Event Builder and Level 3 Trigger at the CDF Experiment

K. Anikeev[1] *G. Bauer*[1], *I. Furić*[1], *D. Holmgren*[2], *A. Korn*[1], *I. Kravchenko*[1], *M. Mulhearn*[1], *P. Ngan*[1] *Ch. Paus*[1], *A. Rakitin*[1], *R. Rechenmacher*[2], *T. Shah*[1] *P. Sphicas*[1], *K. Sumorok*[1], *S. Tether*[1], *J. Tseng*[1], *F. Wüerthwein*[1]

[1] Massachusetts Institute of Technology, USA
[2] Fermi National Laboratory, USA

Abstract

The Event Builder and Level 3 trigger systems of the CDF experiment at Fermilab are required to process about 300 events per second, with an average event size of 180 KB. In the event building process the event is assembled from 15 sources supplying event fragments with roughly equal sizes of 12-16 KB. In the subsequent commercial processor-based Level 3 trigger, the events are reconstructed and trigger algorithms are applied. The CPU power required for filtering such a high data throughput rate exceeds 45000 MIPS. To meet these requirements a distributed and scalable architecture has been chosen. It is based on commodity components: VME-based CPU's for the data read out, an ATM switch for the event building and Pentium-based personal computers running the Linux operating system for the event processing. Event flow through ATM is controlled by a reflective memory ring. The roughly homogeneous distribution of the expected load allows the use of 100 Mbps Ethernet for event distribution and collection within the Level 3 system. Preliminary results from a test system obtained during the last year are presented.

Keywords: CDF, Level 3, Event Builder, Run II

1 Introduction

The Collider Detector at Fermilab (CDF) [1] is a general purpose particle detector which has taken over $100\ \mathrm{pb}^{-1}$ of data at the Fermilab Tevatron since 1987 and is scheduled to take data again in 2001, accumulating well over $10\ \mathrm{pb}^{-1}$ per week. As is common in high-luminosity experiments, a three-level hierarchy is used, where each succeeding level filters events on the basis of increasingly refined reconstructions of objects within the event. The first two trigger levels will reduce the event rate from 7.6 million Hz to about 300 Hz. The Level 3 trigger, implemented as a "farm" of computers analyzing the whole event record, will further reduce that rate to roughly 30 Hz which can then be recorded for off-line analysis. Extension up to Level 2 and Level 3 output rates of 1000 Hz and 75 Hz, respectively, are envisioned. The amount of computing resources brought to bear on Level 3 processing is specified to be such that the processing time per event is on the order of seconds, rather than hundredths of seconds or less which characterize the first two trigger decisions.

This article describes the development of the subsystems for Run II which assemble the event for Level 3 processing (the "Event Builder") and then deliver the whole event to a single computer for analysis. It must therefore assemble and deliver events at the specified input rate, Hz, though it is also desirable that it be able to operate up to the Level 2 limit at 1000 Hz. The 15 distinct event fragments will each contain on average 12 to 16 KB of data, with the total being around 150 to 250 KB per event. The aggregate data throughput of the system must therefore be at least 44 MB/s, with up to 244 MB/s desirable.

Such high throughput is readily available with commercial network technology. The Run II Event Builder will be based on an ATM switch. The use of inexpensive Pentium-based personal computers, organized into "subfarms" hanging off of Event Builder output ports, is also being investigated for the purpose of satisfying the sizable computing requirements for Level 3.

In this paper only an overview of architecture and components the Event Builder and Level 3 is given. A more detailed description is found in [2].

2 Event Builder Overview

Event data enters the Event Builder system through the Scanner CPU's (SCPU's) and is sent through the event network, which in this system is the ATM switch, to Level 3 via "converter nodes". The flow of data is controlled by the Scanner Manager, which communicates with the Scanners and converters via the separate command network.

Currently, a FORE Systems ASX-1000 non-blocking ATM switch [3] is used as the event network. "Non-blocking" refers to the fact that the switch's internal bandwidth equals the maximum input bandwidth. The switch delivers event fragments from 15 input to 16 output ports and can be extended up to 64 I/O ports total. The I/O ports are connected via OC-3 (155Mbps) optical fiber.

On the input side, MVME2603 processors running VxWorks 5.3 scan data from the VME readout boards. The processors function as FIFO buffers into the switch. The ATM interface is an Interphase 4515 PMC/ATM adapter with 1 MB on-board data RAM. The driver has been developed by our group at Fermilab and implements only the bare AAL5 protocol.

On the output side, Intel processor-based PCs running Linux receive and assemble the event for shipment to the processor nodes. For the test system, 400/500 MHz Pentium II/III processors running Linux 2.0 are used. Before Run II starts new up-to-date PC hardware will be purchased with an updated version of Linux installed. The PC adapter card for the ATM connection is a FORE-Runner [4] LE155 PCI card. The Linux driver is based on a freeware version adapted to the specific application at Fermilab.

The event network data flow is controlled by the Scanner Manager. The Scanner Manager is an independent MVME2604 node running VxWorks 5.3 and is connected to all ATM switch input and output nodes via a command network that is implemented with a SCRAMNet [5] reflective memory ring. Polling is used to test for new messages in the reflective memory. Event data is sent through the ATM switch using the bare AAL5 protocol. In order to prevent cell loss due to output overflows, each sender is allocated a fixed rate for sending to each receiver. While the transmission is going on, other VxWorks tasks load new events and prepare them for shipping so that the ATM interface remains saturated as long as the fragment size is larger then 4 KB [2].

3 Level 3 Overview

The Level-3 trigger is a processor based filtering mechanism which has access to the full event record. The CDF Run II Level-3 trigger is realized as a PC farm. The farm is organized as several "subfarms", each subfarm hanging off of one Event Builder output port. All the PCs in the farm run the Fermilab-supported Linux operating system, based on the Red Hat 5.0 distribution.

A node receiving event data from the ATM switch is called a "converter node". Its function is to assemble the event fragments and distribute them to the "processor nodes" belonging to the subfarm. The processor nodes run the filter algorithm on the event. To cope with the ATM switch output each converter node is connected via two Fast Ethernet ports to a Fast Ethernet switch.

An "output node", connected to several subfarms, collects all accepted events and directs

them to the Consumer Server which is responsible for distributing the data to the data logger as well as on-line monitors.

The full Level 3 system contains 16 subfarms with each subfarm consisting of one converter node and eight processor nodes. Each pair of subfarms is connected to an output node.

In the present test system dual 400/500 MHz Pentium II/III processor PC's are used. The test system consists of 16 converter, 32 processor and 4 output nodes. The final full PC farm will be purchased as close to the start of Run II as possible to obtain the best performance/price ratio.

4 Control and Monitoring

Control and monitoring is a major concern for a system which has to be able to incorporate up to 300 computers. For the initial tests of the Event Builder and Level 3, a prototype control and monitoring program based on the script language *expectk* was developed which was useful to point out certain problems and test concepts. The development of a system appropriate for real operating conditions is underway.

For control and monitoring a CORBA-based mechanism is favored. There are certain restrictions on the choice of CORBA implementation. Two different operating systems have to be accommodated, VxWorks and Linux. In the case of VxWorks, the ORB must fit into less than 1MB memory and provide a C binding. Linux offers more flexibility but must be able to communicate with the VxWorks ORB.

These boundary conditions are relatively tight and from the many candidates only a few survived. After investigating several commercial and free ORBs the following candidates remained: ILU[7], a nice slim product which we ported to VxWorks; and ORBacus [8], also a solid and well implemented ORB which, while not ported to VxWorks, talks to ILU without problems.

4.1 Event Builder

Here we describe the VME section of the Event Builder while the converter PCs are discussed later. The VME CPUs are the Scanner Manager and Scanner CPUs.

The central program for the Event Builder control and monitoring is the Event Builder Proxy. It serves as an intermediate stage between the top-level CDF Run Control software and the VME CPUs of the Event Builder. The function of this Proxy is overall control of the SCPUs and SM; the Proxy also supplies feed-back information to Run Control. The Proxy performs the system state transitions for normal running (e.g.., setup, configure, activate, end), facilitates expert operations and collects monitoring information. The interaction of the Proxy with other components of the DAQ are shown in Fig. 1. Note that, as the Event Builder and the Level 3 farm are on a private network, the Event Builder Proxy is a necessary entity running on the Gateway PC, which is connected to both the internal and external networks.

The communication between Run Control and the Event Builder Proxy is accomplished through SmartSockets [1] while on the internal network, several different protocols are used: CORBA IIOP (using ILU), direct communication through sockets and the Zephyr messaging system[2].

At present, control and monitoring software is written, tested and integration with top-level Run Control is close to completion.

[1] Commercial powerful publish-subscribe messaging system used extensively in on-line software at CDF.

[2] A simple publish-subscribe messaging system developed at MIT.

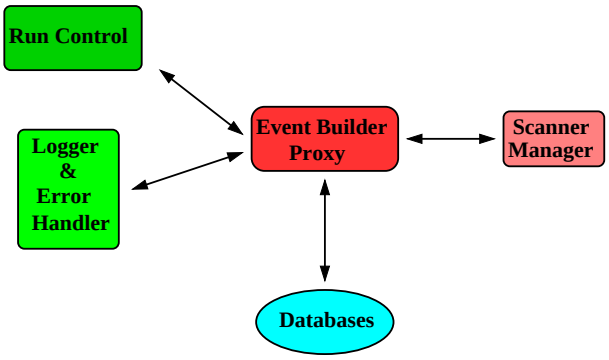

Figure 1: Relations between the Event Builder Proxy and other components of the CDF DAQ system.

4.2 Level 3

For a system of several hundred nodes, each running numerous processes, a carefully designed Control and Monitoring system is necessary. The objective of the Monitoring System for Level 3 is to provide extensive information on all the components of the system with low impact on the event flow.

The Level 3 nodes are also controlled from the Gateway node The framework of the control and monitoring system is based on ROOT [3] extended by representing nodes and configurations. adding appropriate classes for our purposes. We exploit ROOT's fundamental capabilities as an interactive object manager. The farm can be managed conveniently using simple CINT[4] scripts.

For the communication with numerous computers in the internal Level 3 network a "Relay Mechanism" has been developed. Among the main functions of the Relay are starting and stopping processes; requesting and collecting monitored information; and distributing text, script and binary (e.g., executables) files, calibration constants and trigger tables to Level 3 filters. The general design is shown in Fig. 2. Communication to/from a node in a subfarm proceeds through the a designated node that serves as a gateway for a given subfarm. Each node in the Level 3 farm is running an ORBacus-based relay server. Routing tables specify the paths of object request propagation. The relay servers on any given node start and further control the event flow and monitoring components, as shown in Fig. 2.

The monitoring processes on a node collect information from the Level 3 software (such as event count, Event Buffer status), from the operating system (such as process count, load average) and hardware (such as CPU temperature, voltages). To provide shift users with relevant information, and allow experts detailed access to node information, the Monitoring system has been divided into two subsystems. These are Steady State Monitoring (pushed out periodically, see Fig. 3) and Expert Monitoring, both utilizing the Relay Mechanism.

The communication between the event flow processes and monitoring processes occurs through shared memory segments, reducing to minimum the interference with the event processing . Monitored information is transmitted in the form of a string using the CORBA IIOP protocol, with designated "concentrator" nodes serving as intermediate gateways for the Steady State Monitoring, as shown in Fig. 3. The concentrator and relay nodes are not necessarily the same. At its final destination, the information is converted into ROOT objects for display and storage purposes.

At present, Monitoring and Control systems for a large farm have been prototyped. Testing on the currently available system (up to 40 nodes) has been successful. Work on integration with

[3] ROOT is an object oriented analysis framework developed at CERN (`http://root.cern.ch/`).

[4] CINT is a simple scripting language used by ROOT.

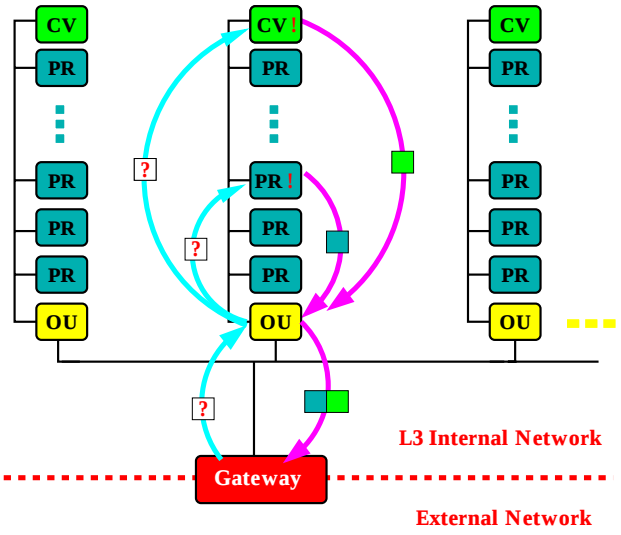

Figure 2: Schematic drawing of the Level 3 Relay mechanism.

top-level CDF Run Control is underway.

5 Integrated DAQ System Tests

As the system expands towards full size we can test how the performance scales. In order to study properties of the integrated Event Builder/Level 3, simulated events are processed by the system testing throughput and general behavior. The events are simulated at a single Scanner CPU. Up to eight converter nodes receive data. The results of these tests are shown in Fig. 4. The rate per converter is, to a good approximation, independent of the number of converters and the rates indicate that the baseline requirements are met. The maximum observed rate is 511 Hz.

Among the most important tests is the one exercising the full data flow chain, starting with detector components all the way to storing the data to disk. Data are read out from three subdetectors: a section of Central and Plug Calorimeter, a TDC crate of the Central Outer Tracker and several components of the SVX detector. The events are received by the SCPUs and assembled through the Event Builder in a single converter node, analyzed by a Level 3 Filter executable at the processor node, and sent via the output node to the Consumer Server/ Data Logger (CS/DL) [6]. The CS/DL saves data to disk and also forwards some events to Consumers, such as the Event Display Consumer. This connectivity test has been successful. However, only a part of Level 3 and Event Builder system has been exercised in this test due to the limited amount of data supplied by Front End crates.

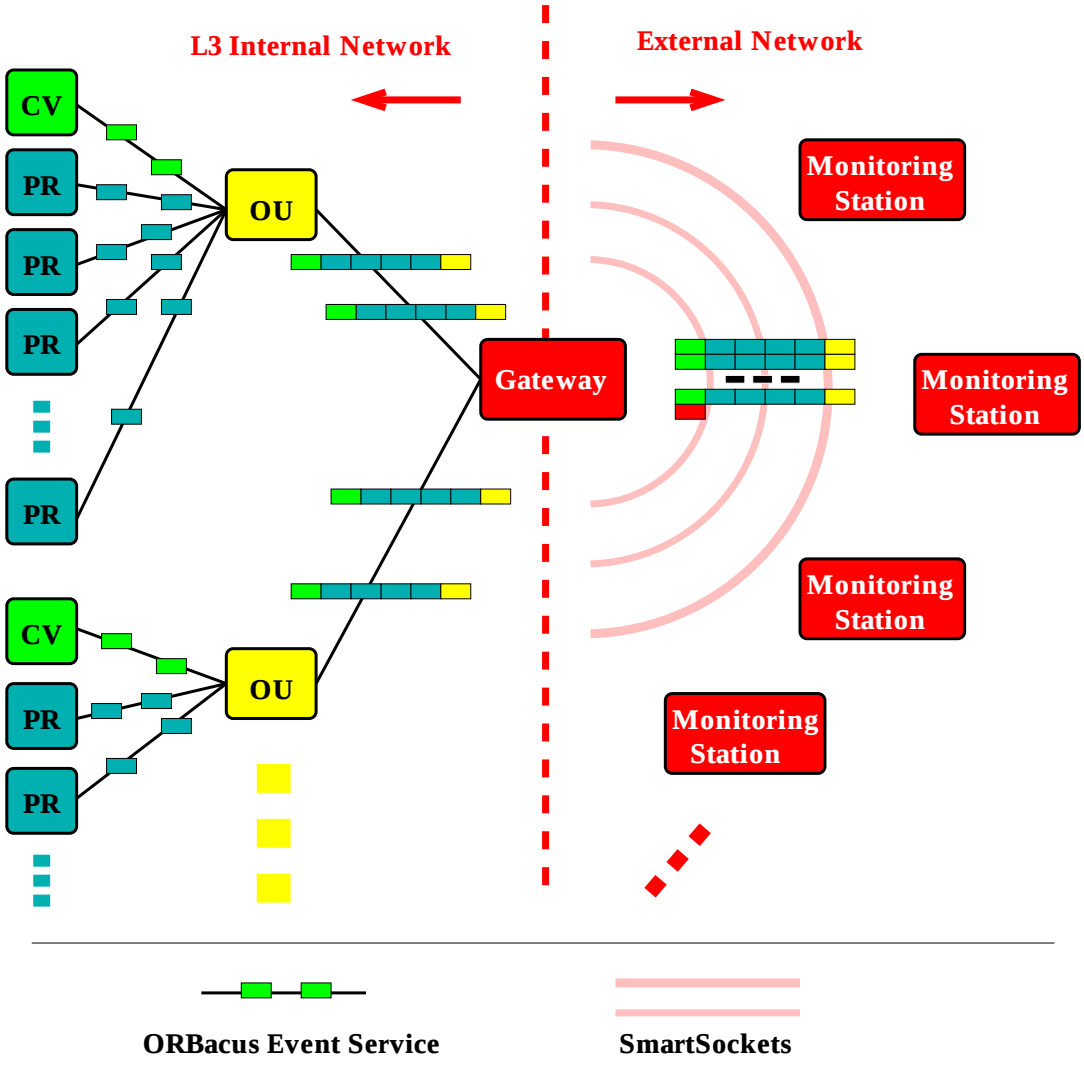

Figure 3: Steady State Monitoring diagram.

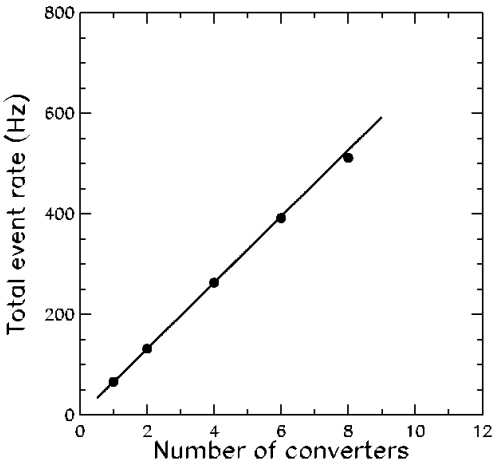

Figure 4: Event rate for the combined Event Builder and Level 3 test. The total rate is shown versus the number of converter nodes.

6 Summary and Conclusions

This paper presents the current status of the Event Builder and Level 3 systems of the CDF experiment. We briefly describe the hardware as well as some of the more recent software development.

The Control and Monitoring system for the Level 3 and Event Builder is discussed in some detail. The basic functionality of this CORBA-based system has been implemented and tested, and is currently being developed and refined.

Large-scale performance tests show that the system exceeds the requirements of the CDF technical design report [1]. Further improvements and upgrades are on the way, and prototype systems are being expanded for more thorough tests.

References

1 F. Abe *et al.* (CDF Collaboration), *Nucl. Intrum. Methods* **271**, 387 (1988). F. Abe *et al.* (CDF Collaboration), *Phys.Rev.* D **50**, 2966 (1994). The CDF II Collaboration, *The CDF II Detector: Technical Design Report*, FERMILAB-PUB-96/390-E, October, 1996.

2 J. Fromm *et al.*, *ATM Based Event Building and PC Based Level 3 Trigger at CDF*, International Conference on Computing in High Energy Physics (CHEP 98), Chicago, IL, August 31 - September 4, 1998; also available as: FERMILAB-CONF-98/348-E.

3 FORE Systems, ForeRunner *ATM Switch Architecture*, April, 1996.

4 ForeRunner is registered trademarks of FORE Systems, Inc., 1999; for further information: www.fore.com.

5 SCRAMNet is registered trademarks of Systran Corporation, 1999; for further information: www.systran.com.

6 T. Watts and S. Lammel, *Overview of CDF Run II Data Handling System*, presentation at this conference.

7 Documentation on ILU can be obtained from the Xerox web site `ftp://ftp.parc.xerox.com/pub/ilu/ilu.html`

8 Information on ORBacus is found at `www.ooc.com/ob` web site.