Fermi National Accelerator Laboratory

FERMILAB-Conf-98/359

Application of PC's and Linux to the CDF Run II Level-3 Trigger

Jim Fromm et al.

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

December 1998

Published Proceedings of the *International Conference on Computing In High Energy Physics (CHEP '98)*, Chicago, Illinois, August 31 - September 4, 1998

Operated by Universities Research Association Inc. under Contract No. DE-AC02-76CH03000 with the United States Department of Energy

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, expressed 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.

Distribution

Approved for public release; further dissemination unlimited.

Copyright Notification

Application of PC's and Linux to the CDF Run II Level-3 Trigger

Jim Fromm, Don Holmgren, Robert Kennedy, Jim Patrick,
Don Petravick, Ron Rechenmacher, G.P. Yeh
Fermi National Accelerator Laboratory
Gerry Bauer, Paraskevas Sphicas, Steve Tether, Jeff Tseng
Massachusetts Institute of Technology
Benjamin Kilminster, Kevin McFarland, Kirsten Tollefson
University of Rochester

October 15, 1998

Abstract

For Run II, the CDF Level-3 trigger must provide a sustained input bandwidth of at least 45 MBytes/sec and will require processing power of at least 45000 MIPS to perform the necessary reconstruction and filtering of events. We present a distributed, scalable architecture using commodity hardware running the Linux operating system. I/O and CPU intensive functions are separated into two types of nodes; "converter" nodes receive event fragments via ATM from Level 2 computers and distribute complete events to "processor" nodes via multiple fast ethernets. We present results from a small-scale prototype roughly equivalent to a 1/16th vertical slice of the final system. With this hardware we have demonstrated the capability of sustained I/O rates of 15 MBytes/sec, more then three times the required baseline performance. We discuss PC hardware and Linux software issues and modifications for real time performance.

Commencing in the year 2000, Run II of the Collider Detector at Fermilab [1] (CDF) will require I/O and computational capabilities a factor of at least twenty beyond Run I usage. The CDF Level-3 trigger has the responsibility of reading event fragments from Level-2 readout buffers via an ATM network, assembling and reordering the fragments into full events, performing on-line reconstruction, and selecting events based upon various trigger criteria for archival storage. Details of event building using ATM are presented in a separate paper.[2]

Table 1 lists various specifications for the Level-3 trigger. Level-3 will accept events from Level-2 at a peak rate of 300 Hz. Because of various factors related to Level-2, this peak acceptance rate may increase to 1 KHz. Further, the mean event size may increase to 250 KBytes. The CPU requirement listed in the table

Specification	Requirement
Level-2 Peak Trigger Rate	300 Hz
Mean Event Size	150 KByte
Maximum Level-3 Output Rate	15 MB/sec (75 Hz)
CPU	45,000 MIPS
# 500 MHz Dual-PII Nodes	80

Table 1: CDF Run II Level-3 Specificationsl

is estimated from on-line reconstruction CPU usage per event in Run I.[1] The number of dual 500 MHz Pentium II nodes is extrapolated from the measured performance of various Pentium II processors.[2] Finally, the maximum Level-3 output rate is fixed at 15 MB/sec by the mass storage budget.

Because of the possible increases in data rate into Level-3, the architecture of the trigger should scale. Commodity components are preferred to maximize the cost effectiveness of the system. In Run I, a single layer approach was used, in which Level-3 computers received events from a network switch (Ultranet), performed reconstructions, and output filtered events to storage. For Run II, the computational requirements prevent this approach with Intel processors. Instead, a distributed architecture will be used.

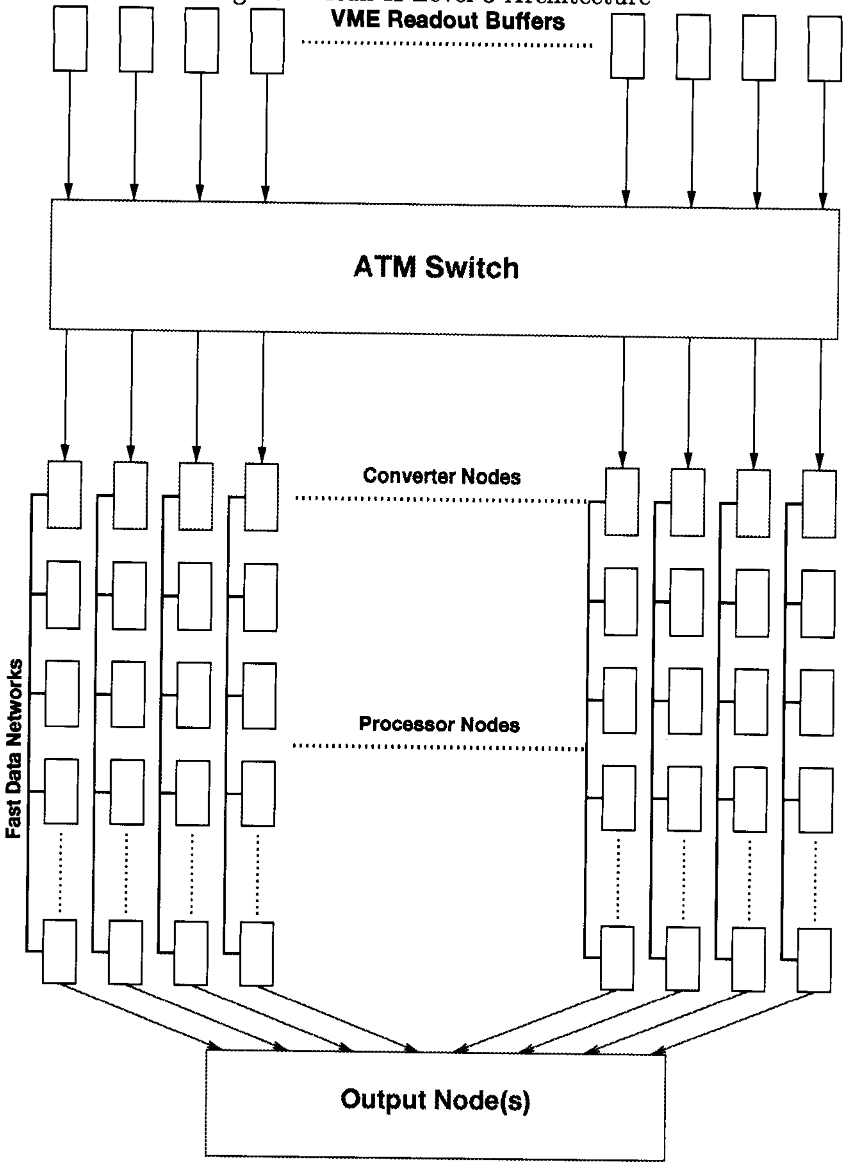
Figure 1 shows the distributed design, consisting of subfarms each connected to one of sixteen ATM switch ports. *Converter nodes*, containing ATM interfaces, accept event fragments sent from Level-2 readout buffers through the ATM switch (*Fore Systems ASX-1000*). Level-2 and Level-3 communicate using *SCRAMnet* reflective memories (not shown). Full events assembled from the fragments are sent to one of many *processor nodes* which perform reconstruction and filtering. Filtered events are sent to one or more *output nodes*, which send the data to mass storage and to other consumers. The peak data rate into each converter node is limited to about 16.5 MB/sec (ATM OC-3 data rate). Each converter node is connected to its subfarm via a four port fast ethernet interface connected to a fast ethernet switch. This layered approach effectively separates the I/O and computational tasks into the converter and processor node groups. To increase computational capability, additional processor nodes may be added to each subfarm; such addition has no effect on the converter nodes. To increase I/O capability, the ATM switch can be upgraded to provide either more output ports or higher data rates per port (*i.e.* OC-12).

A prototype subfarm consisting of one converter and four processor nodes was assembled using Intel-based computers. The converter node initially used a 200 MHz Pentium Pro processor ("FX" PCI chipset), 64 MB of fast page mode memory, an ATM interface (*Fore Systems ForeRunnerLE*), a SCRAMnet interface, and a four-port fast ethernet interface (*Adaptec ANA-6944A*). This

[1] MIPS are measured using Fermilab's *Tiny* benchmark, and correspond roughly to VAX-780 MIPS

[2] *Tiny MIPS* have scaled with processor speed on Pentium Pro and Pentium II processors. 350 MHz Pentium II processors have a 197 MIPS rating.

Figure 1: Run II Level-3 Architecture



node was later upgraded to a 300 MHz Pentium II processor ("LX" PCI chipset), then a 350 MHz Pentium II processor ("BX" PCI chipset). The processor nodes each contain two 200 MHz Pentium Pro processors, 64 MB of memory, and a fast ethernet interface (*SMC EtherPower 10/100*). All nodes also contain an EIDE disk drive and a standard 3.5" floppy drive.

The Run I Level-3 software framework was ported to run on the prototype. The nodes run the Linux operating system, specifically the Fermilab-modified distribution based upon RedHat 5.0.[3] The Linux kernel on the converter node is patched to support ATM using the package available from EPFL.[4] This package is well written, documented, and supported. A number of ATM PCI interfaces were evaluated, with the *Fore Systems ForeRunnerLE* chosen because of its capability to support ATM VPI values greater than zero. This interface is based upon the chipset (*"Nicstar"*) manufactured by IDT. The Nicstar driver included in the EPFL package was modified to support the *ForeRunnerLE.* [3]

Before running the full Level-3 framework, simpler tests were run on the prototype to assess the I/O capabilities of the converter node. These were open loop tests. Level-2 nodes sent event fragments to the converter node using rate division and/or interpacket delay to adjust input data rates. These fragments were either sent directly to processor nodes or, using multiple processes, shared memory, and message queues, assembled into full events which were then sent to processor nodes.

Several problems were revealed by these early tests. First, the converter node often dropped incoming event fragments at the start of a simulated run and when other system processes were started. These drops were caused by exhaustion of ATM socket buffers and were eliminated by kernel modifications and the use of real time scheduling queues. The modifications disable the "slow start" algorithm, which throttles TCP communications rates after ethernet sockets were first opened to processor nodes, and increase the maximum size of socket buffers. The real time scheduling queues, accessed via the *sched_setscheduler()* system call, give the ATM receiver process on the converter node priority over all other processes whenever data are available.

Second, the maximum data throughput was found to be limited by the memory bandwidth of the converter node. When moving event data from an ATM socket to a processor node via an ethernet socket, a total of eight data movements on the converter's memory bus are required.[4] Thus 15 MB/sec of throughput requires up to 120 MB/sec of memory bandwidth (exact amount depending upon details of memory cache). The performance of various memory types were measured using the *STREAMS* memory bandwidth benchmark (see Table 2).[5] By switching the converter node to a Pentium II processor with 100 MHz SDRAM, data throughput matching the maximum ATM OC-3 data rate was achieved.

The Level-3 software framework was also tested on the prototype. One

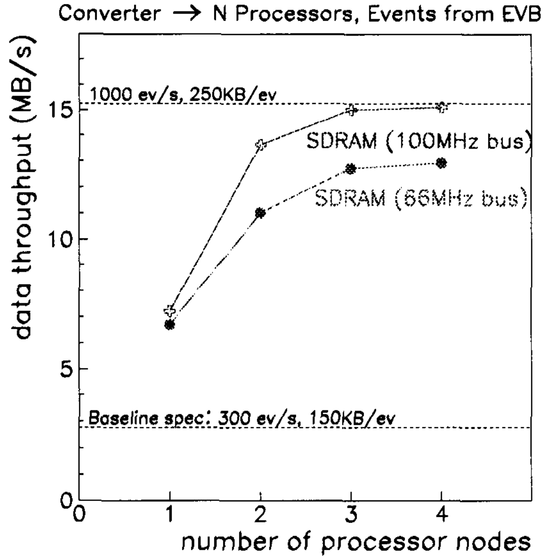
[3] *ForeRunnerLE* driver is available from ftp://linux-rep.fnal.gov/pub/drivers/nicstar/.

[4] ATM interface DMA to kernel buffer (1 move), kernel buffer to user buffer (2 moves), user buffer to kernel buffer (2 moves), kernel buffer to multiple kernel buffers (fragment to ethernet MTU) (2 moves), and finally kernel buffer DMA to ethernet interface (1 move).

Table 2: *STREAMS* Memory Bandwidth by Memory Type

Fast Page Mode (FPM)	80 MB/sec
Extended Data Out (EDO)	108 MB/sec
66 MHz Synchronous DRAM	170 MB/sec
100 MHz Synchronous DRAM	260 MB/sec

Figure 2: Throughput of Full Level-3 Framework



further kernel modification, changing the time slice from 10 msec to 1 msec, was found necessary to maximize throughput and to minimize variations. Some sections of the framework code rely on polling, using the *nanosleep()* system call to delay between tests. In the Linux kernel the minimum sleep achievable is typically one to two time slices, even though shorter sleeps are requested. By decreasing the time slice to 1 msec, more consistent data throughputs were achieved.

Figure 2 shows the throughput of the prototype running the full Level-3 framework as a function of the number of processor nodes configured to request events. The horizontal dotted lines at bottom and top show the required baseline (*i.e.*, 300 events/sec, 150 KB/event, 16 subfarms) and desired enhanced (*i.e.*, 1000 events/sec, 250 KB/event, 16 subfarms) data throughputs. The performance differences between 66 MHz and 100 MHz memory buses, with 300 MHz and 350 MHz Pentium II processors respectively, are shown.

In conclusion, the prototype Level-3 subfarm has demonstrated data throughput of over 15 MB/sec, greatly exceeding the baseline requirement (3.8 MB/sec, assuming 16 subfarms). Commodity computers based upon Intel processors were used, and for the converter node it was necessary to use a system with the fastest currently available memory system (100 MHz SDRAM). The distributed architecture is scalable. Computational capacity can be increased by adding or upgrading the processor nodes. Linux has proven to be a good choice for the trigger. It has adequate quasi-real time facilities, as required by the converter node. Minor kernel networking modifications were necessary to maximize data rates. Such modifications are easily accomplished in an open source environment like Linux. ATM extensions to the Linux kernel from EPFL proved to be reliable, and a driver to support the *ForeRunnerLE* ATM interface was written.

Expansion of the prototype system to four subfarms is in progress. Future testing will go beyond demonstration of I/O capabilities to include computational performance on the processor nodes.

References

[1] F. Abe *et al.* (CDF Collaboration), *Nucl. Instrum. Methods* A **271**, 387 (1988); F. Abe *et al.* (CDF Colaboration), *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," CHEP'98 talk 213.

[3] D. Skow *et al.*, "Linux Support at Fermilab," CHEP'98 talk 220.

[4] W. Almesberger, "ATM on Linux", http://lrcwww.epfl.ch/linux-atm/.

[5] J. D. McCalpin, "STREAM: Measuring Sustainable Memory Bandwidth in High Performance Computers", http://www.cs.virginia.edu/stream/.