PERFORMANCE EVALUATION OF THE REMOTE EQUIPMENT USING DISTRIBUTED REMOTE OBJECT FOR THE 1.8-GEV TSRF STORAGE RING

Nobuyuki Kobayashi, Noriichi Kanaya, and Yutaka Tahara,
Department of Electrical and Electronics, Faculty of Engineering, University of Ibaraki, Hitachi, Ibaraki, 316-8511, Japan
Shoji Suzuki, and Shigeru Sato, Department of Physics, Graduate School of Science, Tohoku University, Sendai, 980-8578, Japan

Abstract

The control system for the TSRF positron storage ring is comprised of distributed computers and equipment such as analogue-digital converters, digital input/output modules. TSRF (Tohoku Synchrotron Radiation Source Facility) is a new third generation synchrotron radiation source that is planned to be constructed at Tohoku University. We have developed a software interface called the remote equipment in Java to be used for the control system in order to access the physical devices on the network. In this paper, the performance of the remote equipment is discussed.

INTRODUCTION

TSRF (Tohoku Synchrotron Radiation Source Facility) is a new third generation synchrotron radiation source that is currently proposed at Tohoku University Japan[1] [2]. TSRF is planned to be constructed at the site of Laboratory of Nuclear Science, Tohoku University, where a 300MeV-Linac and 1.2GeV Stretcher Booster Ring are currently in operation for nuclear physics experiments [3]. By taking advantage of the existing facility, TSRF employs the Stretcher Booster Ring as the injector for the TSRF storage ring. This can greatly reduce construction cost for the TSRF generation synchrotron radiation source. TSRF has a 1.8GeV storage ring with a DBA(double-bend-achromat) type, third-generation storage ring with emittance of 4.9nm·rad, and a circumference of 244m for Soft X-ray and VUV experiments for research.

Control systems for synchrotron radiation sources and high energy accelerators are comprised of many computers and equipment such as analogue-digital converters, digital input/output PCI modules. The control system for the 1.8GeV TSRF synchrotron radiation source controls the storage ring, providing stable synchrotron radiation to users at the experimental hall, and improves the performance of the storage ring. PCs are cost effective equipment as distributed hosts. A remote interface in software is necessary for the control system to access accelerator components.

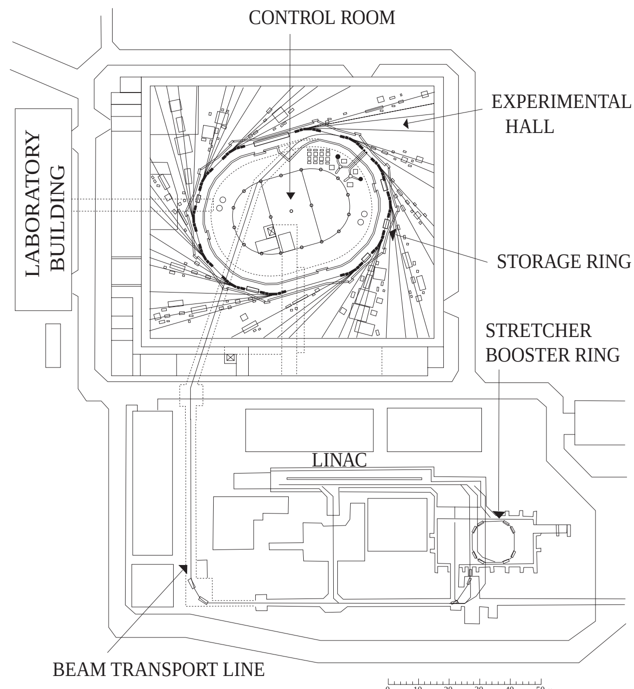

Figure 1: 1.8GeV Synchrotron Radiation Source at TSRF

Since the construction cost for the TSRF storage ring is severely limited we employ Linux as the operating system for control servers rather than a commercial operating system. A few consoles are running under a commercial operating system because of users' preference. In addition, some VME crates are replaced with PCs to reduce the cost.

Remote equipment interface has been implemented using Java Remote Method Invocation (RMI) running under the distributed platforms on the network. Since for time critical operation, however, it has been assumed that the Java garbage collector would impose unexpected delay time into control programs. To avoid such delay, we have ever implemented time critical programs in another language for the first time. However, it is preferable to implement entire system only in Java to keep the system portable as much as possible. We have measured and evaluated the performance of a control system implemented using the remote equipment in Java (v.1.31). The performance of the remote equipment is evaluated, and described for one of control systems for the 1.8GeV TSRF synchrotron radiation source.

SYSTEM CONFIGURATION

Figure 2 shows the block diagram of the remote equipment. Java programs run on the Virtual Machine (VM) that provides homogeneous environment on different platforms independent of their operating systems and hardware architecture. Although the VM runs under any operating systems such as Linux, Solaris, and Windows, the VM has no direct interface to physical devices which are tightly implemented upon specific operating systems. Thus those physical devices are not accessible to control applications. The interfaces to physical accelerator components such as digital I/Os, ADCs and DACs connected to magnets and beam position monitors, are platform-dependent upon each specific operating system. For Linux platform (kernel 2.2), we have developed interface drivers for PCI modules using the Java Native Interface (JNI) [4] through which Java classes can access the physical devices.

Unlike the RPC model, it is not necessary for the RMI model to take the data structure of the arguments and their length into consideration. These arguments have to be exchanged among clients/ servers across the network. This results in costly implementation procedures.

Using RMI, arguments are passed by serializing argument-data into a byte-stream data, and then transmitted it to the client requested. Transmitting a set of control messages from a client to a server is also done in the same way. The stub is the proxy allowing control clients to invoke remote methods and carries out marshaling of their arguments. The skeleton waits calls from a client, and marshals parameters and finally passes them to the associated methods of a server. The transport layer establishes connections and deals with data from/to remote servers.

We have implemented the consoles (RMI clients) under Linux and Windows 2000 with a large display, and a pointing device. Man-machine interfaces, including graphic status displays providing menu driven interface are also coded in Java Swing graphics class libraries. When a console executes a remote method of a remote server, its action is delivered through the network to the skeleton of the server. The skeleton marshals parameters to the actual method in the server. And finally the actual remote method is executed at the server side.

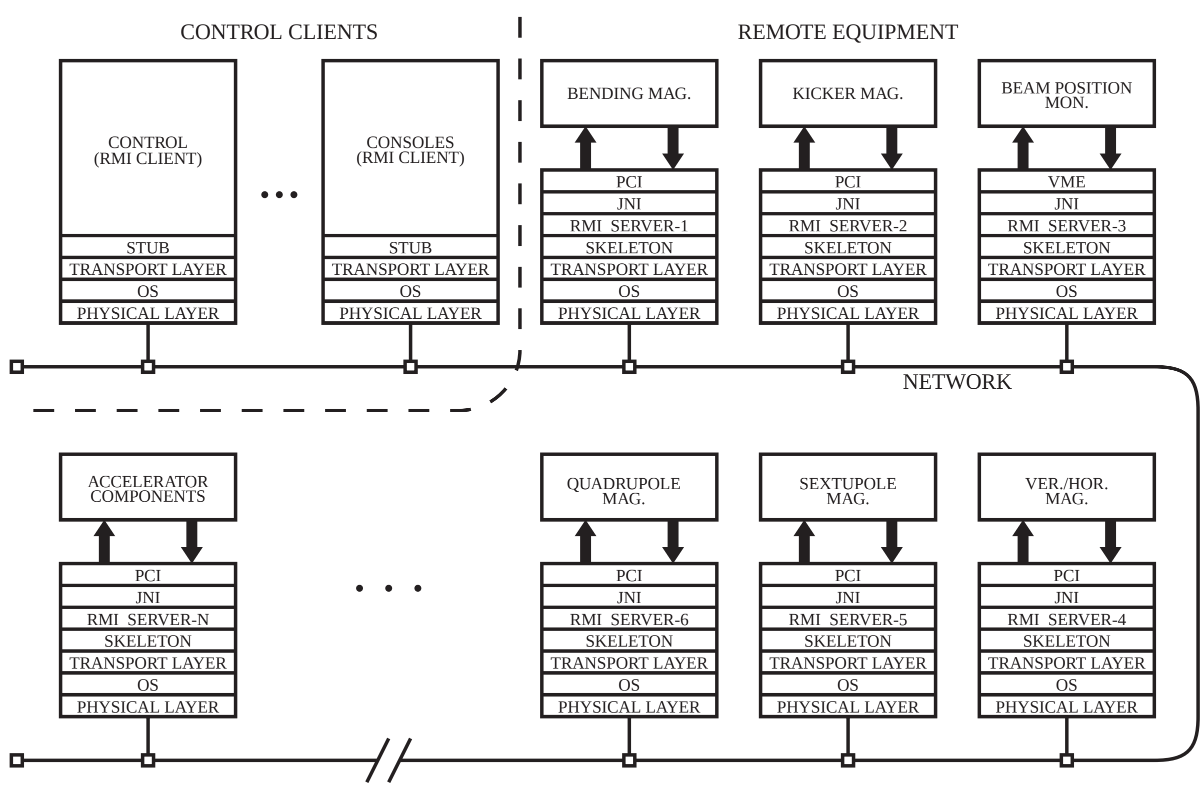

Figure 2: Remote Equipment using RMI for the 1.8GeV Synchrotron Radiation Source at TSRF

There is a registry server (not shown in Fig.2) that is running as a background process, and it allows control programs (clients) to locate a remote server. During bootstrap, a server asks the registry server to register or bind its name in order to make the methods of the server available to all control programs. This name is queried by a client to locate the server. Once the control program locates where the remote equipment and its related methods are, it can access the remote equipment. Then the control program carries out the remote method as if it were a local method to control the remote device. Thus any accelerator equipment on the network is transparent to the control programs.

As shown in Fig. 2, PCI modules are connected to the bending magnets, quadrupole magnets, beam position monitors, Octupole magnets, sextupole magnets and other accelerator components. PCs are connected to a 100-Mbps network during implementation and test phases.

The magnet control system has been designed using the remote equipment [5]. The system has 27 PCs to control 224 magnets, and one PC for a console (client) that has to setup DC current values to all servers within 50msec. Thus, the minimum response time expected for the console to set up a DC current to a magnet on a PC must be less than 1.85 msec (=50msec/27PCs). In the next section, measured the response time between a console (client) and a server is described.

MEASUREMENTS

The response time is defined by the time expired from the time when the console (client) executes a remote method of the server to the time at the reception of the argument reply from the server at the console., *i.e.*, it is the execution time of a remote method at the server. Using an oscilloscope connected to a parallel digital-output PCI module at a remote magnet-control server, we have measured the response time when the console sets a digital value, which corresponds to a DC current value in the magnet, onto the PCI module. Continuous measurements have been made to obtain average response times for different platforms and network loads as shown in Table 1. The graphics user-interface at the console was disabled to obtain actual response times during the measurements. The results were reproducible.

Table 1 Measured average response times between the console (client) and a remote server for the magnet (msec)

console(client)	server	
	Linux A (1.2GHz)	Linux B (800MHz)
Linux A (1.2GHz)	-	0.459
Linux B (800MHz)	0.425	-
Windows2000(2.5GHz)	0.314	0.387

The average times measured are less than 1.85msec that suggests that the remote equipment can satisfy the minimum response time required for the magnet control system.

An additional function or callback has been implemented for the PCI parallel digital I/O module. The console does not need to suspend until a remote method completed at the server side for the magnet control system. Upon completion of the remote method, the console is notified of it from the server and receives the result. The average response time measured for continuous callbacks is approximately 4 msec under the same measuring conditions as those in the previous measurements.

These results show that the remote equipment implemented can provide sufficient response time for the magnet control system to be operated for the 1.8GeV synchrotron radiation source at TSRF.

CONCLUSION

The remote equipment deals with PCI modules on remote Linux control system. The measured performance of the client/severs using Java RMI is described for the TSRF 1.8GeV storage ring. Note that before actual commissioning phase for the TSRF synchrotron radiation source, the network will be replaced with a latest, i.e., faster, FDDI (Fiber Distributed Data Interface) with token-passing, dual-ring network using a fiber-optic link suitable for exchanging control messages.

ACKNOWLEDGEMENT

The authors wish to express their gratitude to the project members at Department of Electrical and Electronics, University of Ibaraki for their assistance and cooperation, and to Prof.J.Kasagi, Laboratory of Nuclear Science (LNS), Prof.M.Watanabe, Research Institute of Scientific Measurements, Tohoku University for their valuable discussions.

REFERENCES

[1] S.Suzuki, M.Katoh, S.Sato and M.Watanabe, "Design of a Storage Ring Light Source at Tohoku University," Nucl. Instrum. Meth. Vol.A467-468,pp.72-75,2001.

[2] M.Katoh, S.Sato, S.Suzuki and T.Yamakawa, "Lattice Design of the Synchrotron Radiation Source at Tohoku University," Proc. of the 5th European Particle Accelerator Conference,Spain,June,1996.

[3] M.Oyamada et al., Proc. of 10th Symp. on Accel. Sci. Tech. p463, 1995 (in Japanese).

[4] N.Kanaya, N.Kobayashi, S.Suzuki and S.Sato, "Design of the Magnet Control System for the 1.8GeV Synchrotron Radiation Source TSRF," in these proceedings.

[5] *Java Native Interface Specification*, Sun Microsystems, www.javasoft.com.