ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S08NC03

A DISTRIBUTED DESIGN FOR MONITORING, LOGGING, AND REPLAYING DEVICE READINGS AT LAMPF *

Mary Burns
Los Alamos National Laboratory
Los Alamos, NM, USA, 87545

Abstract

As control of the Los Alamos Meson Physics linear accelerator and Proton Storage Ring moves to a more distributed system, it has been necessary to redesign the software which monitors, logs, and replays device readings throughout the facility. The new design allows devices to be monitored and their readings logged locally on a network of computers. Control of the monitoring and logging process is available throughout the network from user interfaces which communicate via remote procedure calls with server processes running on each node which monitors and records device readings. Similarly, the logged data can be replayed from anywhere on the network.

Two major requirements influencing the final design were the need to reduce the load on the CPU of the control machines, and the need for much faster replay of the logged device readings.

I. INTRODUCTION

The Los Alamos Meson Physics Facility (LAMPF) linear accelerator, the injection lines, the experimental areas, and the Proton Storage Ring (PSR) contain over 16,000 beam line devices (approximately 11,000 in the LAMPF Control System (LCS) and 5,000 in the PSR control system) which can be accessed through the LCS Data System [1]. These devices include both monitoring and control devices which occur throughout the facility. Many of the devices are accessed through remote microVAXes which are connected by Ethernet to form a DECnet LAN with the main control computer, a VAX 8650. This system is becoming more distributed as remote computers are added to allow access to related groups of devices, and some of the CPU load is being off-loaded from the main computer to remote machines. The increased distribution of the system has made it necessary to redesign some of the software to take advantage of the changes. One such system is the software that monitors, logs and replays readings from devices located throughout the facility. Before the redesign of the software was started, a complete re-evaluation of the requirements of the software was carried out so that other improvements could be included in the new design.

* Work supported by the U.S. Department of Energy

II. FUNCTIONALITY OF THE SOFTWARE

2.1 The Sampling Software

A large number of beam line devices can be sampled periodically for the purpose of monitoring the device to determine if its reading is within a given range, logging the device reading to a history file, or both. The frequency at which each device is sampled can vary from once a second to once every eight hours. If a device which is being monitored is found to have a reading outside its given range (ie. it is out of tolerance) the accelerator operators are notified via the operator console and an error log. An alarm action may also be initiated. If the device reading is being logged to a history file for replay the reading is recorded together with a device identifier, a compressed timestamp, and the status of the read.

2.2 The Replay Software

The device readings can be replayed for a selected time period and displayed as graphs. Readings from several devices can also be combined in an arithmetic expression and plotted.

III. THE SOFTWARE DESIGN

The sampling software is divided into two parts; the program which collects the data and monitors and logs it, and the operator interface software which allows the operators to interact with the sampling program. Interactions include stopping and starting the sampling of individual devices or lists of related devices, changing the parameters associated with a device, such as the frequency of sampling, and requesting an immediate monitoring of a set of devices.

3.1 Problems with the Original Design

Originally the sampling software was designed to run on the main control computer and to monitor and log data from throughout the facility in a large, central, history file. Replay of device readings was carried out on the same computer.

As more nodes were added to the network the demand for distributed replay grew. Also the number and frequency of device sampling increased, so that the load on the control computer became more noticeable causing a degradation in the speed of execution of other applications, and affecting the ability of the facility operators to respond quickly to requests from experimenters or to emergencies. Due to the increasing size of the history file, replay of the device readings was also

3rd Int. Conf. Accel. Large Exp. Phys. Control Syst. ICALEPCS1991, Tsukuba, Japan JACoW Publishing
ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S08NC03

becoming slower and consuming too much of the control computer's CPU and the operators' time.

3.2 An Interim Solution

One way in which replay has been improved is by the development of command procedures which run a version of the replay program as batch jobs at regular intervals. Instead of outputting the data to a graphics screen the device data is written to a file which is then sent to a laser printer. This automates the replay request process, and allows the replay to be scheduled at times of lower CPU usage so that there is less impact on beam production. However this is only a partial solution.

3.2 Requirements for the New Software

The requirements for the redesigned software include the ability to control the sampling of device readings from a number of nodes on the network, and the ability to replay device readings from any node on the network. It was also necessary to reduce the CPU load on the control computer for acquiring and logging device readings, and to increase the speed of replay of the device readings. The last two requirements were met by redesigning the history file. The software redesign was accomplished in two steps. The first step was to develop a distributed system, and the second was to redesign the manner in which the history file is shared between the logging and the replay software.

3.3 The Redesigned Distributed Software

As many of the devices are read through remote nodes it was decided that there are advantages in storing device readings on the local node to reduce network traffic and the load on the main control computer. The design of the software that interfaces to the sampling program calls for a user interface which can be run anywhere on the network. The user interface

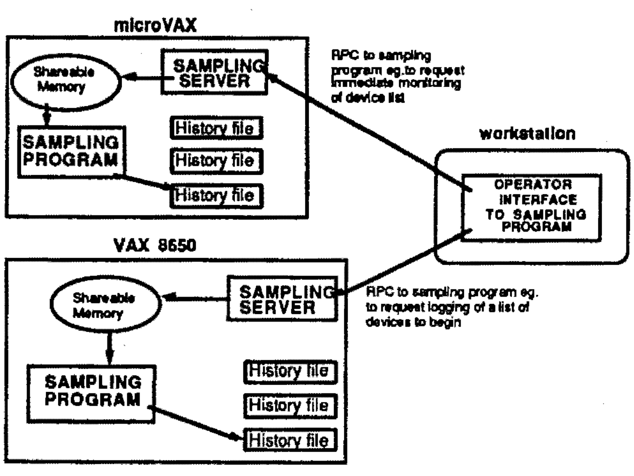

Figure 1. Diagram showing commands being sent to sampling programs on several network nodes, from a remote workstation.

communicates across the network with servers running on each node which is sampling device readings. Communication between the operator interface and the server is done via remote procedure calls (RPC).

An example of the way in which the distributed system of sampling software is used is shown in Figure 1. Sampling programs running on a microVAX and a VAX 8650 can be controlled from a workstation connected to the same network. Using the operator interface to the sampling program on the workstation the microVAX is selected and a request to immediately monitor a list of devices is send to the server on the microVAX. The server sends the request via shareable memory to the sampling program which carries out the action. The VAX 8650 is then selected using the same operator interface, and a request is sent via the server to the sampling program running on the VAX 8650 to start logging the readings of a list of devices to the local history file.

The replay software is also divided into a user interface which can be run from any node on the network, and a server running on each node which has recorded device readings at some time during the production cycle. Communication between the user interface and the server is again done with RPCs.

For example, a user on the VAX 8650 can request device readings from a remote microVAX. The data is retrieved from one or more of the local history files, converted to the appropriate units and returned to the VAX 8650 which then plots the data on a graphics screen (Figure 2).

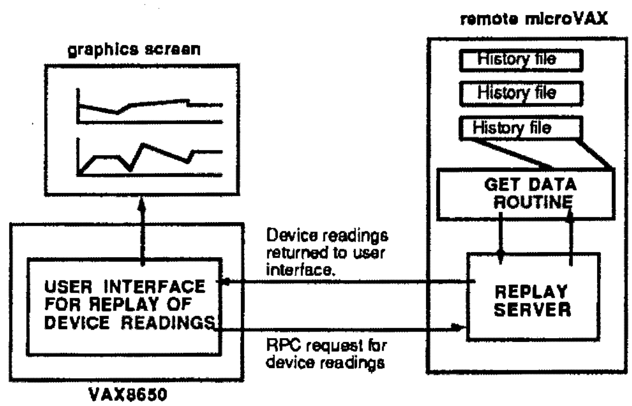

Figure 2. Method of retrieving data across the network and displaying it on a local screen.

3.4 Changes in the Structure of the History File

Originally device readings were recorded to a single file for the complete run cycle, which was typically five to six months, each year. In 1990, this file reached a size of about 300 Mbytes (ie.25,000,000 device readings). The file was sequential with direct access to the nearest eight hour pointer. Shared access to the file for both the logging program and the replay program was achieved with VMS locks, which resulted in both programs being slowed by lock manipulation. Another cause of slow data retrieval for replay was that each record read resulted in disk I/O.

The redesigned logging software opens a new history file every twenty four hours on each node where device readings are

ISBN: 978-3-95450-254-7 ISSN: 2226-0358 doi:10.18429/JACoW-ICALEPCS1991-S08NC03

being logged. When data replay is requested the replay software maps the relevant section of the history file into process memory and extracts data for the required devices. As replay of the most recent data must be available it may be necessary for the replay software to map to the section of the file being updated by the logging program. To prevent conflict between the two sets of software the history file is sequential and new records are appended to the file.

This method of sharing access to the file has significantly improved the rate at which data can be replayed because it has decreased disk I/O and does not require any file locking. It has also decreased the CPU usage for logging device readings as the file structure has been simplified and the need for disk I/O to access an intermediate time index file has been eliminated. Another benefit of the smaller history files is that they are easier to backup.

3.5 Other Design Changes

A high frequency of data sampling (every one or ten seconds) may be required for one or more devices over a period to study some effect or diagnose a problem. At LAMPF one device being logged every one second accumulates more than 1.3 Mbyte (86,400 device readings) of data per day. This frequency uses CPU during the logging process, and adds significantly to the volume of data stored in the history files so that it slows data retrieval for replay. To avoid unnecessary sampling at higher frequencies, the new software will allow the user to specifically enter the time period for which devices are required to be read at the one or ten second frequency, before they drop back to a less frequent sampling time. If no specific time is entered a default time will be used.

IV. FURTHER POSSIBLE IMPROVEMENTS

One possibility which has yet to be explored is whether the device readings should be grouped into seperate 24 hour files based on some relationship such as type of device, or location in the beam line. This would increase the CPU usage at the time the device readings were logged but the improvement in replay speed achieved by the further granularity of the history files could be significant. Changing to a different history file organization, such as an index file is not possible while the retrieval software is accessing the history file by mapping it into process memory. Another disadvantage of index files is that they are space consuming. At the moment it is possible to maintain the history files for a whole year on disk which allows for efficient replay of device readings over a complete production run.

V. PERFORMANCE GAINS

CPU usage by the new sampling software on the main control computer has been reduced by a factor of five but this is partly due to changes in the way in which the LCS Data System handles requests for device readings (message passing has been reduced by buffering device readings). It is hard to acquire accurate figures for the improvement in the speed of replaying device data, as this depends on the load on the computer and the size of the relevant history files. However for a short time both software systems were running on the same computer recording the same device readings in seperate history files. At that time it was possible to make accurate comparisons which showed that the speed of replaying data has improved by a factor of six. The average rate of data retrieval for replay is 200 to 300 readings per second on the main control computer. Replay of device readings from other nodes in the system is faster as history files are much smaller.

VI. PRODUCTIVITY GAINS

The sampling of device readings is now being carried out on several nodes on the network. In 1991 approximately 2,000 devices were sampled throughout the production run. Since the new software was implemented the amount of data logged has almost doubled to 40,000,000 device readings (ie. 650 Mbytes) for the 1991 production run. Replay of the device readings can be done anywhere on the the network.

VII. ACKNOWLEDGEMENTS

The author would like to thank Eric Bjorklund, Dave Schultz, and other members of the controls section of MP-6 for helpful ideas during the software design phase.

VIII. REFERENCES

[1] S. Schaller and E. Bjorklund, "Providing Common Data Access for the LAMPF and PSR Control Systems", Proc. 1987 IEEE Particle Accelerator Conference, Washington DC (IEE Publishing, New York, 1987) p 745.