Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

LBL--16148

DE83 016139

Presented at the Conference on Real-Time Computer
Application in Nuclear and Particle Physics,
Lawrence Berkeley Laboratory, Berkeley, CA,
May 15-19, 1983

DATA ACQUISITION AT THE BEVALAC--AN EXAMPLE:
THE HISS FACILITY

C. McParland

June 1983

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DATA ACQUISITION AT THE BEVALAC, AN EXAMPLE: THE HISS FACILITY

Charles McParland
Nuclear Science Division, Lawrence Berkeley Laboratory
and Space Sciences Laboratory
University of California, Berkeley, CA 94720

The Heavy Ion Spectrometer System (HISS) is a relativistic heavy ion research facility at the Bevalac accelerator complex, Lawrence Berkeley Laboratory (LBL). This spectrometer can provide fields of up to 30k gauss within a volume of more than 3 cubic meters. This facility is equipped with a number of large, flexible detector systems. Over the last four years, a loosely coupled data acquisition and analysis system has been written for use at this facility. This system will be described in detail with particular emphasis on development of high data rate capabilities.

Hardware

Data Acquisition Computer

Data acquisition is performed on any of three PDP-11 front-end computers. Each of these has local disk storage and magnetic tape (either 800 or 1600 BPI). In addition, each front-end is interfaced to a single disk controller shared with a VAX 11/780. Experimental data are acquired from a number of CAMAC crates interconnected by a standard branch highway. These branch highways are interfaced to each front-end computer through a MBD programmable branch driver. The only CAMAC hardware the data acquisition system requires is a 12-channel input and 12-channel output register used by data acquisition software. These modules reside in stations 1 and 2 of each system's crate number 1. The system does not require any specialized interfaces or trigger modules. An optional high-capacity CAMAC memory module (detailed below) can be added to increase performance. Since several major detector systems are shared among experimental groups, dedicated CAMAC crates containing their electronics are typically moved from one front-end branch highway to another. It should be noted that all hardware (as well as system software) is unmodified and supported either locally or by the vendor.

Shared Disk System

The shared disk system consists of a single disk controller with a single 80 Mbyte disk drive. This controller emulates a Digital Equipment RMD3 disk system and uses the DEC-supplied software disk driver. The controller is interfaced to all three front-end PDP-11's and to the VAX (via the Unibus). Each processor accesses the disk as if it were a privately owned device. The "sharing" function is performed by the controller, which round-robins among all four interfaces, pausing at each long enough to service any pending disk requests. Latencies caused by multiple simultaneous requests appear to individual systems simply as excessively long disk seek operations. Since neither the RSX-11M or VMS operating systems attempts seek-optimization for these disks, no system intervention is triggered.

Analysis Computer

The analysis computer is a VAX 11/780 with 4 Mbyte of memory and over 1 Gbyte of on-line storage. In addition to the shared disk system described above, the system has a simple magnetic tape drive (800/1600 BPI), a Versatec printer-plotter, several line printers, three DECNET high-speed data links, and a host of terminal interfaces.

This manuscript was printed from originals provided by the author.

HISS COMPUTER SYSTEM

Software

Data Acquisition-DATAQ and the MBD

The set of programs that run on each of the front-end acquisition machines is called DATAQ.* These programs are an outgrowth of an earlier system developed at LBL (QDA/MULTI). Although these systems share a similar organization and command repertoire, internally they are substantially different. These systems differ principally in their dealings with the MBD.

The MBD CAMAC branch driver is a stored-program processor with two external data paths. One is a standard CAMAC parallel branch highway. The other is a Unibus (DMA) access path to PDP-11 memory. The Unibus data path performs only full word transfers (i.e., no byte addressing) and only addresses the lower 64K words of PDP memory. Although schemes exist to allow full 124K word addressing, these necessitate unacceptable programming and performance concessions within the MBD. The MBD therefore accesses that portion of the lower 64K words of PDP-11 memory not occupied by the RSX-11M executive.

The MBD data acquisition program has sole responsibility for actual CAMAC data acquisition. Each interrupt trigger (of 12 possible) starts a section of user-written MBD code that acquires and formats an event. These events are of variable length and contain a fixed header consisting only of a byte count and event id. The remainder of the event is formatted by the user and contains the output

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

*We wish to acknowledge the efforts of F. Weik (formerly of GSI) for major development of DATAQ.

of whatever CAMAC modules are desired. The MBD code places these events into a buffer area in PDP-11 memory. When it appears that an event will cause a buffer to exceed some predetermined length (currently 8K bytes), the MBD declares that section of PDP-11 memory to be a valid data buffer and releases it to the program DATACQ. It should be noted that the MBD is initially given a large area of memory by DATACQ, and it is responsible for filling in and breaking up this area into smaller, variable length data buffers. The only restrictions placed on the MBD are that it use only memory "given" to it by DATACQ and that it not produce buffers exceeding a predetermined length.

DATACQ is primarily a sophisticated buffer handling program. Although it does pass messages to the MBD code (e.g., begin run, end run), its only real task is to "give" memory to the MBD and then distribute the results of the MBD's efforts to tape and disk. The initial implementation of DATACQ passed data buffers from the MBD directly to tape and disk. This meant that during the time of actual tape and disk transfer, these buffers were locked down and unavailable to either DATACQ or the MBD. As the Bevalac accelerator produces an experimental beam only one second out of every six, it was felt that some form of in-core data caching during the one-second beam spill would be of great advantage. Accordingly, the tape and disk output routines of DATACQ were rewritten to pass data buffers off to a larger (up to 32K word) buffer in upper PDP-11 memory. The transfer was accomplished by use of system memory mapping services and a programmed move loop. Although time consuming, this method was able to release data buffers to DATACQ far quicker than direct transfers to tape would allow. Data transferred to this secondary buffer area was subsequently routed to tape and shared disk. In addition, data buffers within this area were available, on a sampling basis, to any monitoring program requesting them.

The Quest for Data Rate

existing, unaltered PDP-11 code is given five seconds to process and record one second's worth of data. When data rate is not compromised, the system inefficiencies detailed above suddenly become acceptable. In recent tests of a similar experimental setup, we have seen data rates of about 100K words/second instantaneous. It is somewhat amusing to see the PDP-11 finishing up tape and disk transfers from the previous spill at the same time the MBD is collecting experiment data from the current spill.

Poor Performance:

Experience with this two-tiered buffering scheme had shown performance of about 25-30K words/second in a typical experimental setup. Although this was within a factor of two of the design goal, it was less than spectacular performance. Investigation of the system's behavior was undertaken. Parts of the MBD data acquisition code were instrumented with writes to a CAMAC output module. Resulting waveforms demonstrated that, on occasion, 40 to 100 milliseconds elapsed while the MBD waited for a buffer from DATACQ. Further investigation revealed that this lost time was distributed among extensive buffer consistency checks within DATACQ, inter-task communication overhead, the extremely slow nature of system memory mapping directives, and, naturally, poor magnetic tape performance. Although increased tape speed (perhaps 6250 BPI) would help performance, it was noted that other factors mentioned above were inhibiting full utilization of the secondary buffer memory—tape speed was not the limiting factor.

Performance Regained:

The obvious antidote for such poor performance is more memory. However, given the limitations of MBD addressing and the time-consuming manner in which memory must be moved around within the PDP-11, we chose to place a large (256K word) memory buffer where it would be of greatest advantage—in CAMAC. By placing such a large memory at the top of the data chain, the MBD is able to store collected events with no regard for either room availability or buffer formatting considerations. It is only after storing a one-second spill of data that the MBD begins to transfer events into PDP-11 memory. Thus, the

Shared Disk Architecture

Pieces of software in both the data acquisition and the analysis machine are co-responsible for writing and reading data in the appropriate shared disk file. Unfortunately, while the disk controller is capable of transparently processing disk I/O requests from up to four computers, it is quite ignorant of any higher level structure on the disk. This on-disk file structure is built and maintained by the operating system. In an effort to increase file system throughput, all four operating systems (RSX and VMS) maintain in-core caches of currently used directories and disk allocation bit maps. Therefore, it is impossible to maintain a dynamic and fully shared file system on this disk. Our approach has been to use standard file utilities to create three large data files (one for each front-end) and a number of small descriptor files. The file system access codes for all these files and, in fact, for all file directories residing on this disk are then modified to deny all users the ability to create, delete, or extend any file. In short, the disk contains a standard—but static—file structure. This allows the use of standard system file I/O routines to access both data and descriptor files.

Each front-end data file has a small associated descriptor file. This file contains pointers to the location within a data file of the latest data, the oldest data, and the most recent beginning-of-run dataset. In addition, counters are kept for the number of updates by a given front-end and the current calendar time of that front-end. These descriptor files are updated only by the data acquisition program in the front-end computer; all other processes have read-only access. It is very important that the algorithm for updating these files be both reliable

and restartable. In our application, restartability implies the ability to determine the correct place to append new data regardless of where and how the data collection process stopped. The need is most evident when the data acquisition machine crashes. Our algorithm maintains each front-end data file as a circular buffer. There exists, between the head of the buffer (i.e., newest data) and the tail (i.e., oldest data), a large gap. Starting at the newest data, the front-end appends data to a given file by writing into the gap between head and tail. Only after all data are written to disk is a new set of descriptor pointers calculated and written into the appropriate descriptor file. This algorithm guarantees a consistent on-disk data structure and has proven itself restartable through great numbers of system crashes. In the event of a front-end system crash, analysis and monitoring programs running in the VAX require no intervention. They simply await a reboot of the data acquisition machine and the subsequent availability of new data.

The VAX makes experimental data available to user programs through the use of a standard program interface. This interface allows access to the following: live data from any of the three front-end computers, previously collected data stored on disk, or a series of special facility monitoring data sources. The data interface is dealt with in detail elsewhere in this volume (see A Uniform Data Interface, Mark Bronson). However, a few details are worth mentioning here. There is no single VAX process that polls the shared disk to look for recent data. Each process that opens a logical channel to the shared disk invokes subroutines that perform this polling on the user's behalf. A small global section exists for each of the three front-ends. These sections contain, among other things, copies of the descriptor files mentioned earlier. When more than one process has a channel open to live shared disk data, the data base routines interrogate a time stamp placed in the appropriate global section. If a recent polling by some other process has provided timely information, an unnecessary polling is eliminated. Thus, a large number of analysis programs can access the same live data source without imposing an unacceptable overhead in descriptor file polling. In contrast, the actual data requested by separate processes is not shared and must, unfortunately, be read by each process individually. It is our current experience, though, that enough analysis tasks are

compute or display bound to not cause serious contention problems for the use of this shared disk.

The use of a uniform data interface has given experimenters a great deal of variety in analysis packages. The Fermilab MULTI analysis package has been adopted for use with the uniform data interface routines. The ease of redirection has made MULTI useful in monitoring dipole engineering data (cryogenic plant states, etc.) as well as experimental data. Experimenters have also provided programs that connect the uniform data interface to the CERN HPL0T/HBOOK package. In addition, the large FORTRAN-based analysis package LULU (see article by H. Crawford, this volume) has been interfaced to these subroutines and is currently in use for on-line monitoring as well as off-line analysis.

Observations

After several years' experience with this system, we have found some of our initial assumptions borne out. Our initial design sought to keep the connections between front-end acquisition machines and the VAX analysis machine quite "loose". We understood that it would be unacceptable to allow the failure of any single machine (e.g., system crash) to affect the state of the others. However, we also sought to eliminate any need for a spiders' web of intercommunicating tasks spanning the gap between the acquisition and analysis machines. We felt that close participation between a VAX process and DATA00 would violate the "clean" separation between data acquisition and data analysis/monitoring and would, inevitably, lead to stopped programs in unforeseen system scheduler states.

Our initial decision has proven to be correct. No real need has arisen for closer participation (e.g., task-to-task communication) between machines. We feel this is due to several factors. First, the data accessing routines have proven to be both easy and desirable for experimenters to implement. Simple redirection of analysis programs among live and stored data sources is a difficult feature to pass up. These routines cannot be fully implemented within the confines of the PDP-11; they are available only on the VAX. Experimenters have incorporated these routines into both large analysis packages and simple monitoring programs. In each instance, we are able to place the experimenter as "close" to live data as seems necessary. As long as these routines obviate the need to get "closer" to the data, programs will not migrate into the front-end machine.

A second factor in attracting, and keeping, experimenter code in the analysis machine is the VAX itself. The variety of software tools available on the VAX make it a far more pleasant and productive programming environment than the PDP-11. Freed from the constraints of the 16-bit address space of the PDP-11, data can be dealt with in a straightforward manner using large data arrays.

Future Plans

As a final note, it should be noted that the current shared disk installation has been implemented at two additional beam areas. At these locations, completely different experimental areas share a single VAX 11/780 in the same manner as the three separate front-end machines at HISS. An additional VAX 11/780 has just been purchased for use in a third shared disk configuration. This system will be positioned to offer on-line data analysis to the remaining Bevalac experimental areas.

This work was supported by the Director, Office of Energy Research, Division of Nuclear Physics of the Office of High Energy and Nuclear Physics of the U.S. Department of Energy under Contract DE-AC03-76SF00098.

This report was done with support from the Department of Energy. Any conclusions or opinions expressed in this report represent solely those of the author(s) and not necessarily those of The Regents of the University of California, the Lawrence Berkeley Laboratory or the Department of Energy.

Reference to a company or product name does not imply approval or recommendation of the product by the University of California or the U.S. Department of Energy to the exclusion of others that may be suitable.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

NOTICE
PORTIONS OF THIS REPORT ARE ILLEGIBLE.
It has been reproduced from the best available copy to permit the broadest possible availability.