SCALING UP OF THE MADOCA DATABASE SYSTEM FOR SACLA

T. Hirono, T. Hamano, A. Yamashita,
JASRI/Spring-8, Sayo, Hyogo 679-5198, Japan

T. Fukui, K. Hagihara, T. Maruyama, K. Nemoto, M. Yamaga,
RIKEN/SPring-8, Sayo, Hyogo 679-5198, Japan

Abstract

The database system in Message and Database Oriented Control Architecture (MADOCA) is required to be highly redundant and stable owing to MADOCA's reliance upon this system. We have designed and scaled up the database system for MADOCA in SPring-8 Angstrom Compact Free Electron Laser (SACLA) to reduce the server load. It consists of two database servers for operation and one server for relief. The first server for operation, the main server, holds small but frequently accessed data, whereas the second server, the archive server, holds data that is accessed in large blocks. The installation was successfully completed during the summer shutdown of 2012, resulting in a drastically lower central processing unit (CPU) load.

INTRODUCTION

MADOCA [1] was adopted to control the SACLA system [2]. The database system in MADOCA is one of its vital components. Data logging, alarm monitoring, access control of signals, and other features of MADOCA rely heavily upon its database system [3]. High redundancy and stability are required to realize a reliable control system for accelerator operations. We designed the MADOCA database system of SACLA as a copy of that of SPring-8. This system consists of one main server with network storage and a relief server with local disk storage (Fig. 1(a)). This system served as a redundant system during the first stage of commissioning of SACLA.

The number of signals required to control MADOCA has increased drastically. The number of signals exceeds 45,000, which is larger than that for any other MADOCA system. The development of accelerators and beamlines is anticipated in a few years. This would lead to an increase of more than 20,000 signals, which is almost two times larger than the number of signals controlled by SPring-8.

Therefore, we have designed and installed a scaled-up MADOCA database system. We explain the structure and installation of the system in the following sections. Finally, we summarize our study in the final section.

STRUCTURE OF THE SYSTEM

The schematics of the scaled-up database system and the original database system are shown in Fig. 1. The MADOCA database system adopted Adaptive Server Enterprise 15.5 Linux 64-bit (Sybase ASE) [4] as a relational database application. The original database system was designed to realize all features of the MADOCA database system by one server. This server was run by Sybase ASE. The second server of the original system was used as a relief server; however, the relief server was not used in normal operations. Our improved database system uses two database servers for normal operations and one server for relief. The two servers are fault tolerant (FT) servers that realize high redundancy. When one of servers is broken, the relief database takes its place. Table. 1 lists the specifications of each server. The CPU processing power, total amount of memory, and disk I/O speed of the scaled-up database system were expected to be more than double than those of the original database system.

Table 1: Specification of the scaled-up MADOCA database system

Main server	
Machine	NEC Express 5800 R320b-M4
CPU	Xeon 2.93 GHz x 12 core (8 cores for Sybase ASE engine)
Memory	48GB
Disk	Local disk: SAS 15krpm 1TB
Archive server	
Machine	NEC Express 5800 R320a-E4
CPU	Xeon 2.00GHz x 8 core (7 cores for Sybase ASE engine)
Memory	48GB
Disk	Local disk: SAS 15krpm 177GB Fiber channel: 8TB
Relief Server	
Machine	HP ProLiant DL180 G6
CPU	Xeon 2.67GHz x 12 core (8 cores for Sybase ASE engine)
Memory	48GB
Disk	Local disk SATA 15krpm 600GB x 14

There are five groups of client processes in the MADOCA database system. The first two groups involve data logging. Data logging is one of the most important

hirono@spring8.or.jp

ISBN 978-3-95450-139-7

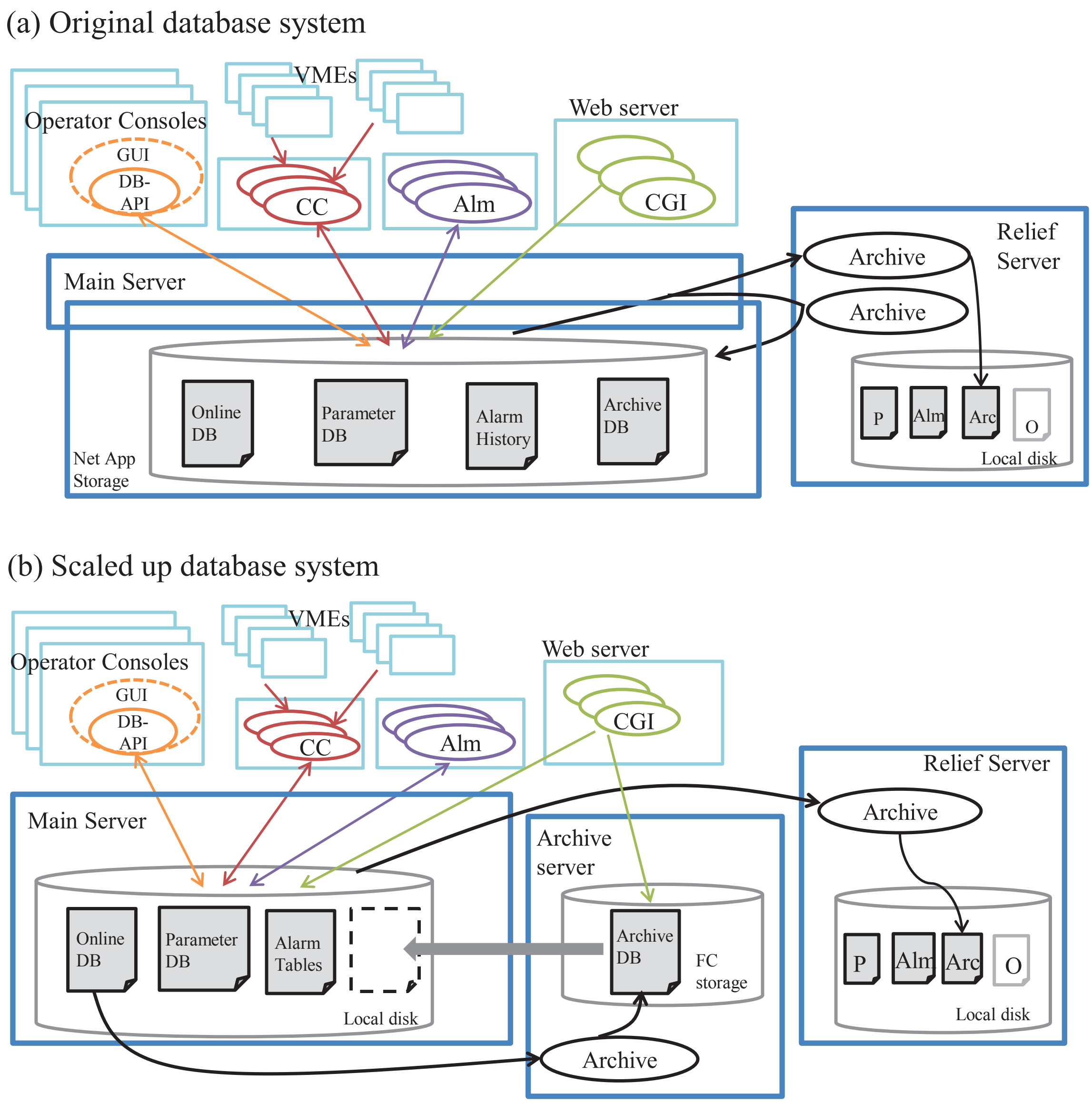

Figure 1: Schematics of the original (a) and scaled out (b) database system.

features of MADOCA. Processes called Collector Clients (CCs) collects data from VMEs and other control nodes. The time period for collection is typically 2 seconds, but can be as short as 1 second. The CC places a time stamp on the collected data and stores them in a small relational table called Online DB. Online DB holds data for only 8,000 seconds. Logged data is archived for future use in a larger table called Archive DB. The archive process also copies data to the relief server. Data logging configurations, such as period of data logging, are saved in Parameter DB.

We installed Online DB and Parameter DB on the main server so that CC could work without any changes. The scaling up was required to be realized with minimum changes to the client processes because the shutdown period of SACLA is limited to 1 month.

Archive DB, which contains logged data of the past, is rather large compared to Online DB and Parameter DB. We installed Archive DB on the archive server. In this way, the archive server is able to optimize the server parameters for large data block access. For example, the database page of Sybase ASE, which is the unit of disk access, was set to 8k byte while that of main server is 2k byte. A request for large data from Archive DB does not affect the main server. This stabilizes the response to the CCs, which require the server to respond in less than 1 second.

The third group of client processes of the MADOCA database system comprises alarm monitoring processes. An alarm monitoring process obtains the latest data from Online DB and compares it to thresholds in Parameter DB. If a value exceeds the threshold, the alarm process writes time and value data into tables called Alarm Tables. The alarm information is then displayed on a graphical user interface (GUI) according to the data contained in Alarm Tables. We installed Alarm Tables on the main server to avoid changing the alarm monitoring processes.

ISBN 978-3-95450-139-7

Logged data and properties of signals can be used in the GUI or other processes for accelerator operation by the database application programing interface (DB-API) with C source code. DB-API could be connected to only one server. Because many GUIs have already been developed, we avoided making changes to DB-API. Instead, Archive DB was mounted on the main server using "proxy table" of Sybase ASE [5]. A request for Archive DB in the main server is transferred to the archive server. Data is then sent to DB-API through the main server. Although the response of the request is slower than that of a direct request, the request to Archive DB is made through a block of data, which the overheads relatively small.

The last group comprises CGIs for data browsing. Data contained in the database servers is viewable through web servers. Data browsing from CGIs is adopted mostly for data in Archive DB and Parameter DB. As the CGIs were written in Python, it was easy to connect the main and archive servers; hence, we modified the CGIs to access both servers.

INSTALLATION

We installed the new scaled-up database system to the control system of the SACLA accelerator. The installation, carried out during the summer shutdown of 2012, was completed on time. As the database system is vital to the MADOCA control system, we performed a test prior to installation, and the system performance was closely examined [6, 7]. In addition, the parameters of Sybase ASE were optimized by a Sybase engineer.

Fig. 2 shows the CPU usage of the original system and the scaled-up database system during a time study of the accelerator. CPU usage was calculated based on the ratios of time used by CPU busy or I/O wait. Only the CPU cores with Sybase ASE engine were considered.

Fig. 2 (a) and (b) cannot be compared directly because access through the DB-API in both systems varied owing to frequent changing of the GUI to realize new methods of accelerator operation. As a result, a 60% improvement was achieved by the scaled-up system. The scaled-up database system was approximately 20%. The scaled-up system consists of 8 cores with a 2.93GHz clock in the main server and 7 cores with a 2 GHz clock in the archive server. The original system only consists of 7 cores at 2 GHz. The CPU power of the scaled-up system was 2.7 times greater than that of the original. In addition, we used Net App for network storage in the original system. Net App is highly redundant and easy to maintain; it is particularly suitable for a small number of signals in MADOCA. Unfortunately, it consumed CPU power of the main server by reading and writing data over the network. On the other hand, local disk were used for the main server of the scaled-up system. This also helped to reduce the CPU load.

Since the CPU load was very high in the original system, some GUIs were restricted by specific conditions to keep the CCs stable. The CPU load of the scaled-up database system was kept at a minimum. No restrictions for the GUI were required to keep the CCs stable.

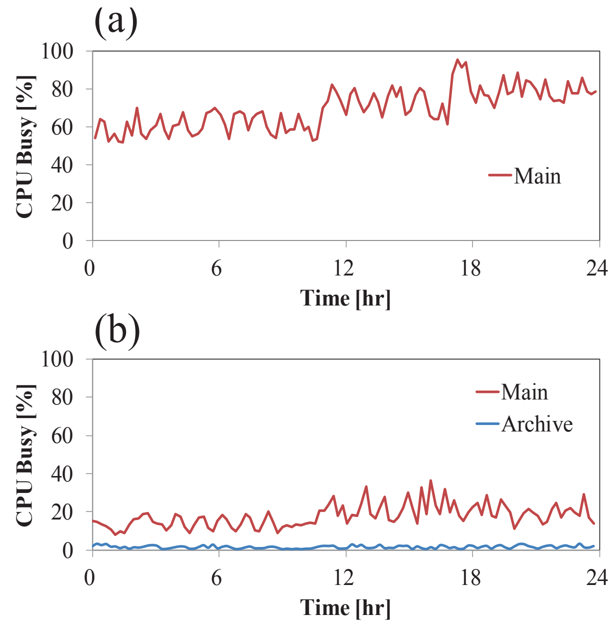

Figure 2. CPU usage during a machine study of the accelerator over a 24 hour period. (a) and (b) depict the original and scaled-up database systems, respectively.

SUMMARY

We designed and successfully installed a scaled-up database system. The system was designed to make CCs and other vital processes stable. The CPU load was kept at a minimum in the new system while the signals increased. Since the installation of the scaled-up database system, the database has operated efficiently without any lack of resources from the server.

REFERENCES

[1] R. Tanaka et al., "Control System of the SPring-8 Storage Ring ", Proceedings of ICALEPCS'95, Chicago, USA, 1995, p.201.

[2] R. Tanaka, et al., "Inauguration of the XFEL Facility, SACLA, in SPring-8", Proceedings of ICALEPCS 2011, Grenoble, France, 2011, p.585-588.

[3] A. Yamashita et al., "Data Archiving and Retrieval for Spring-8 accelerator complex", Proceedings of ICALEPCS '99, Trieste, Italy, 1999, p.434.

[4] http://www.sybase.jp/products/databasemanagement/adaptiveserverenterprise

[5] A. Yamashita, "Distributed database in SPring-8", Proceedings of ICALEPCS 2003, Gyeonginu, Korea, 2003, p.436-438.

[6] T. Hirono et al., "Load Test for the Database System of SACLA", Proceedings of the 8th Annual Meeting of Particle Accelerator Society of Japan, Tsukuba, 2011.

[7] T. Hirono et al., "MADOCA database system with dual servers for SACLA", Proceedings of the 9th Annual Meeting of Particle Accelerator Society of Japan, Osaka, 2012.

ISBN 978-3-95450-139-7