Fermi National Accelerator Laboratory

FERMILAB-Conf-91/171

Uniform Remote Control of Front End Processors in PAN-DA

D. Berg, E. Berman, B. MacKinnon, T. Nicinski, G. Oleynik, D. Petravick,
R. Pordes, G. Sergey, D. Slimmer, J. Streets, and V. White

Online and Data Acquisition Software Groups, Online Support Department

Fermi National Accelerator Laboratory
P.O. Box 500, Batavia, Illinois 60510

June 1991

* Presented at the *7th Real Time Conference*, June 25-28, 1991, Julich, Federal Republic of Germany.

Operated by Universities Research Association Inc. under contract with the United States Department of Energy

Uniform Remote Control of Front End Processors in PAN-DA[1]

David Berg, Eileen Berman, Bryan MacKinnon, Tom Nicinski,Gene Oleynik, Don Petravick, Ruth Pordes, Gary Sergey, David Slimmer, Jonathan Streets, Vicky White[2]

Online and Data Acquisition Software Groups
Online Support Department
Fermi National Accelerator Laboratory
Batavia, Illinois 60510

Abstract

The PAN-DA data acquisition software system encompasses support for a variety of front end processors including the Struck General Purpose Master (GPM), the Fermilab Smart Crate Controller (FSCC), and the Motorola MVME133-A VME single board computer. PAN-DA provides support for the remote control of these processors through its Remote Procedure eXecution (RPX) software. This software runs over serial lines, and in some PAN-DA environments, over the Ethernet and TCP/IP as well.

Experiments E791 and E771 have successfully used RPX software in both FSCC software development and data collection during the 1990-1991 fixed target run at Fermilab. RPX software has facilitated development and testing of the Fermilab Silicon Strip Detector readout system, for which the FSCC is an integral component. RPX based control and monitoring of the GPM and the MVME133-A are essential parts of the PAN-DA data acquisition system used by E687 and E773 during the 1990-1991 run.

I. INTRODUCTION

Although a greater amount of intelligence is finding its way into front end processors, because of their dedicated task and real time requirements, they will always lack the full functionality of general purpose computers. Because of this, central control is usually provided by more appropriate general purpose computers.

This paper discusses the software provided for controlling PAN-DA [1,2] front end processors from remote hosts. This software provides a uniform high level interface through remote procedure calls for a variety of front end processors, and a variety of other levels of communications mechanisms depending on the specific front end processor. Two specific applications of the control software are presented: The PAN-DA data acquisition control and remote control of FSCCs.

II. SYSTEM LEVEL SOFTWARE

The communications software uses Software Components Group's operating system, pSOS, and was developed with their pROBE[2] board level debugger. The software and device drivers are developed on VAXes using cross tools from Microtec[3], Inc., including C and Pascal compilers, assembler, librarian, and linker. Source code and product management are also handled on the VAX. The Microtec Run Time Library, which supports both C and Pascal, has been slightly modified to provide the process reentrancy necessary for multitasking.

Our embedded systems environment is supported on 68000 based processors, and has been ported to all PAN-DA front end processors discussed here. It integrates the operating system, the board hardware, the Run Time Library and other high level language support, utility libraries, a serial port driver, and drivers for other processor peripherals. The serial ports are accessible by standard language I/O constructs.

System PROMs provided with the FSCC [3,4] include an Ethernet downloader called DETH [4]. DETH comes up automatically when the board is reset, and listens for download requests from VAX hosts. Applications can be started automatically when downloaded by DETH, or at a later time. The MVME133-A can be downloaded over the in-house vertical bus (Branch Bus) or RS-232. The GPM can be downloaded only over RS-232. The embedded environment also provides the capability to boot applications from ROM, or other non volatile memories, by copying into RAM, or running directly out of ROM.

III. COMMUNICATIONS SOFTWARE

Supported front end processors are able to communicate with the outside world over RS-232, Ethernet, or both. The software we have layered on top of these two media is depicted in Figure 1.

[1] Sponsored by DOE contract No. DE-AC02-76CH03000

[2] pSOS and pROBE are trademarks of Software Components Group, Inc.

[3] Microtec is a registered trademark of Microtec Research, Inc.

COMMUNICATIONS SOFTWARE

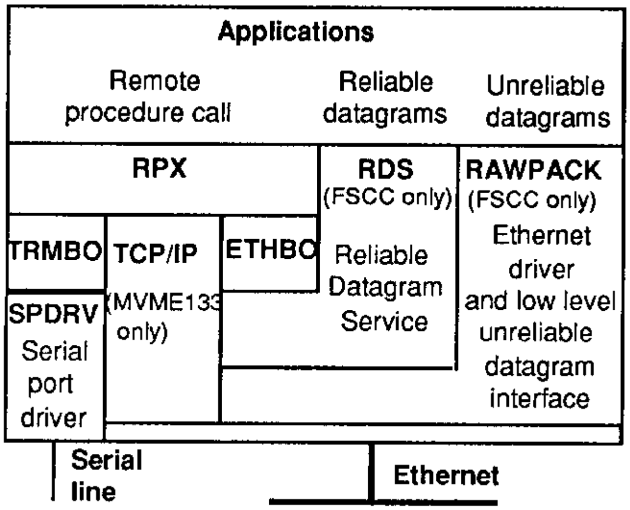

Figure 1.

The Ethernet interface, in addition to providing a higher speed path than RS-232, reduces the number of lines required to access a data acquisition system with multiple front end processors to one.

A.Ethernet

We support communications over Ethernet for two front end processors, the MVME133-A and the FSCC. The GPM does not have an Ethernet interface.

The MVME133-A uses an Ethernet and TCP/IP co-processor board, the ENP-10[4] from Communications Machinery Corporation, to provide raw Ethernet and TCP/IP communications. A MVME133-A resident multitasking socket driver provides a subset of Berkley Unix BSD compatible socket functions for TCP/IP and raw Ethernet communications. A generic TCP server shell is provided to aid in the writing of servers, and is used to provide the standard TCP echo and discard services for remote diagnostics. The typical per connection rate to a VMS or Unix system is 140 Kbytes/sec, but the total throughput from the MVME133-A is limited to 190 Kbytes/sec by the processing power of the ENP-10.

TCP/IP is not currently available on the FSCC. Instead, the FSCC uses an in-house Ethernet communications protocol, the Reliable Datagram Service (RDS). RDS can provide assured delivery of packets of up to 1500 bytes in length at up to 70 Kbytes/sec between FSCCs and VAXes. The FSCC Ethernet downloader is RDS based.

RDS is layered on top of a lower level multitasking datagram service, RAWPACK, which is the FSCC's Ethernet driver. The driver has a routine call interface and can be called directly by the user. A routine package with a nearly identical routine call interface is provided for VMS systems.

Remote Procedure Execution software runs on top of Ethernet basedTCP/IP on the MVME133-A, and the RDS based ETHBO software on the FSCC as indicated in the figure. RPX on the VAX supports either protocol, as well as the RS-232 TRMBO protocol.

B. RS-232

The Serial Port Driver, SPDRV [5], was designed to provide functions for use in normal terminal dialog with the user and as a communication medium over RS-232. All boards support RPX over RS-232 via the in-house TRMBO protocol. Since the Ethernet based RPX is much faster (20x on the FSCC, 40x on the MVME133-A), RS-232 based RPX is only used on front end processors without an Ethernet interface, like the GPM.

C. Remote Procedure Execution Software

In order to distribute applications among different processors and operating systems, nearly all communications within the PAN-DA data acquisition system are based on our implementation of a Remote Procedure eXecution package (RPX) [5]. It allows application tasks on VMS, pSOS, and Unix systems to communicate through subroutine calls in a standard way over Ethernet or RS-232.

Allowing for RPX requires the simple awareness that a subroutine could execute on the same or different computer than its calling routine. By designing and implementing code with this in mind, one can code and test a package on a single processor and subsequently easily split the package, at the subroutine level, to run on multiple processors.

RPX is based on the client server model in which the client makes a procedure call and the server executes it remotely on behalf of the client. A single Ethernet or RS-232 line can service routine calls from multiple clients. There is also a need for clients to make procedure calls to multiple servers, for example starting up data acquisition on multiple FSCCs from a VAX via remote procedure calls. Switchable RPX client software is provided for this purpose. A switchable RPX client can specify the server for its next procedure call by making a (local) routine call.

[4] ENP is a trademark of the Communications Machinery corporation.

IV. E687 and E773 PAN-DA CONTROL

Two Fermilab experiments, E687 and E773, are using nearly identical PAN-DA data acquisition systems. Each uses a GPM to read out FASTBUS memories, build events from the data, and transmit the events to ACP processors in VME. The ACP processors are controlled from an MVME133-A processor, named the zookeeker, which runs software to log events to tape and distribute selected events to VAX and Unix workstations for analysis over TCP/IP. The entire system is controlled by remote procedure calls between a single VAX workstation and the PAN-DA components. Figure 2 illustrates the basic PAN-DA components.

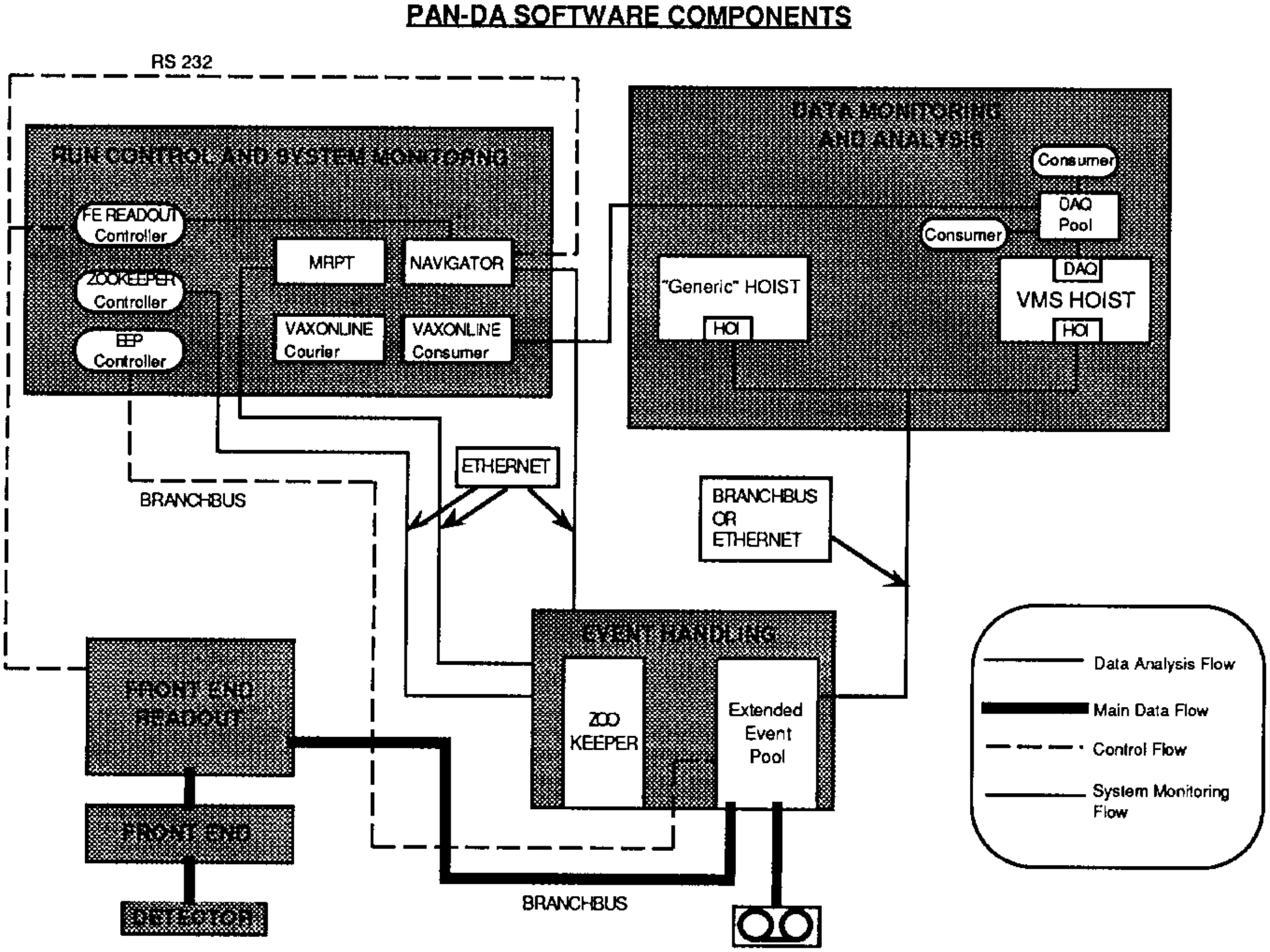

Figure 2.

The VAX controls the zookeeper by placing RPX calls to a zookeeper resident RPX server. Each component process of the zookeeper has associated standardized run control stub routines associated with it start the component up, pause it, resume it, or shut it down. The VAX run control calls these routines in a prescribed order to start the data acquisition process on the zookeeper, etc. The zookeeper periodically sends status data to a process on the VAX, which keeps the information in the PAN-DA Database (PDB) for use by run control and monitor programs, such as PANDA VIEW (a standard component of PAN-DA). Run control polls the database for state change information in order to synchronize distributed data acquisition functions. Status data is sent by placing remote procedure calls, and like the run control, each component provides status data by calling standardized stubs. Zookeeper components have the option of causing the data to be sent immediately to the VAX by declaring a significant PDB event.

The GPM is controlled in the same manner as the zookeeper board, though over a serial line. Since the basis of the run control and PDB updating is via remote procedure execution, the GPM implements them in the same manner as the zookeeper. Two PDB servers, one for the zookeeper (TCP/IP based), and one for the GPM (RS-232 based), run on the VAX to handle the different media for the PDB database.

Except for initializing the software that enables the zookeeper to communicate with the ACP processors, ACP processors are controlled over the in-house vertical "branch bus," which enables the VAX to read and write to VME memory.

V. E791 AND E771 FSCC CONTROL

Both E791 and E771 must download and control multiple FSCCs which read out the digitized data from the various components of their detectors. Both use the DETH software to download the FSCCs, and switchable RPX to control them with remote procedure calls.

E771 used the remote procedure functionality for development of FSCC software. They debugged complex diagnostic software on the VAX, while performing FASTBUS I/O primitives remotely with RPX.

E791 uses RPX in a more fundamental manner - to control the triggering and readout calibration data between accelerator spills. For the first half of the 1990-1991 run, they used RS-232 based RPX, and are now switching to Ethernet based RPX for the second half.

VII. REFERENCES

[1] Berg, et al, "The PAN-DA Data Acquisition System", IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[2] See accompanying paper ""PAN-DA and Beyond - Data Acquisition for the Next Generation Experiments"

[3] Bowden et al, "An Intelligent Readout Controller for FASTBUS, The Fermilab FSCC", IEEE Traansactions on Nuclear Science, Vol. NS-38,No. 2, April 1991.

[4] Berg et al, "Software for the Fermilab FASTBUS Smart Crate Controller" IEEE Traansactions on Nuclear Science, Vol. NS-38,No. 2, April 1991.

[5] Berg, et al, "A Real Time Integrated Environment for Motorola 680xx-based VME and FASTBUS Modules" IEEE Transactions on Nuclear Science, Vol.NS-36, No.5, October 1989.

[6] E.Berman, D.Petravick, G.Sergey, "Remote Procedure Execution Software for Distributed Systems" IEEE Transactions on Nuclear Science, Vol.NS-36,No.5, October 1989