DEVELOPMENT OF PC BASED FIELD CONTROLLER WITH LINUX

T.Masuda, T.Fukui, M.Kodera, R.Tanaka and A.Yamashita
SPring-8, Mikazuki, Hyogo, 679-5198, Japan

Abstract

In SPring-8 storage ring control system, VMEbus systems with real-time UNIX(HP-RT) are used as front-end controllers. For increasing requirements of temporary measurements, it is useful to introduce inexpensive I/O controllers. Linux based PC system was adopted as a supplement system with the same functionality as that of the VMEs. Control software running on the VME system was migrated to Linux system and the new system was used for such as measurement of cooling water temperature, air temperature at the machine tunnel and so on. At this moment, four PC systems are actually used as field controllers. We report the development of the PC based field controller with Linux and measured performance based on our control software.

1 INTRODUCTION

According to the standard model of accelerator control system, VMEbus systems have been used for the SPring-8 storage ring(SR) control system[1]. The CPU board is HP9000/743rt with 64MHz PA-RISC 7100LC and its operating system(OS) is HP-RT which is real-time UNIX based on LynxOS. The VMEs are connected with 10Mbps optical fiber Ethernet and linked to 100Mbps FDDI backbone LAN through a FDDI-Ethernet switching HUB[2]. Currently, 27 VMEs are distributed around the SR to control equipments such as magnet power supplies, vacuum system, and RF system.

After machine operation started, requirements of temporary measurements mainly for machine studies have been increasing. The data sources are distributed around the SR and the number of data points of each source is small. In order to investigate the correlation with other data, it is necessary to store those data into database. Since the temporary measurements should not request the major configuration change of VMEs used for the machine operation, we introduced PC based field controllers with Linux as a supplement system of the VMEs within the same control software framework.

2 CONTROL SOFTWARE

Our control software is based on client/server scheme as shown in Figure. 1[3]. In order to install for PC based field controller to the frame, we have to migrate those processes such as Equipment Manager(EM), poller, and Collector Server(CS).

2.1 Main Framework

The main framework consists of Message Server(MS), Access Servers(ASs) and EMs. All application programs running on operator consoles to control equipments send control messages to MS. After checking access control, the MS sends the messages to the appropriate AS which is a client process of ONC/RPC. The AS sends the messages to the server process(=EM) running on each VME system. The EM operates the VME boards according to the received message and replies the execution result to the client. The AS returns the result returns to the application program through the MS.

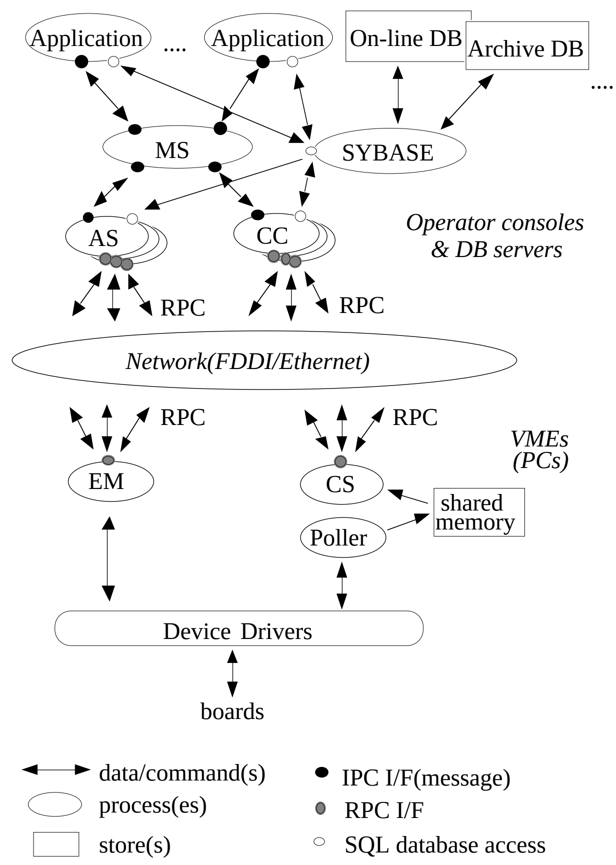

Figure 1. The control software structure.

2.2 EM

The EM is an RPC server running on the VME system[4]. It resolves the received message according to a configuration file and determines which VME boards and channels are allocated and operates the boards by calling appropriate control functions.

2.3 Poller/Collector Data Acquisition System

For the purpose of monitoring and logging of the equipments status, periodically data acquisition system called poller/collector is adopted[5].

The poller is a stand-alone process running on the VME. Since it was implemented with the same framework of the EM, it takes data by using the same control command as that of the EM. The data taken by the poller is stored in ring buffer implemented on shared memory which is used for communication between the poller and the CS. The CS which is an RPC server process on the VME collects the newest data from the ring buffer. The CS sends the data according to a request of the collector client(CC) on the operator console. After receiving the newest data, the CC sends them to the on-line database.

3 DEVELOPMENT OF PC BASED FIELD CONTROLLER

We developed the PC based field controller realizing the almost same functions as those of the VME system by migrating our control software to Linux.

3.1 OS

Because our control software is built on UNIX system, OS for the PC controller must be based on UNIX. Our candidates were Linux and Solaris. As far as considering the complete migration of the EM, the poller, and the CS, Solaris for the real-time OS was seems to be preferable to Linux which has no real-time features. Because they were written with some real-time functions of HP-RT such as priority control and real-time timer. However, the most important and serious problem was how to prepare device drivers for I/O boards, especially for GP-IB interface boards. Since some free device drivers for I/O boards including the GP-IB were available for Linux, consequently we decided to adopt Linux. At this moment, the real-time parts of our control software was not so critical for the temporary measurements.

Version 2.0.30 kernel was installed to field controllers with Slackware 3.4.0 distribution.

3.2 PC Hardware

As the PC hardware, DELL Optiplex series with mini-tower chassis was prepared because we needed as many ISA slots for I/O boards as possible. It can provide maximum 4 ISA slots and has a 333MHz Pentium II CPU, 64MB main memory, 4.3GB hard disk, and a 10/100Mbps Ethernet Interface.

3.3 Boards and Device Drivers

For our temporary measurements, three types of ISA boards were required, that is, GP-IB interface board for controlling instruments, analog input board for measuring temperature with thermocouple, and TTL level digital input/output board to control CATV switch.

As the GP-IB interface board, AT-GPIB/TNT from National Instruments was selected because its free device driver was available[1]. For the analog input board, AISA2606 from Advanet Inc. was used since it could be considered as the ISAbus version of Advme2602 VMEbus board which was already introduced to temperature measurements. This board has 4 channels for wide range of analog input and thermocouples can be connected directly to it. For TTL level digital input/output board, 32-bits AISA1422 from Advanet Inc. was adopted.

We developed Linux device drivers for the later two boards by ourselves with the help of our experience in HP-RT. The both were very simple I/O boards, so that writing drivers was not so difficult. From the experience of the development of simple drivers, the basic differences between Linux and HP-RT drivers are summarized as follows.

- For board register access, we have to use service calls according to the data size and the data direction(read/write) with an argument of hardware address in Linux device drivers. On the other hand, HP-RT allows to use a memory pointer to the virtual address translated from the hardware address at the driver installation.
- To transfer data between user and kernel space, Linux requires to use service calls with giving the address of user space buffer as an argument. But HP-RT handles this transfer by itself.
- Loadable module function supported on Linux is not available on HP-RT.

Loadable module function helps to reduce development and debugging time.

3.4 Migration of Control Software

Since our control software was written with UNIX System V system calls and POSIX system calls, the migration from HP-RT to Linux was really easy. We succeeded to migrate the EM, the poller and the CS except the real-time parts of HP-RT described below.

- In order for the EM to respond operators' demands as soon as possible, the poller should run at lower priority than the EM. The CS on HP-RT realizes this by using priority control function which Linux does not support.
- For providing high resolution(~200µs) sleep function, the EM on HP-RT uses timer functions handling hardware timer.

About the former, we replaced the priority control function of HP-RT with nice value control function of Linux. And about the later, we did not realize such a high resolutuion sleep function on the EM on Linux.

Another main modification was to change stub function names between Linux and HP-RT generated by rpcgen command.

---

[1] ftp://ftp.llp.fu-berlin.de/pub/linux/LINUX-LAB/IEEE488

By using XDR format of ONC/RPC, we didn't have to consider the endian difference between PA-RISC and Pentium II.

4 APPLICATION

The PC based field controllers have been applied mainly to the measurement of cooling water temperature for the SR magnets. For this measurement, three PCs are used and they have three AISA2606 boards with 7 thermocouples inputs and 3 another level of analog inputs. The poller is running on the each PC to take temperature data periodically and put them on shared memory. By receiving request from a CC with 5 seconds interval, the CS collects the newest data set from the shared memory and sends them back to the CC to store those data into database.

Also a PC controller has been used to control CATV switches with a AISA1422 board and to get the data from a time interval counter with a AT-GPIB/TNT board. The EM manages the operator's demand to switch the CATV, and the CS collects and sends the interval data taken by the poller to an appropriate CC.

Above application systems had been working with no troubles for three months during the machine operation.

5 PERFORMANCE

For the comparison of performance between the PC based controller and the VME, we measured some execution times of our control software. PC and VME system used for the measurement of cooling water temperature were selected as the test system. The VME system consisted of two Advme2602 boards occupied with the same number of inputs as that of the PC. Both controller were connected with 10Mbps Ethernet.

Table 1 shows the average processing times of one control message to get one temperature value in each EM. The board access time with the device driver shown in Table 1 is included in the whole processing time. By comparing the whole processing time, the PC controller marked about 5 times faster than the VME controller. Even the board access time of PC through the slower ISA bus is about 2.5 times faster than that of the VME system.

Table 1: Processing time in EM

	PC controller	VME controller
processing time of one control message (μs)	239	1225
board access time with device driver (μs)	5	14

Table 2 shows the average data collection time of the CC which takes the same size of data from the each CS through the network. The PC controller is about 3 times faster than the VME controller.

Table 2: Data collection time of CC

	PC controller	VME controller
whole processing(μs)	1361	3924
data collection in CS(μs)	16	92
transfer to XDR in CS(μs)	24	130
transfer from XDR in CC(μs)	35	31

6 SUMMARY

The PC based field controller with Linux was developed based on the SPring-8 control software as the supplementary system of the VMEs. It was easy to migrate the control software from HP-RT to Linux without the real-time features. The PC controllers was actually applied to the measurement of cooling water temperature of the SR magnets and so on. They have been working well and are very stable. The performance of the PC including network access is about three times faster than that of the VME, on the other hand the cost is less than about one sixth. The PC field controller with Linux provides very powerful and economical solution to the temporary measurement or control over the network.

As the next step to use real-time feature, it is interesting to intend the system to introduce real-time Linux.

REFERENCES

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

[2] T.Fukui et al., "Design and Performance of the Network System for the Storage Ring Control at the SPring-8", Proc. of ICALEPCS '97, Beijing, China, 1997, p.312.

[3] R.Tanaka et al., "The First Operation of Control System at the SPring-8 Storage Ring", Proc. of ICALEPCS '97, Beijing, China, 1997, p.1.

[4] A.Taketani et al. "Equipment Manager of the VME Control System for the SPring-8 Storage Ring", Proc. of ICALEPCS '95, Chicago, USA, 1995, p.625.

[5] A.Taketani et al. "Data Acquisition System with Database at the SPring-8 Storage Ring", Proc. of ICALEPCS '97, Beijing, China, 1997, p.437.