DØ Run II Online Computing System

An Introduction

DØ Note 3580

Stuart Fuess

11-Jan-1999

1. Overview

1.1 Functions

The principal functions of the DØ Online Computing system are:

- Detector hardware control and monitoring;
- Primary (secondary) event data acquisition at rates up to 50 Hz instantaneous with 250 kbyte/event (20 kbyte/event) data size;
- Event data monitoring;
- Control room activities.

This document is an introduction to the requirements and design of the Online system.

1.2 Scope of the Document

Computing and data acquisition portions of the DØ Upgrade project are divided into *Trigger* and *Online Computing* sub-projects. The *Trigger* sub-project deals with the hardware components of the physics event trigger, customized fast trigger processors, and the infrastructure of the high level (Level 3) software trigger farm. The filter code which runs in Level 3 is managed as part of the DØ *Offline* computing effort and is not part of the WBS structure of the Upgrade hardware effort. This document, on the *Online Computing* sub-project of the DØ Upgrade, excludes all components of the event trigger system other than those associated with higher level configuration, monitoring, and control activities. Details of the trigger hardware and Level 3 software infrastructure can be found in the *Trigger* sub-project documentation set. Details of the Level 3 software filter code are part of the *Offline* documentation set.

2. Hardware Architecture

2.1 Functions

The hardware design of the Online system is dictated by the various functions to be performed. These functions and the different hardware strategies suggested are summarized in Table 1 *Hardware System Functions*.

The complete specification of the requirements of the hardware components of the Online system is dependent upon the software architecture. The following Design section

assumes some of the schemes to be discussed within the Software Architecture component of this document.

System Function	***System Requirements***
Database operations	High availability
Primary event data acquisition	High availability High I/O bandwidth High network bandwidth Large application memory buffer space Large I/O disk buffer space
Secondary event data acquisition	Moderate network bandwidth
Detector control and monitoring	Embedded processors GUI interfaces Isolated network (security)
Event monitoring	High network bandwidth High CPU capacity
Control room applications	GUI interfaces

Table 1 ***Hardware System Functions***

2.2 Hardware Design

To supply the previously listed functionality, the computing hardware design includes the following components:

- Gigabit Ethernet Network;
- Central UNIX Host System;
- Control Room PCs;
- Monitoring PCs; and
- Front End Processors.

Figure 1 illustrates the design. Each of the basic components is described in the following sections.

Note that there is no tape writing capability (other than for backups) local to the Online system. Completed event data files of approximately 1 Gbyte size are sent over the network to the Feynman Computing Center (FCC) for logging to tape.

2.2.1 Gigabit Ethernet Network

The network backbone is a multi-port Gigabit and 100 Megabit Ethernet switch. Only the UNIX Host nodes and the link to the FCC are at full Gigabit speeds; other devices are connected at 100 or 10 Megabits.

2.2.2 Central UNIX Host System

A set of Compaq/Digital AlphaServer UNIX servers is chosen to perform those activities which are most demanding in terms of availability, I/O bandwidths, network bandwidths, and process memory space. Thus these servers are the hosts for database and the main

event data acquisition activities. Note that the central Host system is not a high capacity CPU resource, having only the database access and data transport requirements.

The servers are arranged in a cluster fashion, with a shared RAID disk set housing the databases and the important Online applications. The cluster also allows rapid failover of applications from one node to another. High availability is achieved by using redundant controllers and power supplies in the shared RAID arrays and by having redundant Host nodes which can singly provide all Host functions.

The Host servers each have local large high speed Ultra SCSI disk arrays for event data buffering. They additionally support high speed Gigabit Ethernet network interfaces.

Two of the three eventual Host servers have been purchased (as of 1/10/1999). They are both model AlphaServer 4000s, each with four independent PCI busses providing a total of 16 PCI peripheral slots per node. In practice a PCI bus will likely be dedicated to serving the Gigabit Ethernet interface, with the other busses used to support disk controllers. The AlphaServer 4000 can support two CPU boards; currently each has only a single processor.

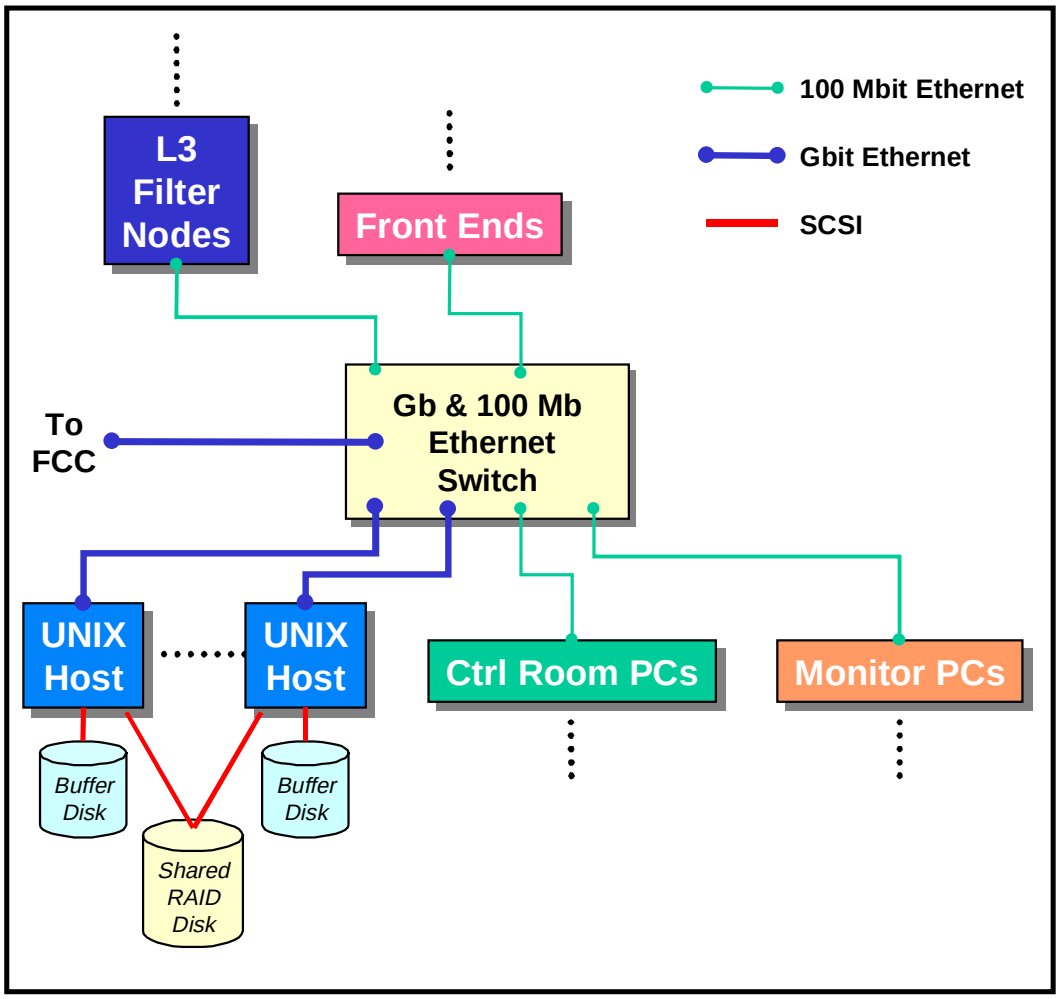

Figure 1 ***Basic Hardware Components***

2.2.3 Control Room PCs

A number of computing stations are required for Control Room operations. For this function we have chosen to use commodity PCs running Windows NT. The PCs provide an inexpensive and easily manageable platform. We have chosen Windows NT both because of the familiarity of DØ personnel with its interface and because of our growing familiarity with its programming environment, particularly as related to the Level 3 software development effort. As will be noted in the Software section, the Control Room nodes are typically running GUI client applications which communicate with external server processes. We are developing the GUIs using a suite of standard (e.g. Python) and DØ-specific (Client / Server) packages. The use of the NT platform is restricted to these user-interface applications.

2.2.4 Monitoring PCs

The real-time monitoring of the event stream by partial reconstruction and analysis is a highly CPU intensive activity. For certain types of analysis, it is also a network intensive activity. The software to perform these analyses is typically a variant of the Offline reconstruction and analysis software, and hence development and operation benefit from a similar environment. We have chosen to meet these needs by the use of a collection of PCs running the Linux operating system. We choose PCs because of their excellent processor price/performance, with expected continued improvement. We use a collection of PCs to divide the monitoring duties and network loads among a number of processors. Finally, we choose Linux in a pseudo batch environment because of the similarity to the Offline reconstruction and analysis environment.

2.2.5 Front End Processors

There are three classes of embedded processors in use at DØ, with either hardware, readout, or trigger control functions. The trigger processors are detector type specific, and are the responsibility of the detector hardware groups. Other than the support for network connections, downloading, and remote access, they are not considered part of the Online system documented here.

The processors situated in the detector readout crates have several functions. During normal event data taking they can parasitically access the detector raw information and perform low-level data quality monitoring. During dedicated calibration periods, events are directed to the local processor for determination of the calibration constants. The local processor does not participate in normal data acquisition, but a debugging and commissioning mode allows for readout of the VME crate and data transmission to a host receiver. Finally, the embedded processor serves as the remote link to the VME crate address space, and manages the configuration and downloading of the devices within the crate. Most of the functions of these processors are determined by and developed by the individual detector groups. However, the Online Computing group provides much of the infrastructure for the operations and software aspects of these nodes.

The third class of embedded VME processor is dedicated to control and monitoring of the detector elements, including for example low voltage, high voltage, and environmental conditions. These processors are an integral part of the Online Control System, and are managed and programmed as part of the Online Computing sub-project.

3. Software Architecture

3.1 Component Introduction

The software components of the Online system provide the following required functions:

- Shared software infrastructure;
- Hardware control and monitoring;
- Event data acquisition;
- Event monitoring; and
- Calibration.

Each of these components is discussed in more detail in the following sections.

3.2 Shared software infrastructure

There are several software packages that are used by numerous higher level applications. These are:

3.2.1 Database

The database management system of choice for the Online system is ORACLE. Within the DBMS will be kept several independent databases for:

- EPICS control system parameters (Hardware Database)
- Detector configuration sets
- Trigger configuration sets
- Run and data file information
- Calibration information
- Detector monitoring information
- Luminosity information

The tools being developed for managing the DBMS and the entries within the databases will be shared among the above activities.

3.2.2 Inter-task Communication

The distributed, real-time nature of the Online software applications dictates the need for efficient and reliable communication among cooperating tasks. DØ has chosen to address this need by creating a C++ Client / Server library, providing a common means by which applications can exchange messages. This package is built upon the publicly available ACE product.

We have chosen to use the messaging paradigm as opposed to a distributed object paradigm, as would for example be supported by a CORBA implementation. We believe that the messaging abstraction fits well our application environment. The detailed operational aspects of the socket-based messaging package are also easily controlled and monitored.

3.3 Hardware Control and Monitoring

With a wide range of hardware device types to control, and a need for numerous custom interface applications, it is clear that there are a large number of software packages and applications to be developed. The DØ Online group has chosen to use the EPICS control system package as the basis for our control system. Upon this base are built a number of additional applications for specific DØ purposes. DØ uses the Python scripting language to build many of these applications.

The package and application list follows.

3.3.1 Hardware Database

EPICS makes use of distributed database-like files on the front ends to characterize specific device properties. DØ has chosen to derive these distributed databases from a central ORACLE database instance, denoted as the Hardware Database. Into this database goes information for each hardware component within the detector to be accessed via the control system. The database includes access (name to address translation), field bus parameterization, data transformation (into natural units), and alarm limit information.

3.3.2 VME Downloading

Numerous VME-resident hardware components need to be configured or otherwise downloaded in preparation for data acquisition activities. Access to the VME memory space is provided via EPICS devices.

3.3.3 Rack Monitors

Environmental monitoring for the detector electronics is provided by a standard Rack Monitor device. Software support for the Rack Monitor includes the EPICS device definitions and a GUI application interface.

The Rack Monitors are specific examples of devices reachable over the 1553 bus, for which DØ has developed EPICS device and record support.

3.3.4 Low Voltage Power Supply Control

The remote control and monitoring of the Low Voltage power supplies is again achieved through the use of EPICS as the layer below custom Python graphical user interface applications.

3.3.5 High Voltage Power Supply Control

The software developed for High Voltage supplies supports higher level complex operations, for example *Ramp*, in addition to the simpler *On/Off/Read* operations. The GUI interface is also more complex in order to manage effectively the hundreds of HV channels within the detector.

3.3.6 Clock Control

The DØ Clock synchronizes the detector to the accelerator cycle. The Clock can be configured, via EPICS control, to provide various signals to detector elements.

3.3.7 Alarm System

The DØ detector is continuously monitored for failing components. The bulk of the monitoring information is acquired by the embedded processors, either by accessing the local VME hardware or by connection to devices on the 1553 auxiliary busses. Cooperating software applications may also generate alarms to indicate abnormal conditions.

The Alarm System contains a number of software components:

- Alarm Server
- Alarm Display
- Alarm Logger

There is also a connection between the Alarm System and the Run Control system, allowing interruption of data acquisition upon notification of error conditions.

3.3.8 Secondary Data Acquisition

It is possible to use the Control path to perform data acquisition from a limited number of VME crates via their associated Front End processors. This secondary DAQ system uses software within the Front End to access the data buffers over the VME back plane, then transmits the information to an Event Builder application which collects data from multiple sources. The Event Builder passes the events to the Collector / Router application within the primary DAQ path for further logging and monitoring. Much of the Secondary DAQ path is built upon the Fermilab `DART` product.

3.3.9 EPICS Extensions

There are several general extensions to `EPICS` which need to be performed in support of some of the aforementioned Control and Monitoring applications. These include:

- Device specific records
- Alarm broadcast modes
- Vector generalization of process variables
- Hierarchical process variables.

3.4 Data Acquisition

There are three aspects to Data Acquisition:

- Detector configuration and Run Control;
- The Event Data Path; and
- Trigger and DAQ Monitoring

The principal software components of each are described below. In addition, the event data path is illustrated in Figure 2.

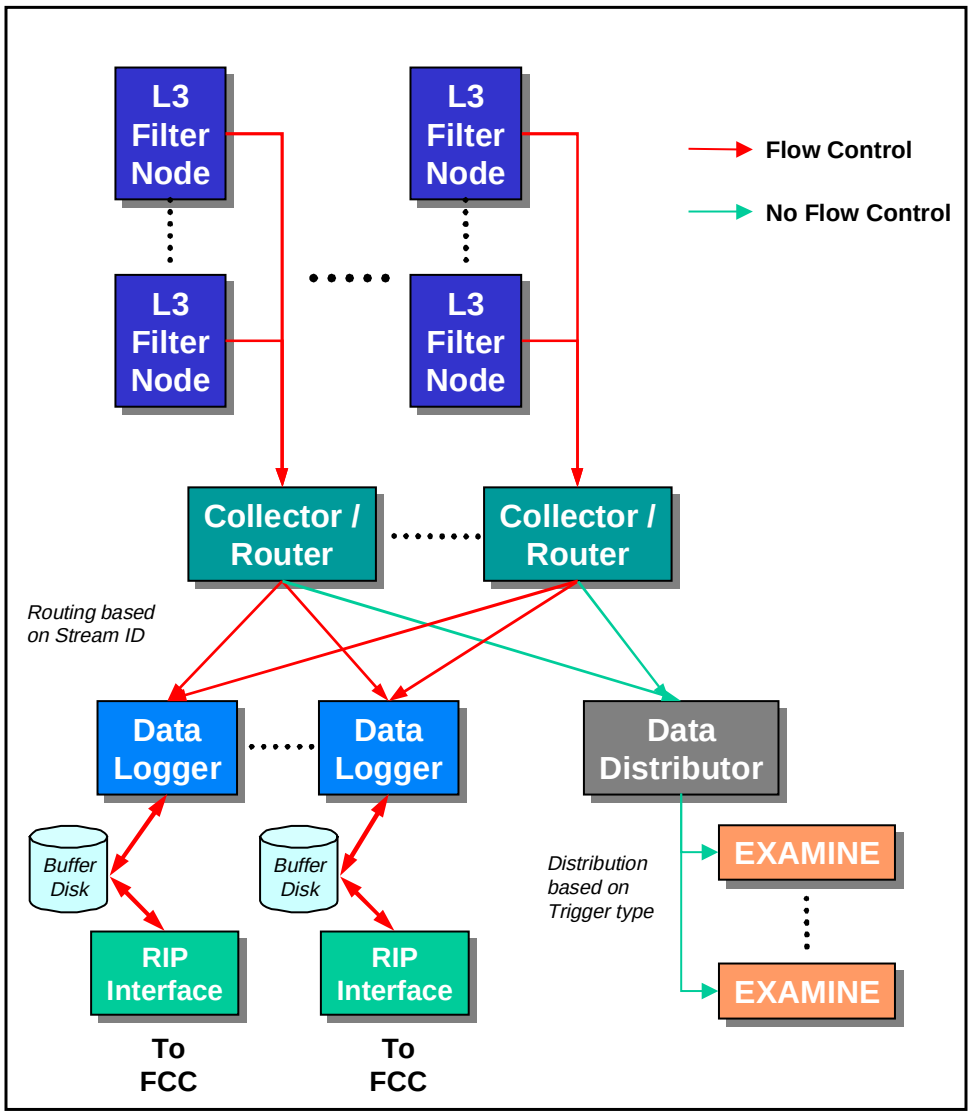

Figure 2 ***Event Data Path***

3.4.1 Configuration and Run Control

A single application, COOR, is responsible for establishing and managing the configuration of the detector, trigger systems, and data taking run status. This application communicates with almost every other Online sub-system: it accesses pre-defined configuration sets in the database, commands detector configuration and downloading, controls the trigger system, initiates and terminates runs, and makes log entries into the database. Associated with the COOR application is TAKER, the user control interface application.

3.4.2 Download Manager

COOR's interface to the detector hardware system is via another application which acts to translates COOR's logical configuration requests to the appropriate control system actions. Embedded within this download manager are the detector-specific functions for downloading calibration parameters, as well as for general hardware configuration.

3.4.3 Collector / Router

The Online system, with scope as defined in this document, sees data events as originating from the Level 3 filter nodes. The first application in the event path within the Online system, hence the target for Level 3 transmission of events, is the Collector / Router. There are multiple possible instances of the Collector / Router, so as to provide parallel data paths supplying a higher aggregate bandwidth. Data events are passed on to the appropriate Data Logger, as determined by the event trigger characteristics, and to one or more instances of the Data Distributor.

3.4.4 Data Logger

The Data Logger is where events are assembled into files on disk. DØ has chosen to produce nearly-exclusive data streams at the Online system as an aid to post-reconstruction data access. The Data Logger(s) are thus managing multiple data streams and building multiple simultaneous data files. As with the Collector / Router, the Data Logger is designed to be run with multiple simultaneous instances, where each instance would be associated with a subset of streams.

The Data Logger is responsible for synchronizing data file creation with database entries associated with accelerator luminosity recording.

3.4.5 Reconstruction Input Pipeline (RIP)

Once completed data files are assembled by the Data Logger(s), they are transmitted over the network to the Feynman Center for logging to serial media. Data management at Feynman is part of the RIP project. The Online interface to RIP provides the method for transmitting the data files and for monitoring the transmission.

3.4.6 Trigger and DAQ Monitoring

With numerous trigger levels, possible simultaneous runs, and distributed and parallel data path components, the operation of the Online data path is quite complicated. There are a set of monitoring applications connected to each independent trigger and data path component, and an integrated monitoring application devoted to summarizing the activity of the system.

3.5 Event Data Monitoring

In addition to the lowest level monitoring of the detector hardware, it is essential to monitor the quality of the data acquired. To do this, event data is distributed parasitically to a set of processing nodes. The components of this activity are described in the following sections.

3.5.1 Data Distributor

Events originating in the Level 3 nodes are distributed by the Collector / Router applications to the Data Loggers and to one or more Data Distributor applications. The transmission from Collector / Router to Data Distributor is by broadcast or on a best effort mode, such that the state of the Data Distributor has no possible effect on the successful smooth operation of the data logging path. The Data Distributor receives events in an unbiased fashion, and can further distribute such without bias.

Event consumers, as described below, subscribe to specific subsets of events as determined by trigger type. The Data Distributor maintains a queue of events for each consumer, transmitting the next event upon request. The Data Distributor is also responsible for distributing the run state information to the consumers.

3.5.2 EXAMINE

The ultimate data event monitoring consumer application is `EXAMINE`. There are many `EXAMINE` varieties and instances, where the varieties monitor different functions or trigger sets and the instances can distribute the load and pool results.

The `EXAMINE` applications are basically reconstruction programs, sharing the same structure and much of the same code with the Offline reconstruction and analysis programs. The typical output of an `EXAMINE` is a set of histograms or plots, which can be remotely viewed by a browser portion of the application.

3.5.3 Express Line

In addition to the real-time monitoring supplied by `EXAMINE`, it may be necessary to provide a farm of processors to perform more complex reconstruction and analysis operations. The Express Line will serve this function by acting as a small local batch farm, operating upon (parasitically copied) event data files.

3.6 Calibration

Calibration is very detector specific. Some calibration activities occur on local Front End processors, others within Level 3, and still others occur at the Host or even Offline. These various activities are coordinated in two locations – the databases containing the calibration information, and the Calibration Server, which acts as an interface between the user and the various sub-systems. Many of the details of calibration are still in design phase.

Calibration activities occurring in the Front End processors will be very similar to the local monitoring activities and the Secondary DAQ path operation. Data from the local crate will be asynchronously collected and analyzed, with the results made available via Control System calls.

Level 3 calibration operations are quite similar to normal event filter activities, but the results must be made available by task-to-task communication techniques.

4. Conclusion

This document has given a system overview and brief description of the components of the DØ Online Computing system. Many more details can be found on the Online web pages:

http://d0server1.fnal.gov/www/online_computing/online_computing.html

This document is also intended to give the description of the Online system as planned. Many components already exist, but some are still in design or development phases. There are companion documents available on the above web site to indicate the status and schedule for the Online system.