The first operation of control system at the SPring-8 storage ring

R. Tanaka, T. Fukui, K. Kobayashi, T. Masuda, A. Taketani, T. Wada and A. Yamashita
SPring-8, Kamigori, Ako-gun, Hyogo 678-12, Japan

Abstract

The control system based on the standard model of the UNIX and VMEbus system has successfully performed accelerator operation with 8GeV electron beams. The software architecture of the control system designed with the client/server scheme has well operated over the distributed computing environment with the FDDI optical-fiber network. The relational database system has been widely used for machine data acquisition, archiving, and management of equipment parameters. The results of the control system from the experience of the storage ring operation are summarized.

1 Status of SPring-8

The SPring-8 storage ring, a third generation light-source, has been started its beam commissioning since March 13th, 1997. On March 25th, the 8GeV electron beam was captured by the RF system, and the first synchrotron radiation light from a bending magnet was observed on the next day. The commissioning of the storage ring with insertion devices has been continued with 20mA beam current until October 3rd. After the commissioning, the beam lifetime reaches up to around 70hours at 20mA. Now, the synchrotron light from ten beamlines consisting of seven insertion devices (1 wiggler and 6 undulators) and three bending magnets are available.

2 Control system

2.1 System construction

The control system has been constructed keeping the basic design reported at the last conference ICALEPCS'95 in Chicago [1]. It is based on the standard control model which is built on top of the distributed computing environment with industry standard equipments. For the network and front-end hardware system, it is summarized as follows:
- 100Mbps FDDI network with optical fibers,
- Switching hubs for the front-end optical fiber Ethernet (10Mbps) connections to FDDI,
- Firewall gateway for tight security,
- UNIX workstations (HP9000/700 series etc.) for consoles,
- VMEbus system for the front-end control,
- Commercially available VME boards,
- CPU board powered by PA-RISC, HP9000/743rt,
- Fieldbus(RIO) using optical link[2],

and for the software system:
- TCP/IP protocol for communication,
- HP-UX for console operating system,
- X-Mate as the GUI builder based on X11,
- Sybase SQL server 11 as the relational database management system(RDBMS),
- UNIX based real-time OS, HP-RT for VME CPUs.

The concept and architecture is kept, however some modification for the system are made as follows. A network firewall machine is newly added in order to keep tight security from the internet, and also it plays a role as an access control gateway to separate the control network from the laboratory public LAN[3]. The file server machines exporting executable files and configuration tables etc. are divided into two machines. The operation applications are developed on the R&D server machine connected to the public LAN, and files are copied onto the NFS server machine on the control network. Such an isolation guarantees the independence of the accelerator operation preventing the possible interference coming from software development activities, that is mis-operation, wrong RPC calls and so on.

3 Software system

3.1 Client/server model

The software structure is designed by the client/server scheme as can be seen in Figure 1. The client/server scheme is promising to support Rapid Application Development (RAD) and it provides also the higher level degree-of-freedom for the software design. The framework consists of the Message Servers(MS), the Access Servers(AS) and the Equipment Managers(EM) are backbone server processes of the system. The set of the MS and ASs plays a role as the middleware of the local communication on operator consoles, and the AS can make multiple connections to the EMs running on the remote CPU boards. Some application needs to work synchronously or asynchronously according to the control sequence of the front-end equipments. The middleware framework provides both way for applications by the local process communication with System V message and remote process communication based on the ONC/RPC. The former provides asynchronous access as the Message Oriented Middleware(MOM) and the later allows synchronous communication over the network. The queuing mechanism is one of the best way to realize co-existence of the control processes with minimum interference because the most of processes are working asynchronously on the same workstation. Application programmers can build suitable access sequence at any level of the scheme.

3.2 Control message distribution

A control messages is formed to a 255 man-readable

character string as the English like syntax S/V/O/C, ex. “123_tanaka_oprgui_oprcon2/get/sr_mag_ps_st_h_1_2/curr ent_dac”. The MS is an implementation of the MOM and plays a role of message distribution to the AS processes via a queue. The queue has enough depth to be able to store more than 260 messages at a time. The message transaction speed on a UNIX workstation is found to be so fast(<1msec) that there is almost
no message stacked on a queue in average. A typical

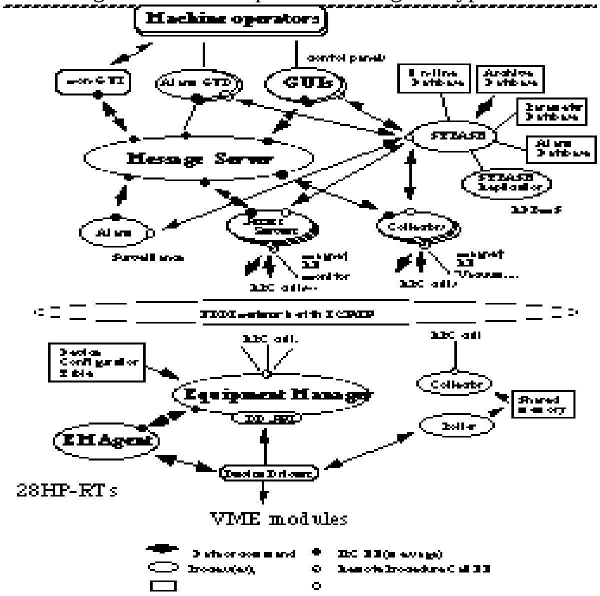

Figure 1. The software structure of the storage ring. The EMA feedback process is added.

communication takes about 10msec for a round trip between the MS and the simulated EM (no real device access) including RPC overhead. For the real device, hardware access time depends on the front-end controller(for a slow controller it takes more time).

An operation program(ex. man-machine interface) sends a message and receives a reply from the equipment. There are three waiting methods for a reply receiving, that is, no-wait, wait and time-out(2sec) modes. In the no-wait mode, an application tries to receive a reply then returns immediately if there is no reply message. Of course, for the wait mode, it is available to wait a reply until it comes back. Application programmers are able to send messages to the MS by calling the minimum number of C language functions provided as the application program interface (API) like ms_open(), ms_close(), ms_send(), ms_recv() and so on. The connection to the MS is dynamically and there is no static compile/link phase. The plug&play connection scheme is suitable for the development of the large scale software because the integration of application software to the control framework is usually performed with different time schedule.

3.3 Equipment access

The EM is a RPC server process[4]. It accesses VME devices and fieldbuses by translating the SVOC object message to the proper board or channel number(=O) with the requested action method(=VC). The lowest layer of the device access is performed by device drivers. The device drivers are encapsulated by wrapping low level functions with the API layers[5]. The EM is implemented by using driver API as the building blocks. The performance of the EM is good enough for an equipment access in object by object way. It takes 1msec for one VME action, and overall execution time form the GUI to the EM round trip takes 10-30msec including the network overhead and the reply event handling.

3.4 Equipment Manager Agent

There are requirements to achieve faster feedback or monitoring on the VME system which has no network communication. It is the consequence to develop an additional software process which can be used to realize the faster device access. It is called as the Equipment Manager Agent(EMA) process[6]. The EMA has the same software structure as that of the EM because it is implemented by using the EM framework. The EMA process is controlled as a virtual device which can be created or destroyed by the EM. Once it is created, it continues to run until destroyed by the EM. The performance of the EMA is measured to be 2msec for one VME access. For example, the EMA is used to ramp up the klystron voltage by watching its vacuum status as a feedback input.

4 Data acquisition

For the monitor and logging purpose, status of equipments are taken periodically by the data acquisition processes. The poller/collector system is implemented using the same framework as that of the EM[1,7]. The pollers are the stand-alone processes, on the other hand, the collector processes are either client or server processes on the client/server scheme. The one set of data coming from the storage ring and beamlines reaches up to 5600 signal points which becomes about 1GByte/month[8].

4.1 Cyclic data acquisition

Accelerator data such as klystron power, vacuum gauges, magnets current etc. are taken in every 2-5seconds, and only the current of the electron beam is taken in every second. The data is stored onto the database as the on-line data. The status monitoring processes like GUI’s with graphical plots get data from the on-line database with common data access method. Alarm surveillance system is also getting the data from the on-line database by comparing the pre-defined threshold to the equipment status[9]. The on-line database reduces unnecessary network transaction caused by the overlapped data access to the same equipment. The data is sampled by 60sec period and archived onto the database permanently.

5 Beamline control

5.1 Beamline control system

The beamline control system is constructed with the same architecture as that of the storage ring control. It consists of one UNIX workstation and two VMEbus system, that is, one is for an insertion device(ID) and another for front-end and transport channel. The interlock network is built with the Programmable Logic Control unit(PLC) for the purpose of human safety and equipment protection.

As can be seen in Figure 2, beamline workstations and VMEbus system are directly connected to the machine control network via routers. On the other hand, user PC's or workstations for the experiments are isolated from the beamline control network. Only the serial connection by the RS232C is supported at this moment. In order to keep the system security from the internet, beamline workstations are isolated from the laboratory public LAN as well. The benefits of interconnection of the network between the machine and the beamline is the direct communication between the accelerator and beamlines. Especially for operation of insertion devices, an equivalent control either from the machine control or the beamline control is available under the certain grant. It allows beamline-user oriented operation of the insertion device after its commissioning, and also it makes synchronous operation of the insertion device and the experiment detector system possible.

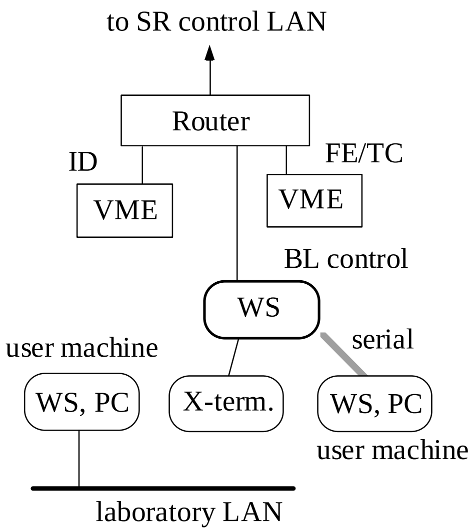

Figure 2. The beamline control system.

5.2 Beamline data acquisition

It is the benefits of the same control architecture between the storage ring and beamlines that the poller/collector system can be exported to the beamlines as well. The data from the ID system, beam position monitors(BPMs), vacuum gauges of the front-end transport line, and steering magnets for the beam correction are taken with 10sec period and stored onto the machine on-line and archive database.

It is important to make the correlation analysis between the data of ID gaps, rfBPMs, and the closed orbit distortion(COD) of the electron beams in order to understand the beam dynamics around IDs. The common data structure and access method has good advantage for such purpose.

6 Development

6.1 Conditions

Because of the limited schedule of the development, there are some initial conditions to be concerned about the choice of the programming language, builder tools, development methodology and so on. The most of the accelerator physicists and equipment specialists are Fortran or Basic language users. For the client/server scheme with UNIX environment, it is natural decision to use the C language under such a condition and the RAD method is taken as the essential methodology.

6.2 Methodology

At the first stage of the development, machine operation sequence is summarized by scanning over requirements from equipment specialists. The overall scheme of the data flow and control sequence is analyzed with the Structured Analysis(SA) at the beginning. The software structure is determined considering data flow and control sequence on top of the client/server scheme. Once the structure is determined, the software is developed with the spiral ptototyping method. Especially for the client/server scheme, it is necessarily to repeat analysis and development phase several times. It is quite specific for the client/server software that there is ambiguity about the detail of the software specification at the beginning. However, software development can be started with prototyping method and the contents of the software can be gradually fixed by repeating its development cycle. It is very promising that after some spirals development is proceeded by the waterfall method to come to finish. The prototypes are useful to make ambiguity to be concrete in the middle of the development phase, and the rapid prototyping is good enough to confirm the system availability at the early phase.

6.3 Construction

The software development is divided into two parts, that is, one is the development of the control framework software (core part) and the other is the equipment application software. The control group (equipment group) is responsible for the framework (the applications), respectively. At the first stage of the software development, the core part was analyzed and designed to be robust and stable all over the system, and well established in order to be able to develop applications with the minimum knowledge about the core part.

The most important process, the determination of SVOC object list is proceeded spending enough time by consulting equipment specialist object by object. It takes typically two months to make the first version then takes one month to finalize it for each equipment group. The software development takes 20 man-months for the framework and 30 man-months for the application. The beam control software of the accelerator operation is developed by machine physicists which is not included for the equipment controls.

7 System reliability

7.1 Hardware system

The VMEbus system has been working well with a few minor board troubles. A fieldbus connection to the VME system should be paid attention if it needs lock/unlock mechanism or has any service request, i.e. GP-IB is the case. It limits the software performance and design freedom in some level, and the system transparency becomes worse.

The network system built with LAN switching technology is fine and its performance is satisfactory[3]. The switching hubs connected to the FDDI provides Ethernet full-bandwidth to each workstation port without collision. The firewall guarantees tight security and provides access control freedom at the application layer. It is different from as that of simply supported by a router or workstation gateway.

7.2 Software system

The control software should be stable at any time during accelerator operation. Even if it is in trouble, the damage should be minimized and localized over the system. The distributed computing environment is tough against the trouble. Because it provides redundancy such as multi-consoles can be running equivalently with the same purpose and VME systems of the RF or the magnet etc. are running independently. It is available for the maintenance and modification with minimum interference. For example, VME system trouble is possible during the accelerator operation, however, troubles can be minimized because it is necessarily to restart the troubled VME only with the rest of the system kept running. The software is designed to support such a function. The client/server software scheme is found to be robust, and provides redundancy to the system at the higher level without loosing system availability.

8 Expandability

The integration of the large number of software processes to the large software system is problem and any modification after the integration is difficult. The software architecture like plug&play is flexible for the system upgrade or configuration change of applications. The interference to the framework by connection or disconnection of the application software can be minimized. It is realized by using the small number of well defined API functions. The connection to the MS is one of the examples.

9 Data accessibility

Accessibility of the data is crucial for data utilization. The relational database management system(RDBMS) is powerful to store the accelerator data and useful for its management. Most of the data, like vacuum gauges or the COD of the beams etc. are non-object data different from that of the image or voice. The RDBMS provides the standard access scheme to the data by the C functions API through the ct library provided by Sybase. One of the benefits of the commercial base
RDBMS is the availability to use C-coded API functions not only in house but also widely used free software libraries
based on SQL. It is essential to utilize the stored data by calling well organized C functions in the off-line and the on-line analysis with minimum knowledge of the database system. World-Wide-Web, for example with the Netscape browser, is very useful as the man-machine interface. It is used as the data access engine and presentation tool. For example, in the accelerator operation the status of equipments is monitored by point&click way through the browser.

10 Summary

The control system of the storage ring has worked well. Its overall performance is found to be satisfactory. The client/server scheme is robust and early backup from the trouble is available. It provides redundancy and prevents the serious breakdown of the system. The data access is performed with uniform way using SQLs either from application programs or WWW browsers. The RDBMS is good enough as the data management system and its performance is satisfactory.

References

[1] R.Tanaka et al., Proc. of ICALEPCS'95, Chicago, USA, 1995, p. 201.

[2] H. Takebe et al., Proc. of the 4th European Particle Accelerator Conf., London, June 1994, p. 1827.

[3] T.Fukui et al., “Design and performance of the network system for the storage ring control at SPring-8”, in these Proceedings, ICALEPCS’97, Beijing, China, 1997.

[4] A.Taketani et al., Proc. of ICALEPCS'95, Chicago, USA, 1995, p. 625.

[5] T. Masuda et al., Proc. of ICALEPCS'95, Chicago, USA, 1995, p. 584.

[6] A. Taketani et al., “Medium-speed feedback software based on the existing control system”, in these Proceedings, ICALEPCS’97, Beijing, China, 1997.

[7] A. Taketani et al., “Data Acquisition System with Database at the SPring-8 Storage Ring”, in these Proceedings, ICALEPCS’97, Beijing, China, 1997.

[8] A. Yamashita et al., “The database system for the SPring-8 storage ring control”, in these Proceedings, ICALEPCS’97, Beijing, China, 1997.

[9] A. Yamashita et al., “The Alarm System for the SPring-8 Storage Ring”, in these Proceedings, ICALEPCS’97, Beijing, China, 1997.

Filename:	VVP097.DOC
Directory:	D:\NPAPER97\JRY
Template:	C:\WINWORD\TEMPLATE\NORMAL.DOT

Title: The first operation of control system at the SPring-8 storage ring

Subject:	
Author:	JIANG MINGBAO
Keywords:	
Comments:	
Creation Date:	01/19/98 4:56 PM
Revision Number:	2
Last Saved On:	01/19/98 5:17 PM
Last Saved By:	JIANG MINGBAO
Total Editing Time:	2 Minutes
Last Printed On:	01/19/98 5:19 PM
As of Last Complete Printing	
Number of Pages:	5
Number of Words:	2,765 (approx.)
Number of Characters:	15,766 (approx.)