential number, the second column stores the time stamp of the acquired data and other columns are used for 32-bits integer or floating point data.

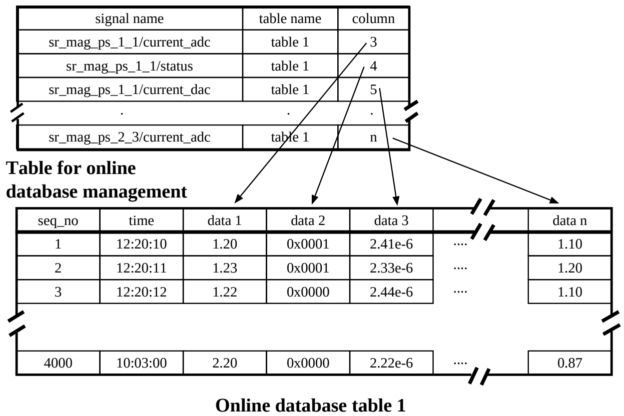

Figure 1: Example of on-line table schematics and a parameter table for signal location management.

Equipment status like valve open/close are packed in data records of maximum 30 bits. Other 2 bits are reserved for the status of data acquisition result. If the data taking fails due to a read error, the highest bit(MSB) is set to 1 which is easy to find as a negative integer value. One more bit is reserved to indicate an equipment trouble status. Analog data like voltage, current and vacuum pressure are stored into floating points columns.

A total of 12 acquisition processes [4] are running for each equipment group as SR-magnet, SR-RF, synchrotron, beamline insertion devices etc. One process writes the data at various frequency from 1 to 120 seconds depending on the requirement of each equipment. On-line database has a size of table sufficient to hold real-time data for 1 to 10 hours.

In addition to the data taken from equipment, the on-line database stores current status of alarms as well [5].

2.2 Archive Database

In fall 1999, the archive database, designed to store data for a long time, reached its size of a total of 50GB including index.

Archiving processes wake up once per one hour and sample the data in 1 to 120 seconds cycles from the on-line database and copy them into the archive database. The data can also be manually stored to the archive tables with arbitrary sampling rate on demands. Tables in archive database have the identical schemata and names to the corresponding ones in the on-line database. It enables seamless data access to both databases.

The size of tables become larger with time but they are not divided to reduce the size. We choose simplicity of software and ease of management over advantages of accessing speed which would be obtained by dividing the tables. With the help of indices of sequential numbers and time columns, it is possible to speed up the data access for accessing the data.

In addition to equipment data, which are copied from the on-line database, COD (closed orbit distortion) data are directly written to the archive database every 30 seconds or manually upon operator's request.

3 DATA RETRIEVAL

SPring-8 database users can retrieve data in two ways. One is by using C functions and the other is via WWW.

3.1 C Function

The application programs for SPring-8 device controls have been written by equipment experts. Because they are not software experts, the retrieval functions were designed as simple as possible to incorporate them easily into their applications. Users prepare their programs using data access APIs written in C language and merely have to know the signal name and recorded date/time to retrieve the favorite data from the database. They need not to know which table contains its target data nor which database to be read either from archive or on-line. The function takes care of all of them. For example, one can get the newest integer value of "sr_mag_ps_b_1_1/status" by calling as

db_get_online_newest_i("sr_mag_ps_b_1_1/status",
&int_value);.

We embedded the location of each signal into a hash table in a C function instead of asking the RDBMS their location signal by signal. With the help of a perfect hash method [7], the function searches location of signal in the database very quickly in 7 microseconds on a Hewlett-Packard's PA-7200 120MHz workstation. Although this method needs a compile and link cycle when a signal added, we choose it because of the advantage in the performance.

3.2 WWW

Besides the device control and off-line analysis applications, the database users retrieve their equipment data with a WWW browser everywhere on the site. Cgi programs written in Python [8] dynamically draws graphs with gnuplot 3.7 [9] in gif format. An example of integer signal graph with bit explanation is shown in Figure2.

Users also can download their data in text file format from WWW browsers. Text file is useful to perform analyses with combination of graph applications running on PC or Macintosh platforms.

4 HARDWARE

The main database server is a Hewlett-Packard's K250 with four PA-8000 160MHz CPUs and 2GB main memory. For redundant disk storage, we rely on mirroring managed by the RDBMS rather than hardware RAID disk system.

The previous server which employed two PA-7200 CPUs at 120MHz retired because of heavy load due to the increased amount of data.

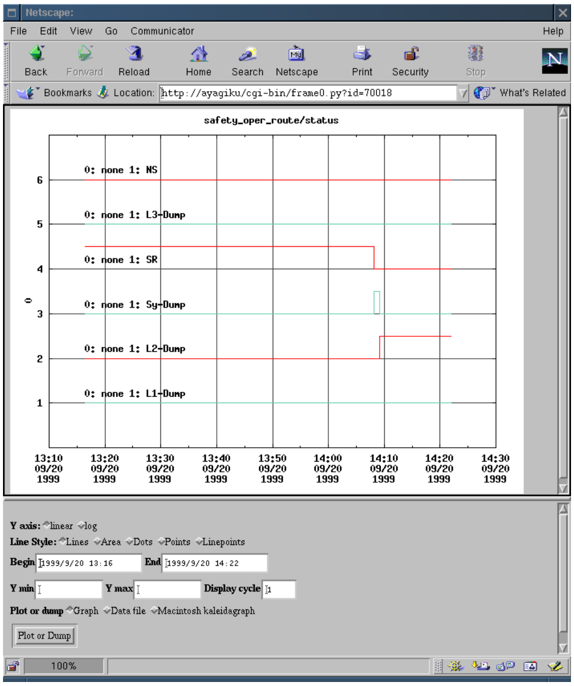

Figure 2: Example of integer signal graph displayed in Web browser.

The database on the main server are replicated to the secondary server for back up and off-line analysis. If the main server fails, the secondary server takes over from the main server by switching its IP address. Because the lack of clustering facility, this switching is done manually, not automatically. We expect higher availability by clustering with the next version of the RDBMS.

The secondary server has an almost identical configuration as the main server except no disk mirroring. It is also used for WWW server.

5 OPERATION

The database has been serving the control system since the commissioning of the storage ring spring 1997. Since then, the storage ring has experienced no major down time due to the database failure, while the control system indispensably depends on the database. The RDBMS was shut down only at maintenance time of the accelerator and never hung up for long time because of caused by the software during these two and half years.

With increasing amount of data, the load on the database server computer in approaching its hardware limit. By virtue of the progress of PC hardware cost/performance and open software operating system, the next candidate of the server computer may be a Linux server running on inexpensive PC platform. The preliminary R&D already started and the first results are reported very promising[6].

6 CONCLUSION

The data archiving and retrieving system for SPring-8 accelerator complex has grown in size, both in the number of managing equipment and users. The database system, which plays an essential role in the control system of SPring-8, proved its reliability and scalability in a large accelerator complex.

7 REFERENCES

[1] T. Ohata et al., “SPring-8 Beamline control system”, in these proceedings.

[2] N. Hosoda et al., “Integration of the booster synchrotron control system to the SPring-8 control system”, in these proceedings.

[3] A.Ando et al., “VUV and Soft Xray Light Source NEW SUBARU”,PAC’97, Vancouver, Canada, 1997

[4] A. Taketani et al., “Data acquisition system with database at the SPring-8 storage ring”, ICALEPCS’97, Beijing, China, 1997.

[5] A. Yamashita et al., “The alarm system for the SPring-8 storage ring”, ICALEPCS’97, Beijing, China, 1997.

[6] A. Yamashita et al,. “RDBMS on Linux for accelerator control”, PCaPAC’99, Tsukuba, Japan, 1999

[7] http://ourworld.compuserve.com/homepages /bob_jenkins/perfect.htm

[8] http://www.python.org/

[9] ftp://ftp.dartmouth.edu/pub/gnuplot/