CDF/DOC/LEVEL-3/PUBLIC/4551
Version 1.1
June 3, 1998

Prototype of a PC Farm for the CDF Run II Level-3 Trigger

Jim Fromm, Don Holmgren, Robert Kennedy,
Don Petravick, Ron Rechenmacher
Fermi National Accelerator Laboratory
G.-P. Yeh
Fermi National Accelerator Laboratory and Academia Sinica
Gerry Bauer, Kevin McFarland, Paris Sphicas, Sham Sumorok,
Steve Tether, Jeff Tseng
Massachusetts Institute of Technology

June 3, 1998

Abstract

We present results from a small-scale prototype of a design for the CDF Run II Level-3 trigger using commodity PCs to provide the processing CPUs.

Contents

1 The Run II Level-3 System

The Level-3 trigger serves many functions in the CDF Run II data acquisition and trigger system. Its primary function is to act as a filter for reducing the number of events written to serial media. Level-3 will also provide the initial event sorting into streams for offline production, for online monitoring, or for high-priority analysis. Finally, because Level-3 will run code similar to the offline reconstruction code, it can also serve as a computing resource, performing some of the initial tasks in data analysis and reconstruction, possibly including full analysis of some calibration and monitoring samples.

The basic functional elements of Level-3 are shown in Figure 1. Event segments from each scanner CPU are built directly into the main memory of the Level-3 computers. The event reformatter then builds the event segments into a contiguous block of memory in the global event buffer system while grouping the detector elements that may have been read out via multiple scanners. From the global buffers, events are examined in the filter software, where the executable may decide to discard an event, discard parts of an event or add information from its calculations into the event. Events remaining after selection by the filtering software are then shipped to serial media and to other online monitoring processes by the dispatcher. These tasks are processor-intensive, and require significant I/O and memory bandwidth.

For the success of CDF Run II, it is crucial that the Level-3 system be able to provide as much event throughput as possible, with the TDR specifications shown in Table 1 providing a lower bound. Furthermore, this

Level-2 Peak Trigger Rate	$\geq$300 Hz, $\leq$1 kHz
Mean Event Size	200 kB
Input Sources to Switch	14–15
Required Input Bandwidth	$\gtrsim$60 MB/s
Level-3 Output Rate	$\leq$75 Hz
Equivalent R4400 CPU time	1.1 sec/event

Table 1: Current best estimate of Level-3 Specifications from the Run II TDR, subsequent re-evaluation[1], and the Run II Data Needs Assessment Document[2]

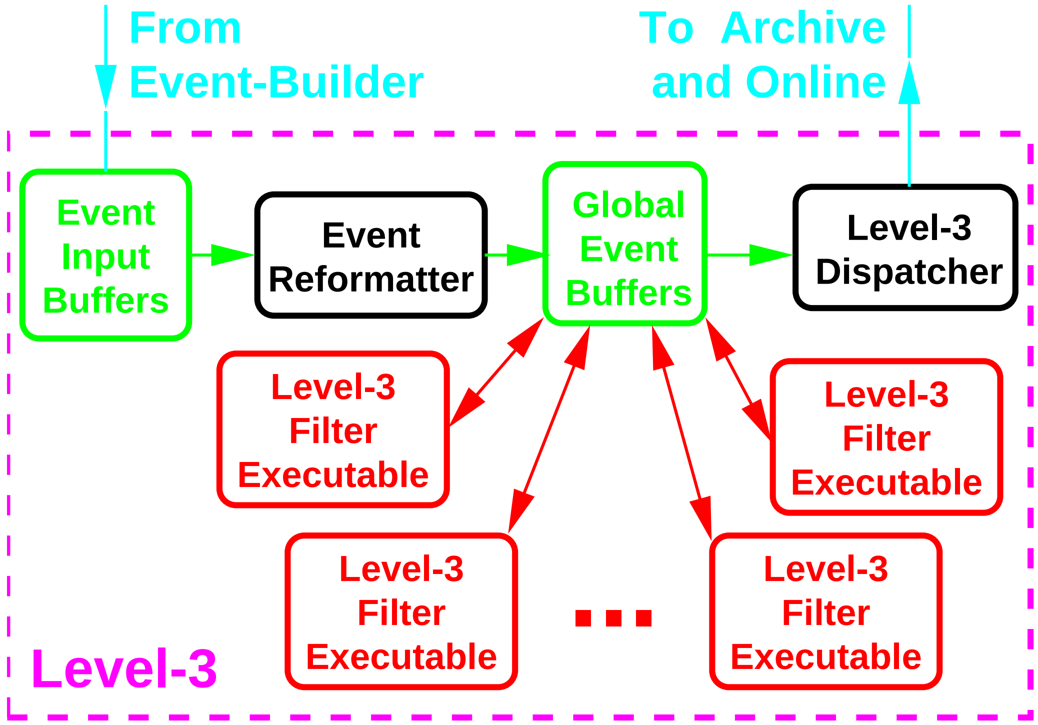

Figure 1: A functional block diagram of Level-3

system will have, by definition, no excess capacity; every bit of processing power and bandwidth in the system can be used to improve the physics from CDF. As a corollary, any improvement to algorithms which can reduce the CPU time required to process an event without sacrificing functionality will also improve the physics output of CDF Run II.

A reasonable upper limit on desired data throughput of a baseline Level-3 comes from considering a modest increase of the CDF event size (to 250 kB) and a reasonable limit on the front-end readout rate, 1 kHz. This would require a total data throughput of 244 MB/sec through the entire system, which is well-matched to the possible measured throughput of 16 155 Mbps outputs from the event builder switch (240 MBps). This implies that Level-3 should provide a data bandwidth somewhere between 2.75 and 15.2 MBps per sub-farm (assuming 16 switch outputs feeding 16 sub-farms).

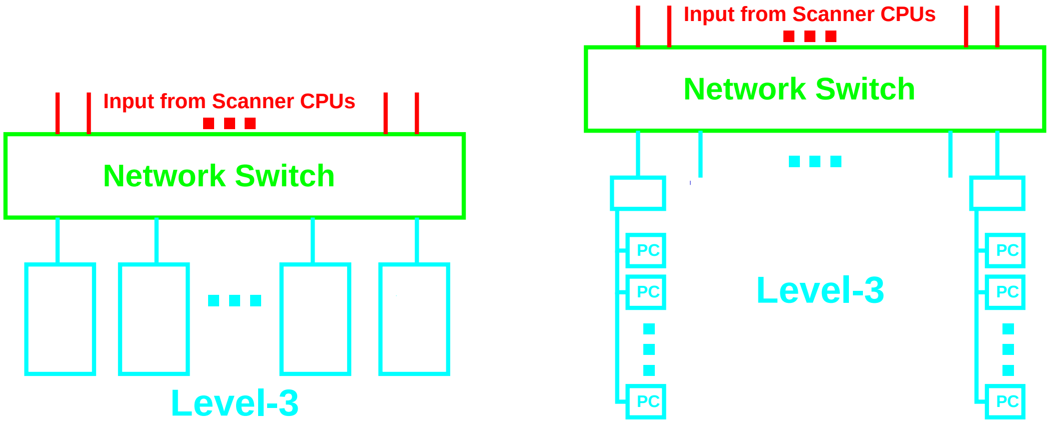

Figure 2: Few-box and multi-box architectures for Level-3

1.1 Level-3 Hardware Options

The proposed architectures for Level-3 can be categorized as either multi-CPU platforms or as sub-farms of lower capacity computers. The two approaches are shown schematically in Figure 2. In Run Ib, CDF used a Level-3 farm based on the former approach, shown in Figure 3. The Run Ib farm consisted of 4 Silicon Graphics Challenge XL systems, each with 8 R4400/200MHz processors. This system ran with a Level-3 input rate of $<$ 30 Hz and each event took 4/7 CPU-sec.

The cost of purchasing a similar system, based on either the Silicon Graphics Origin2000 line or Digital Equipment Corporation 8200 5/625 systems would be about $15K per processor. With the processors available today (SGI R10000/200MHz or DEC Alpha 21164/625MHz), about 140 processors would be needed to provide the equivalent of twice the CPU per event used in Run Ib, assuming 300 Hz input. The I/O capabilities of these systems are impressive, but well beyond the scope of what is required for CDF.

A second alternative would be to buy a large cluster of O200 workstations. This platform currently has a limit of 4 processors per node, so approximately 35 nodes would be required. Therefore either an internal Level-3 network or more event builder switch outputs would be required in order to obtain the baseline computing capacity. The cost is approximately $7.5K per R10K processor, a significantly better deal for CDF's needs than the Origin 2000 platform. Although such a node has not been tested, it should have adequate

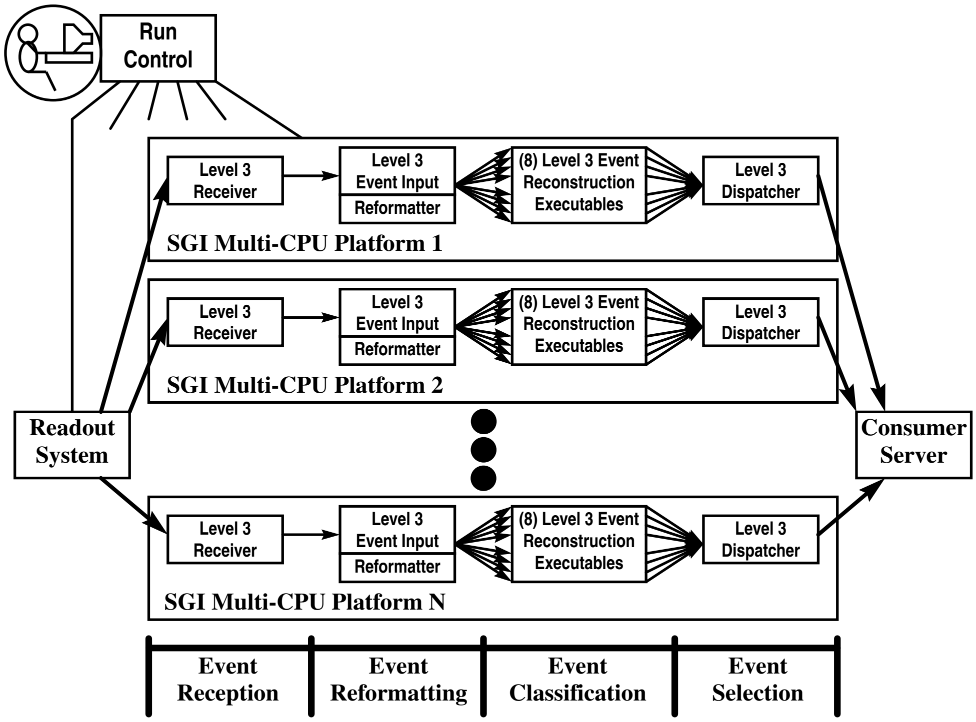

Figure 3: The Level-3 system used in CDF Run Ib.

memory and I/O resources to perform event throughput and processor tasks simultaneously.

However, significantly better price to CPU-performance ratios can be achieved in commodity desktop workstations. The CPU usage of CDF analysis code has been compared on an SGI R10000/180MHz CPU and an Intel P6/200MHz CPU. The required CPU time per event is approximately 2.3 times greater in the P6 processor than in the R10K[3]. However, a dual-P6-based Level-3 computing node would cost about 6 times less than an R10K CPU as part of an Origin 2000 system, and 3 times less than an R10K processor in the O200 platform. In principle, then, by going to low-priced, commodity PCs, a significantly greater amount of computing capacity can be purchased for Level-3.

1.1.1 CPU-Time in Level-3

If a Pentium-based PC architecture is chosen, one might conservatively speculate that the computing power available for Level-3 would be equivalent to approximately 500 P-II/400MHz processors (approximately 1000 R4400/(200MHz,4MB L2) equivalents). This would imply 3.3 R4400-CPU-sec/event available in Level-3 at a 300 Hz input rate. Past experience with the CDF outer tracker analysis in Level-3 was that it consumed approximately half of the available CPU. Even with much more CPU available, the fraction of the CPU consumed by tracking will be substantial in Run II because of the increased granularity of the Outer Tracker and the desire to track in the Silicon detectors at Level-3. However, no quantitative measures of the performance of the offline code in Level-3 are possible at this time because the state of the code does not allow it.

2 Architecture of the Prototype Farm

Initial consideration of PCs for the CDF Run II Level-3 trigger was motivated by their superior price to CPU-performance ratios and by the anticipation of the widening of this advantage as newer generations of commodity hardware were released. PCs did not, however, fit into the "few-box" model used in Run 1b and shown in Figure 2 because of the anticipated limit of sixteen Level-3 nodes imposed by the cost of output ports on the network switch. Commodity multi-processor motherboards typically have only two, and at most four, Pentium Pro processors. The 64 Pentium Pro processors which could be assembled in 16 Level-3 nodes would fall far short of Run II computational requirements. The "multi-box" architecture of Figure 4 provides a scalable solution to this problem. Level-3 nodes in this architecture are one of three types: "converter", "processor", and "output". One converter, multiple processors, and one output node constitute a "leg" of the PC Farm Level-3 architecture.

Converter nodes in this prototype are PCs equipped with ATM interfaces. They accept event fragments via the ATM switch from the Level-2 scanner CPUs. The fragments of any given event are assembled and shipped via a local network to a processor node which has advertised a free global buffer. The converter nodes operate with real time constraints and are intended to sink without loss the entire 155 Mbps input bandwidth of the ATM OC3 connection to Level-2. Because of the relatively small number of boxes, the cost impact of the converter nodes is small; therefore, they need not be PCs in the final system.

As many processor nodes can be added to a Level-3 leg as are necessary to filter events at the required rate. If the required input bandwidth to Level-3 should rise because of increase in event size or Level-2 peak trigger rate, or if the CPU requirements of the delivered Level-3 filter code are higher than anticipated, this architecture will scale via the addition of extra nodes. Further, any such expansion of the number of processor nodes has no effect on the I/O load on the converter node.

Output nodes are responsible for communicating with the online and data-logging systems. Again, these nodes need not necessarily be PCs, depending upon the communications medium chosen by Fermilab for Run II data storage.

Level-3 requires significant I/O and memory bandwidth as well as large computational capability. For example, CDF event reformatting during event building at the 155 Mbps maximum input rate can impact a converter node's ability to perform other tasks because of the consumption of memory bandwidth. Distributing memory- and computation-intensive tasks to the processor nodes allows the converter node to guarantee I/O performance.

2.1 Level-3 Internal Networks

By far the most cost effective network hardware available is fast ethernet, which is capable of reliable communication between computers at up to about 10 MBps. Since the ATM network switch which sends event data into Level-3 operates at up to about 16 MBps, two fast ethernet networks are required to sink the maximum input data rate from the converter to the processor nodes. The output data from the processors provides an additional load which can, in principle, be handled on one of the fast ethernets, or on a third fast ethernet network. Note that should the effective filtering rate be poorer than 75% and flow into the Level-3 leg be at the maximum of 16 MBps, that the dual ethernets will not be sufficient.

Ethernet is not only an inexpensive hardware medium, but with TCP/IP it has a well-developed, robust and reliable software protocol as well. The use of multiple IP addresses within a UNIX node is commonplace and largely

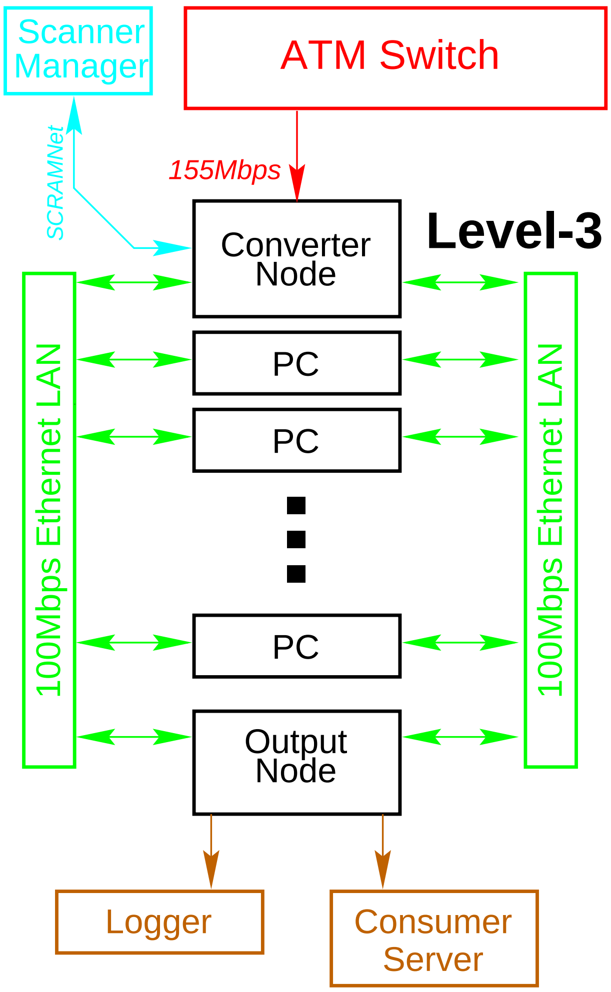

Figure 4: A sub-farm in the PC Farm architecture for CDF Level-3. A converter node receives events from the switch, communicates with the scanner manager and distributes events to Level-3 processors. After the processor accepts an event, it is transferred to an output node which transfers events to the logger and the consumer server.

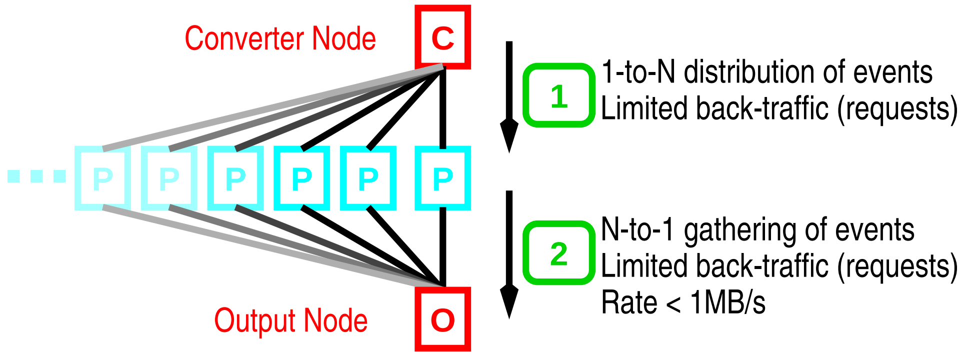

Figure 5: Ethernet traffic internal to Level-3. The majority of the flow is uni-directional, either from converter to processor or from processor to output node..

transparent to user applications.

As shown in Figure 5, data flows on the internal network in Level-3 are asymmetric. The majority of bytes transferred are contained in the events, which all flow from the converter node to the processor nodes, or from a processor node to the output node. Requests for events and acknowledgements from the processor nodes to the converter node are tiny in volume by comparison. Because of this traffic pattern the use of multiple simple repeating hubs is possible. However, because of the need for two 100Mbps networks to sink the total possible ATM rate through the converter node and because only two fast ethernet interfaces were installed on each processor node, we chose to purchase a fast ethernet switch for these tests. An alternate solution would have been to install a three physical LANs in the Level-3 system; two for converter to processor traffic and one for processor to output traffic. The cost per port actually is similar for the two solutions; approximately $100 per port for the hub and $120 per port for the switch.

Because ethernet communications to and from the outside world are required, one of the two LANs (actually VLANs set up by partitioning of the ethernet switch) is "public" and is accessible from the Fermilab wide area network. The other network is "private", with no connections external to the Level-3 cluster. The Level-3 networks are both fast ethernet; the Fermilab WAN is conventional 10 Mbps ethernet. A repeating hub used to link Level-3 networks to the Fermilab WAN internally has a switch between the 100 Mbps

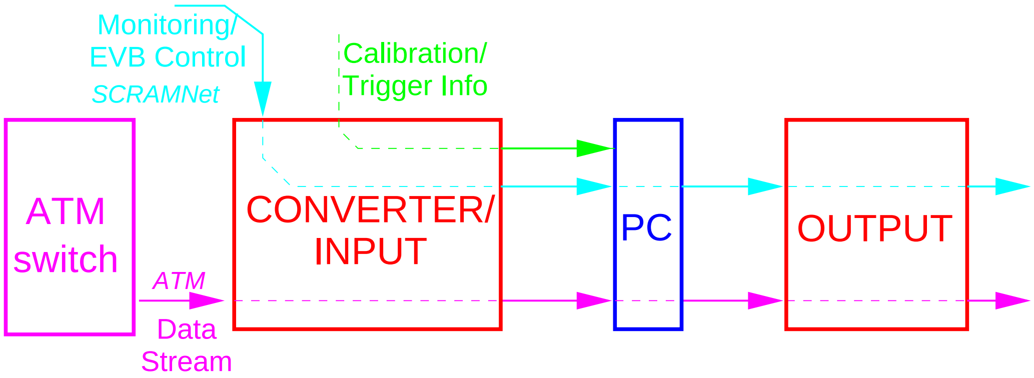

Figure 6: Data flow in a vertical slice of Level-3 in the multi-PC architecture. The flow of three streams, the event data stream, monitoring information and calibration and trigger information, are shown. In the multi-CPU box Level-3 architecture, converter, processing and output functionality could be merged into a single node.

and 10 Mbps connections. This switch in effect isolates the Level-3 internal networks and nodes from Fermilab WAN traffic not addressed specifically to those nodes.

Another type of internal network, SCRAMnet, connects the converter node to computers on the Level-2 side of the ATM network switch. Via this network the converter node communicates the availability and location of free global event buffers in the Level-3 leg.

2.2 Data Flow for Monitoring and Calibration

A diagram of the data flow among different components of the Level-3 system in the multi-PC architecture is shown in Figure 6. In a manner consistent with the direction of event traffic, it is proposed that the data input into the farms be distributed *via* the converter nodes and that monitoring data be reported to the output nodes. In general, calibration and other monitoring traffic either is low-volume or occurs only during an initialization phase, such as at the beginning of a run.

3 Implementation of the Level-3 Prototype

3.1 Linux Operating System

Although several operating systems are available to run on PCs, Linux was chosen for a number of reasons. These included:

- Familiarity with coding and administration issues.
- Portability of existing software to a standard Unix environment.
- Use of standard tools for development and debugging.
- Operating system source code access.
- Cost.

Most of the code used in the Level-3 implementation was ported from the Run 1B framework. As seen elsewhere[3], this port to Linux was not difficult.

Previous to the Level-3 implementation, a low level operating system "TRACE" tool was developed for the Linux operating system. This tool and the experience gained in its implementation proved helpful in debugging the Level-3 system. TRACE allows detailed performance characterization of user and kernel codes. Since the Linux operating system source code is available, any problems revealed in testing can be resolved with modifications to the Linux kernel.

As an example, early testing of ATM to ethernet data transfers on the converter node indicated that some input event fragments were lost. The test code contained a tight loop which read data from an ATM input socket and wrote the data to an ethernet output socket. Analysis of the running system with TRACE revealed that the process occasionally was not scheduled to run in time to process data waiting in the ATM socket buffer. Also, at the start of ethernet communications the output data rate was seen to be throttled, eventually increasing to the expected maximum. Both mechanisms resulted in ATM buffer overflow and data loss.

Access to the Linux scheduler source code revealed how to defeat the first mechanism. Linux has POSIX-compliant real-time scheduling queues which may be enabled on a per process basis. Important processes placed in one of the real-time queues will run whenever ready, pre-empting all normal processes. By placing the test program in a real-time queue and also increasing the size of the ATM input buffer, overruns caused by scheduling were eliminated.

The second mechanism, throttling of the output ethernet data rate, was found to be a documented feature of the TCP/IP protocol, known as "Jacobson Slow Start"[4]. During TCP communications, the sender and receiver negotiate a maximum size for data windows, with each window acknowledged before a later window can be sent. The slow start algorithm prevents swamping an ethernet router which may be present between the sender and receiver. The Linux kernel was modified to disable this throttling, which is not a relevant precaution for the internal networking in the Level-3 prototype.

Debugging such problems in a programming environment without access to operating system source codes and advanced tools would be much more difficult.

3.2 Network Driver Development

The native Linux kernel does not include support for ATM networking. The "ATM on Linux" package from EPFL[5], consisting of patches to the kernel, drivers for a number of common ATM interface cards, and various test and utility programs, was installed on the prototype converter node. Receipt of AAL5 packets from the ATM network switch was successfully tested using ATM interfaces from several vendors.

Many of the available ATM interfaces for PCs do not support the full range of ATM addresses. In particular, most interfaces allow VPI values of only zero. Existing Level-3 codes[6] rely on VPI values greater than zero to specify the output port of the ATM network switch and to indicate the source of event fragment data. Only ATM interfaces based upon the IDT Nicstar chipset were found to support non-zero VPI values. One current and inexpensive ATM interface using this chipset is the FORE Systems ForeRunner LE board.

The EPFL ATM package does not directly support the ForeRunner LE interface. Fortunately, one of the drivers included supported a different board based upon the same Nicstar chipset. After consultation with FORE Systems the Nicstar driver was modified to support the ForeRunner LE interface. Further modifications to the driver were made to eliminate inefficiencies in

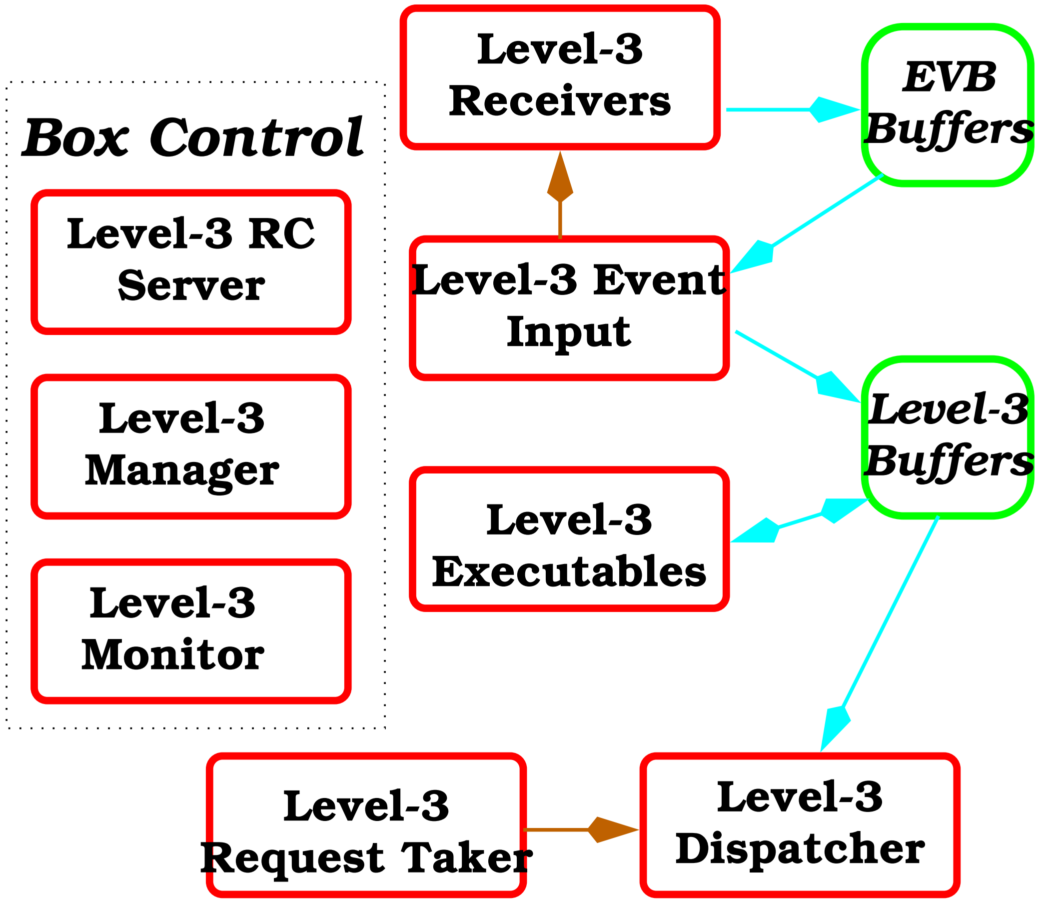

Figure 7: Functional block diagram of the RunIb control code. The various processes running in Level-3 are shown, along with the path of events through the system.

memory usage.

3.3 Level-3 Control and Receiver Code

A large subset of the functionality needed for a realistic CDF Level-3 has already been incorporated into control code running on the PC prototype. This potentially daunting task was accomplished primarily via a straight port of the CDF Run Ib Level-3 from IRIX to first Red Hat 4.2 and then Red

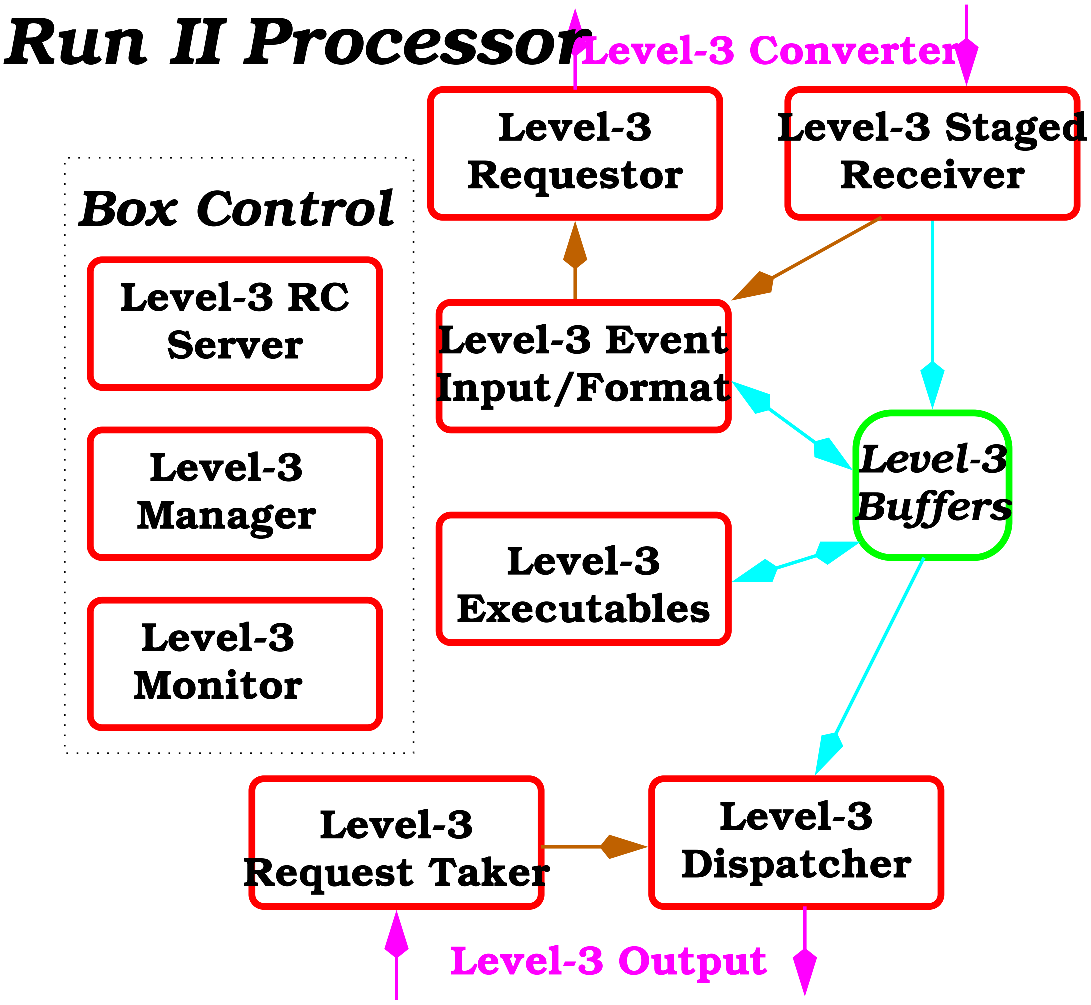

Figure 8: Run II Level-3 processor node control code for the PC architecture. Event reformatting and filtering take place here.

Hat 5.0 distributions of Linux. The similarity of IRIX and Linux operating systems made this port reasonably straightforward. A diagram of control code pieces and the path of events through the Run Ib system is shown in Figure 7.

Significant modifications were required in order to pass events among boxes in the PC farm architecture. As shown in Figure 8, the Run II code

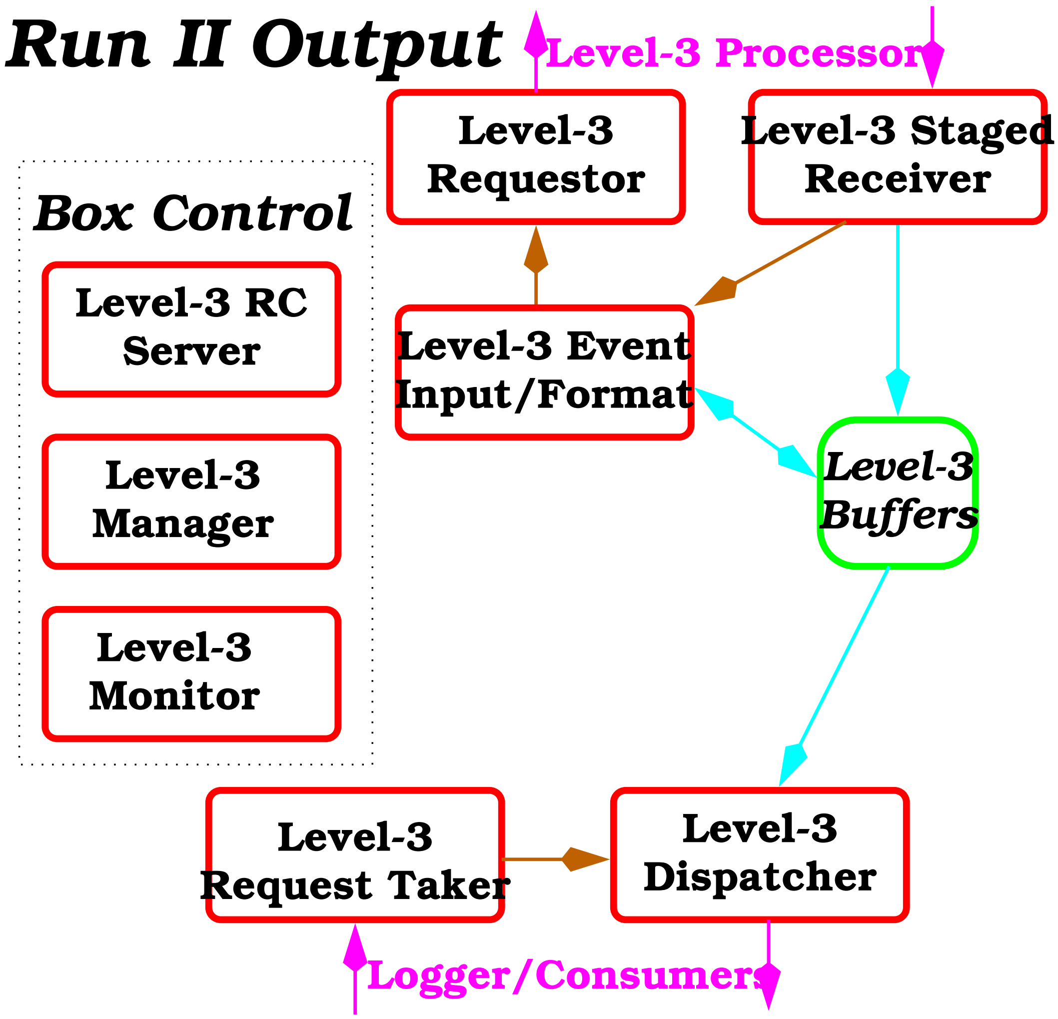

Figure 9: Run II Level-3 output node control code for the PC architecture

uses a different type of receiver for the processor nodes which receive events into PCs. This receiver builds events directly into the global buffers which are accessed by the CDF executable, and only requests a single event at a time, as buffers are available. Although this approach builds significant latency ($\sim$ 20 ms) into this part of the system, it is not excepted that this will be a limitation since events are expected to take $\gtrsim$ 1 CPU-sec, and since significant event buffering can easily be achieved in the memory of the PCs.

For this prototype, the output node (Figure 9) shares the same receiver design as the processor nodes. Here the request-receiver latency is more of a concern, although for the upper output rate limit of 75 Hz divided among 16 sub-farms, even the individual sub-farm peak rates (including upward fluctuations) should virtually never exceed 10 Hz over time-scales of more than a few seconds. If such a receiver design were shown to limit performance, a multi-threaded receiver, patterned on the one used at the ATM Switch input could be developed. At this time, the output network for logging and consumer processes has not been specified, and therefore no tests have been included in this prototype.

As noted above, the node most likely to be a bottleneck in the system is obviously the converter node, because of its high throughput. As shown in Figure 10, we have chosen to remove the old event formatter which copies events from input buffer to global buffers for analysis, and place its functionality in the processor nodes. Instead, the event remains in the input buffers until sent to a processor nodes, and the event fragments are merely concatenated at the time of sending, thus removing the need for one memory to memory copy. While this does limit the functionality of the converter box to intelligently distribute the events based on complicated algorithms which might require access to many parts of the event, it reduces the memory bandwidth required.

The integrated system is shown in Figure 11. The results shown in Section 5 are based on a partial implementation of the design for the converter node event concatenation described above which does not support dynamic changes in event fragment sizes.

3.3.1 Level-3 Receiver Code

The Level-3 `Event_Receiver` is treated like a utility package by the Level-3 `Event_Input` process; it contains all the knowledge required to manage the event builder system, but hides it inside a simple interface to Level-3. All fragments of an event are collected into a single memory, and the list of fragment address is then passed to the Level-3 `Event_Input` code. The package is multi-threaded and written in object-style C. More details regarding the interface to and internal structure of the `Event_Receiver` can be found elsewhere[6, 7].

The `Event_Receiver` code has been successfully ported to both Red Hat

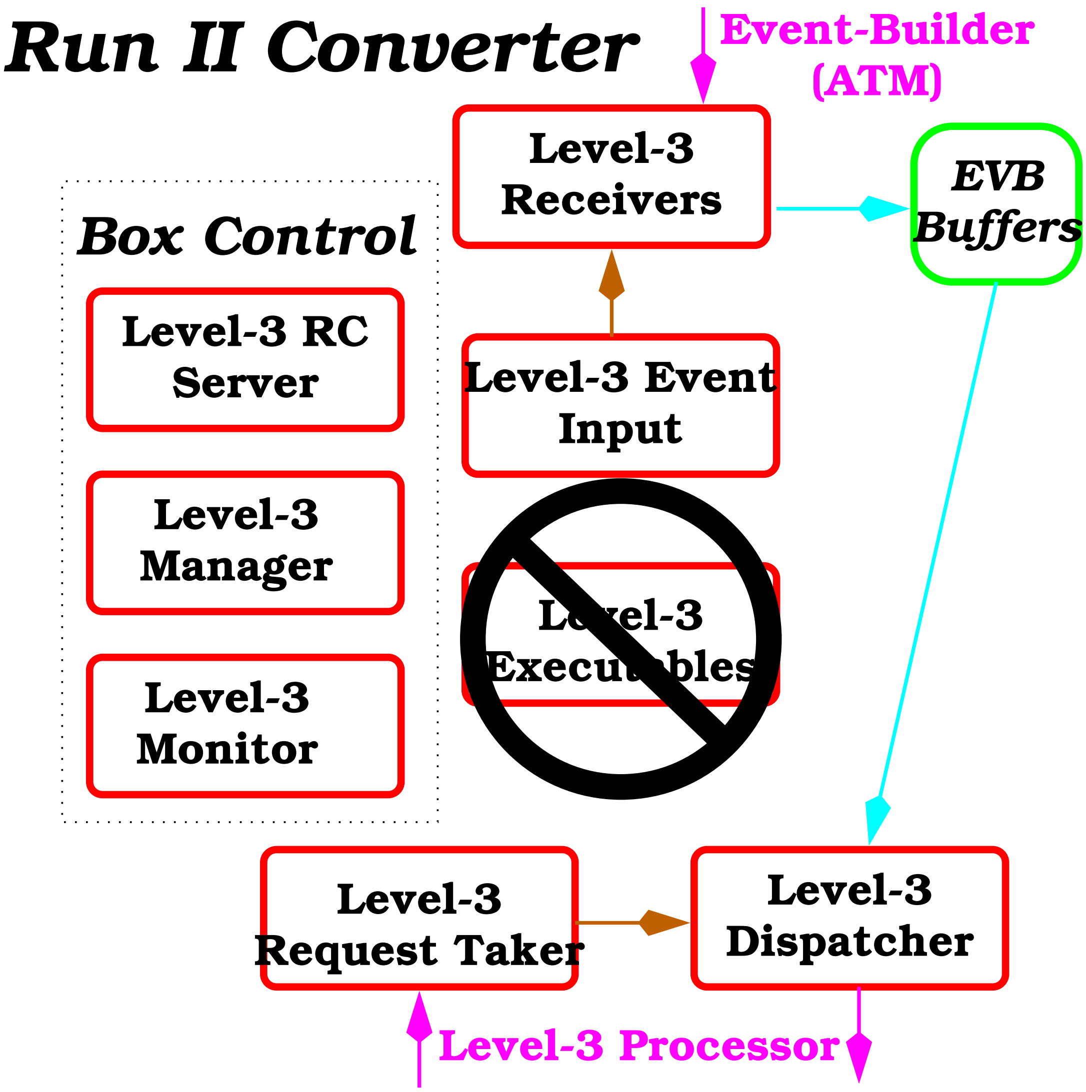

Figure 10: Run II Level-3 converter node control code for the PC architecture. Note the significant changes from the Run Ib code (Figure 7) with the lack of an event reformatter and filtering executables.

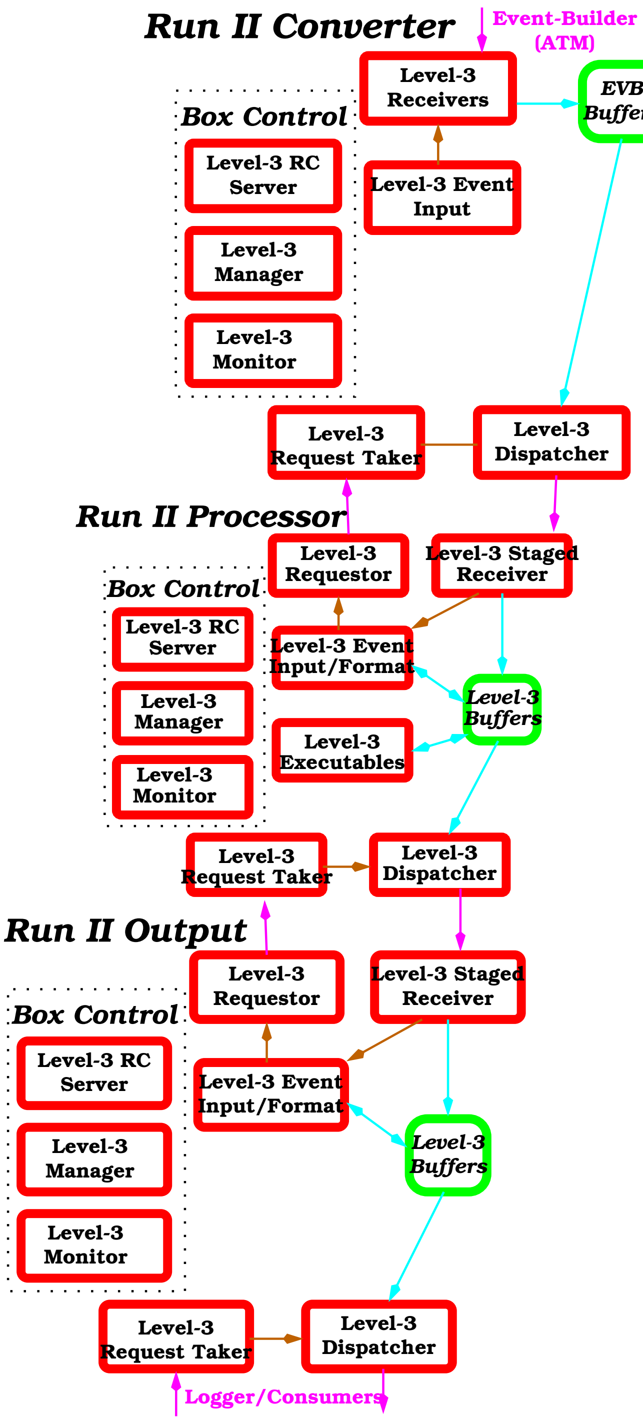

Figure 11: The integrated Run II Level-3 control code for the PC architecture

4.2 and Red Hat 5.0 distributions of Linux, and successfully manages the interface to the Event Builder in our tests.

3.4 Future Development

Significant work on the control code is still required in order to implement the complete functionality needed by CDF. Conceptual designs for distribution of calibration and monitoring data exist[8], but these plans have not yet been implemented. Efforts also continue to remove extra memory-to-memory copies which were convenient in the Run Ib framework, but would adversely affect the performance of a Run II PC Farm.

4 SC97 Demonstration Farm

A vertical slice test of a Pentium-based PC Level-3 has been performed by a collaboration of physicists and computing professionals from MIT, Fermilab and Academia Sinica[1] A schematic of the test system is shown in Figure 12. A scaled down version of the event builder feeds the Level-3 sub-farm through an ATM-based switch. The Level-3 sub-farm consists of dual-P6/200MHz PCs operating under Linux. The sub-farm runs control software based on CDF Run Ib code with modifications to accommodate the different hardware. The current control software incorporates interfaces to the CDF Run Control and Consumer Server systems as they existed in Run Ib. Events are passed successfully through the sub-farm which includes a null analysis process running on the processor nodes. Data throughput rates for this test sub-farm exceed 8 MB/sec which is approximately a factor of 2 more than is required to meet the specifications of Table 1.

5 Results since SC97

A number of different test configurations have been evaluated over the past months. Because of the apparent memory bandwidth limit in the converter node discussed below, unless otherwise noted these results were obtained

[1] The details of this vertical slice test will be documented in a subsequent CDF note. A prototype system[9] was shown at the recent SC'97 conference in San Jose.

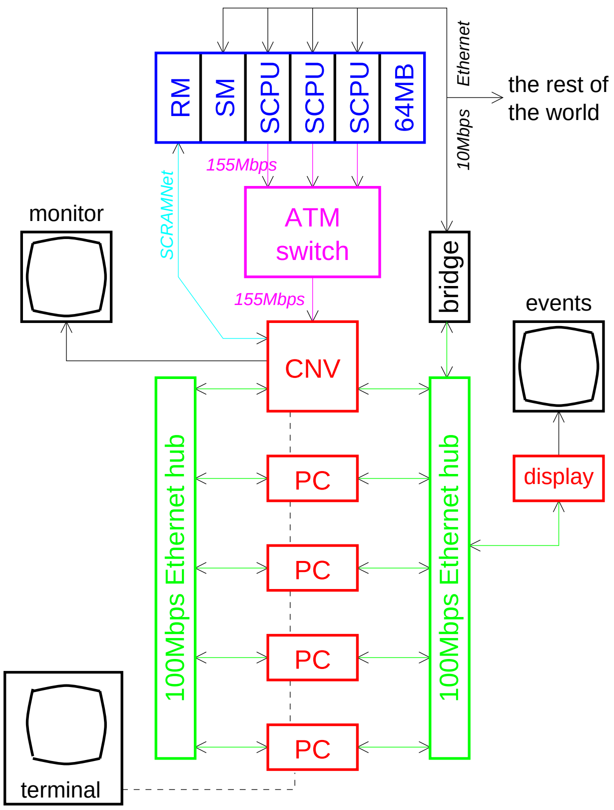

Figure 12: The test Level-3 sub-farm used to benchmark rates.

using the Level-3 Pentium II test node with SDRAM in order to maximize this bandwidth.

5.1 Tests of Simplified Configurations

The initial Level-3 hardware test stand consisted of four dual 200 MHz Pentium Pro processor nodes, one single 200 MHz Pentium Pro converter node, and two fast ethernet repeating hubs. The converter node housed a FORE ForeRunnerLE ATM interface and a SCRAMnet network interface. The converter and processor nodes each contained 32 MB of FPM (fast page mode) memory per processor.

Prior to the porting and conversion of the Run 1B Level-3 code to Linux and to the distributed architecture discussed in this memo, two simplified

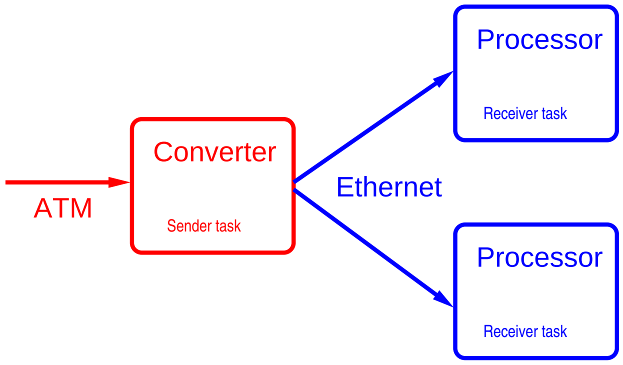

Figure 13: The smaller simulator. Fixed length event fragments are read from ATM by the converter node, then sent via ethernet to alternating processor nodes.

simulators were developed. Each simulator tested total throughput from the Level-2 ATM interfaces, through the ATM network switch, into the converter nodes, and out to one or more processor nodes via the fast ethernet networks. The smaller simulator, shown in Figure 13, consisted of a sender task running on the converter node and a receiver task running on each of two processor nodes; the converter node task read fixed length AAL5 packets from an ATM socket into a buffer, then immediately sent the data within the buffer via alternating ethernet sockets to the processor nodes.

The larger simulator, show in Figure 14, consisted of two tasks running on the converter node and one task running on each of two processor nodes. The tasks on the converter node attached to a shared memory area and a message queue. The ATM task read event fragments into fragment buffers in shared memory and sent a message to the ethernet task when a full event was available. The ethernet task accepted the message, then used a "scatter-gather" routine to send the full set of fragments to one of two ethernet sockets connected to processor nodes. For most testing the converter node was configured to simply alternate destination nodes.

Unlike the full Level-3 software, both simulators ran in open loop mode. ATM data originated from a free-running Level-2 computer, and both rate division and inter-message delays were used to adjust the average rate of sending.

The smaller simulator provided a useful framework to assess the capabil-

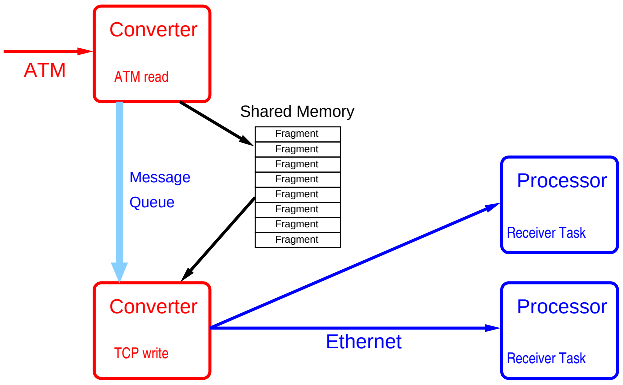

Figure 14: The larger simulator. Fixed length event fragments are read from ATM by the converter node and stored in shared memory buffers. Entire events are gathered from these buffers and sent via ethernet to alternating processor nodes.

ity of the converter node ATM interface to handle full "wire speed" ATM streams. Testing with this simulator revealed that at rates in excess of 12 MBps, ATM packets were dropped within the Linux kernel ATM software because of socket buffer overflows. The user-space process which read ATM data and re-transmitted the data via ethernet was not scheduled to run frequently enough, and at startup the transmission rate of the ethernet layer was throttled well below 12 MBps. Several modifications to the kernel and the simulator code were necessary to stop dropped packets: requesting the rescheduling of user process after each AAL5 packet reception, increasing the size of ATM and ethernet socket buffers, disabling the Jacobson slow-start algorithm in the TCP/IP protocol stack software, and using real time scheduling queues. These changes resulted in sustained data throughput at maximum ATM rates (approximately 16.5 MBps) with no dropped packets.

The larger simulator provided a data movement pattern very similar to the full Level-3 software. In contrast to the smaller simulator, event fragments were buffered in shared memory until entire events could be sent to the processor nodes. Consequently, instead of being available from the microprocessor's L2 cache at the time of output via ethernet, event fragments were read from system memory. This effective extra "memory copy" exceeded the memory bandwidth available on the converter node and limited throughput to approximately 12 MBps.

A number of memory copies occur when moving data through the converter node. An ATM AAL5 packet received by an interface is copied by the kernel to a kernel socket buffer. The data are copied by the simulator code from the socket buffer into a shared memory buffer. On output, data are read from the shared memory buffer and written into an ethernet socket buffer. For transmission the data in the socket buffer are copied to fragment buffers, the contents of which are written to the ethernet interface. In total, seven memory reads and writes are necessary to move a word of data from ATM interface to ethernet interface.

The fast page mode memory originally in the converter node has a bandwidth of 88 MBps (as measured by the "STREAM" benchmark [10]). Because of the seven memory copies, a throughput of 12 MBps requires 84 MBps of memory bandwidth.[2] To increase available memory bandwidth, the converter node memory was replaced with EDO ("extended data out") memory.[3] This resulted in a measured memory bandwidth of 108 MBps. Further, the simulator sustained throughput increased to over 15 MBps. Because the full Level-3 framework by necessity consumes additional memory bandwidth, the converter node was replaced with a dual-Pentium II computer equipped with SDRAM ("synchronous dynamic random access memory"). The measured memory bandwidth of the new converter node is 170 MBps; the sustained throughput of the simulator on this node is over 16 MBps, limited by the bandwidth of the ATM OC3 input.

5.2 Stand-alone CDF Level-3 Tests

After the feasibility of obtaining reasonable converter node throughput with "minimal" Level-3 framework was established, the next step was to test a

[2]This is actually a worst case memory bandwidth requirement. The L2 cache on the Pentium Pro microprocessor will reduce the memory bandwidth consumed when data accesses are close in time.

[3]FPM, EDO and SDRAM differ in the number of memory bus cycles used to access data. For four consecutive data accesses on a given memory page, the number of clock cycles needed with FPM DRAM are 5/3/3/3, whereas EDO DRAM uses 5/2/2/2 cycles and SDRAM uses 5/1/1/1 clock cycles. Thus the maximum effective burst data read rate on a 66 MHz bus is 19 MHz for FPM RAM, 24 MHz for EDO RAM and 33 MHz for SDRAM.

complete Level-3 sub-farm running the CDF-specific control code described in Section 3.3.

Initial tests involved the repeated reading of small samples of events from local disk on the converter node rather than using external (event builder) input as the source of the events. Since the sample files of raw CDF Run I data used were relatively small (approximately 1 MB), these files were effectively resident in local memory (disk cache) during the tests; therefore, disk I/O speed was not a factor in these tests[4].

Figure 15 shows rate results for distribution of 250 kB events from a converter node to between one and four processor nodes where the events are then all rejected. Four configurations are displayed for each set of results: (1) a converter with FP RAM, (2) a converter with EDO RAM, (3) a converter with SDRAM and a single ethernet, (4) a converter with SDRAM sending over two ethernets. The rate with a single ethernet can, of course not proceed at rates over the 100 Mbps limit of fast ethernet, less TCP overhead and losses from packet collisions. The latency between completing reception of an event into the processor, issuing a request and beginning to send the event into a single processor is 25 ms; therefore for 250 kB events, the rate into one processor is limited to approximately 6 MB/s. As illustrated in Figure 15, the crucial converter-to-processor data transfer, where the highest troughput rates will be needed, can handle rates in excess of the maximum plausible readout rate for Run II, 1 kHZ of 250 kB events.

5.2.1 Adding Output Nodes to Level-3

Tests involving output nodes are currently limited in scope by both incomplete development of the control software and by the small number of processor nodes in the system.

The former deficiency in the software exists because reception of events from $N > 1$ sources in the current staged Level-3 design is not supported; therefore, output nodes can receive from only a single processor node. The latter limitation is clear from Figure 15; the system cannot reach full throughput with only 3 processor nodes due to request-receive latencies. This problem is exacerbated by the additional latency from the output stage.

[4] In a separate test, the disk cache copy to user memory speed was benchmarked at 45 MB/s with EDO RAM.

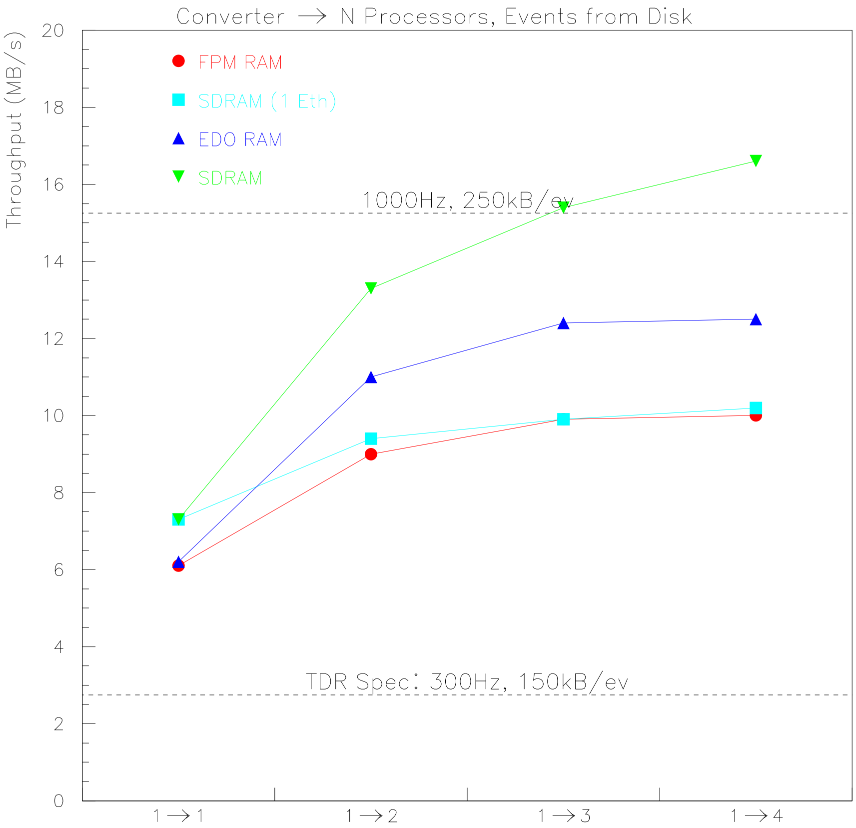

Figure 15: Level-3 data throughput, events read from disk cache. Different network and converter node memory options are shown as a function of number of processor nodes.

The rate results for the stand-alone Level-3 system (events read from disk) are shown in Figure 16. In tests with an output node, the rate to the output node varies from 2–4 MB/s, but in all cases this rate exceeds the upper bound from the CDF-D0 Joint DNAD[2] for a sub-farm, which is 1.2 MB/s (75 Hz of 250 Kb events over 16 sub-farms). As is shown, the rates with and without output nodes are similar for three processor nodes, and these rates with an output can reasonably be expected to rise with an increase in the number of processor nodes.

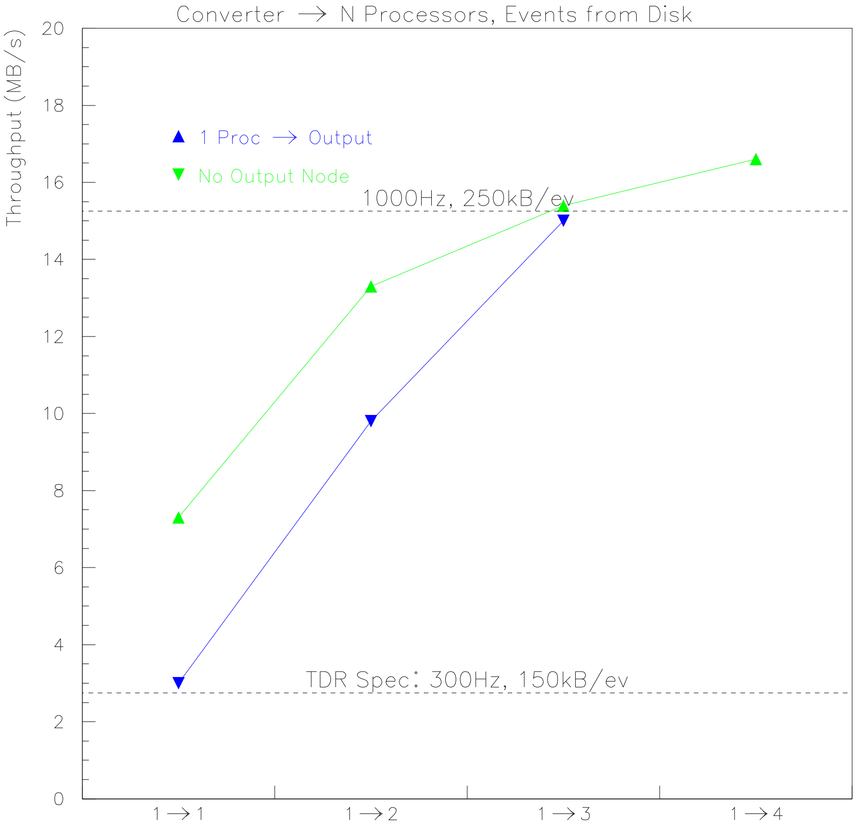

Figure 16: Level-3 data throughput, events read from disk cache. Results with and without transfer of events to an output node are shown. Rates to the output node vary from 2–4 MB/s, but in all cases exceed the maximum output rate for a single leg by at least a factor of 2.

Separate tests of a system with two-way traffic (converter to processor and processor to output) on a single fast ethernet physical LAN or VLAN were also attempted. Rates from these tests were significantly lower, indicating that packet collisions were adding significant overhead to the system. Therefore, our final implementation will be based on an ethernet switch or on three physical LANs (two for converter to processor traffic and one for processor to output traffic).

5.3 Integrated CDF Level-3-EVB Tests

The final category of tests used events from the ATM-based Event Builder currently being constructed for CDF Run II[7]. This test most closely models the expected behavior of the real Level-3 system, since events are coming from an external network into the converter node. Relative to reads from disk cache as in the stand-alone tests described above, the memory bandwidth demands are expected to be similar, but the PCI bus usage will clearly be different because of the two-way traffic on the PCI bus. It should also be noted that the raw ATM OC-3 bandwidth of 155Mbs, which translates to 16.1 MB/s theoretical bandwidth after subtracting AAL5 packet overheads, plus the additional small overheads in the ATM-based event builder will limit the throughput to under $\sim$ 15 MB/s (near the maximum plausible readout rate of 1 kHz of 250 kB events into 16 sub-farms, 15.2 MB/s).

Tests were attempted with a system up to the size shown in Figure 17, 14 Scanner CPUs building into a single converter node which fed 3 processors with events shipped to an output node.

The throughput results from the integrated Level-3-EVB tests are shown in Figure 18. Two sets are rates are shown: rates with the reformatting step done in the converter node and rates with this reformatting deferred until the processor nodes. As discussed in Section 3.3, we plan to take the latter option. In this case, rates for event distribution to processor nodes approach the ATM event-builder limit of 15.2 MB/s once the system has four processors receiving events. As in the Level-3 stand-alone tests, rates are similar with and without output nodes with the differences caused by the request-receive latency becoming less significant as the number of processor nodes increases.

6 Conclusions

We have discussed the need for increased processing capacity in the Level-3 trigger of CDF II and have shown that Pentium-based PCs can provide this capacity. We have then proposed a PC farm architecture which meets the Level-3 criteria for event throughput and which is expandable as more processing power is needed for Level-3.

Tests with a small prototype of a Level-3 PC farm are encouraging. With

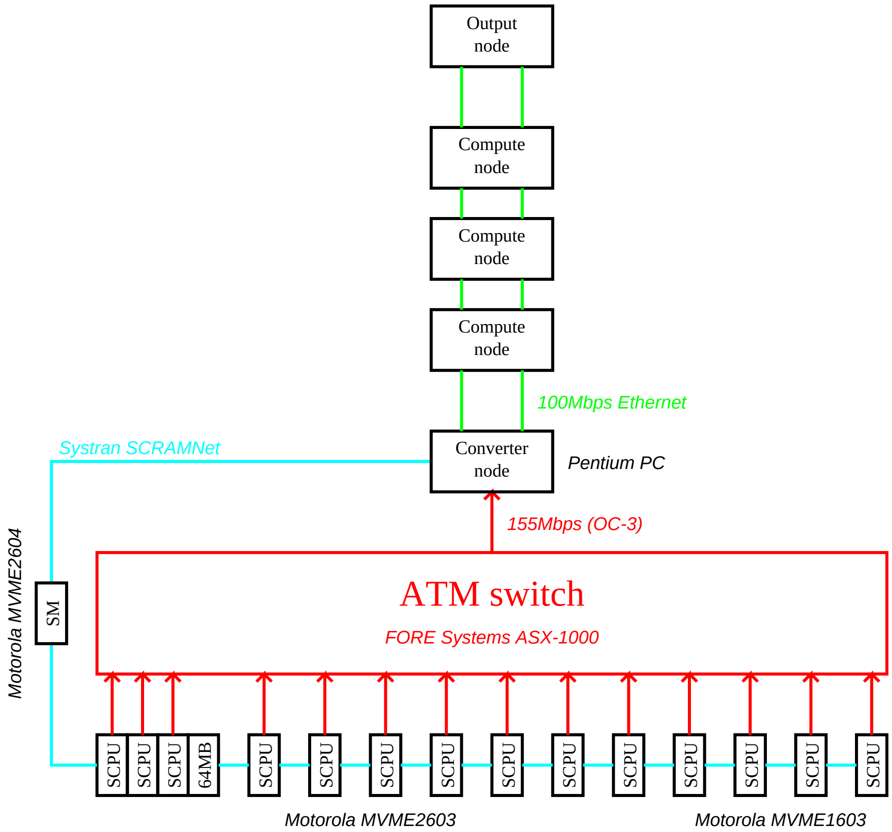

Figure 17: Hardware architecture for the largest integrated EVB-Level-3 tests attempted

a small, five node farm, we have demonstrated throughput to processor PCs and to other output of the Level-3 system which exceeds the CDF Run II specifications by over a factor of 4, and have assembled a control framework which can serve as the basis for the final Run II system.

In the near term, a second stage of testing is envisaged, with one sub-farm worth of processor nodes (approximately 20) and additional converter nodes. The primary aim of these tests will be to establish the throughput and reliability of large-scale PC farms and to prototype the links between Level-3 sub-farms and the consumer-server and data-logger.

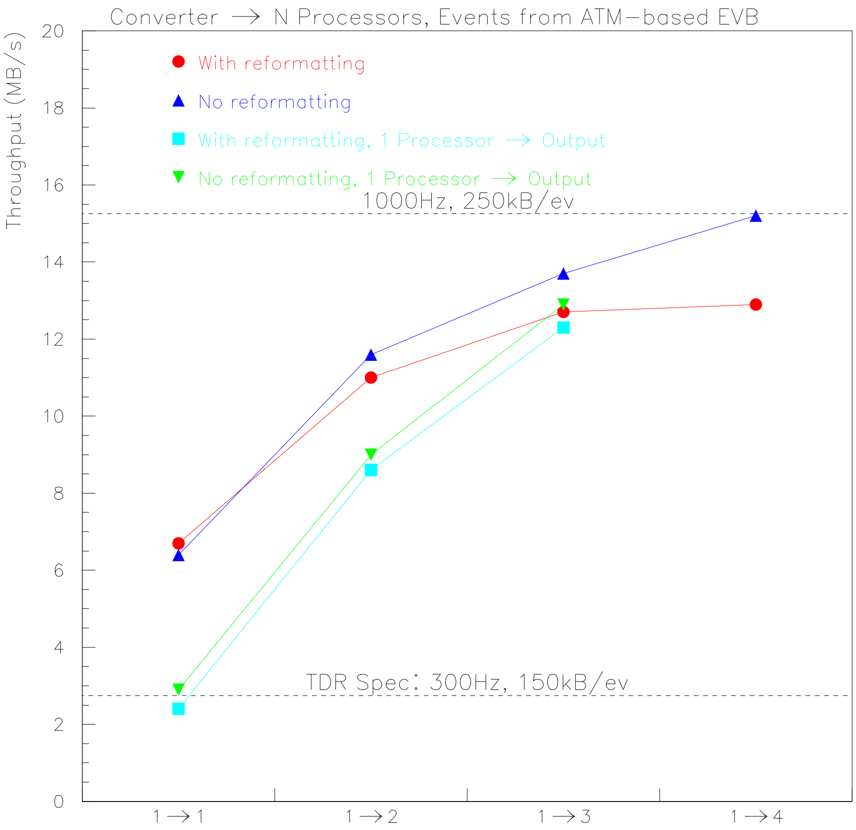

Figure 18: Level-3 data throughput, with events from an ATM-based Event Builder prototype. For these rate measurements, 50 kB fixed size packets from 3 Scanner CPUs. (Similar rates were observed sending 16 kB fragments from 14 Scanners.) For tests using an output node, rates to the output node vary from 2–4 MB/s, well above expected output loads for a single sub-farm. Two types of transfer through the converter were performed: with event "reformatting" (here benchmarked by a single memory-to-memory copy of the event among shared memory segments) and without.

Acknowledgements

We gratefully acknowledge the financial support of the Fermilab Computing Division for purchase of the Level-3 prototype hardware. We have received valuable input regarding this work from Jim Patrick. We are grateful to FORE Systems for discussions about their ForeRunnerLE ATM adapter which led to a functioning driver.

References

[1] J. Patrick, S. Vejcik, DAQ Upgrade meetings.

[2] CDF/D0/CD Joint Run II Data Management Needs Assessment, CDF Note 4100.

[3] A. Beretvas *et al.*, "PC Farms at Fermilab", CDF Note 4111.

[4] M. K. McKusick, K. Bostic, M. J. Karels, J. S. Quarterman, "The Design and Implementation of the 4.4BSD Operating System", Addison-Wesley, 1996, pp. 472-476.

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

[6] S. Tether, "The Level-3 Event Receiver", CDF Note 1864.

[7] G. Bauer *et al.*, "Status of ATM-based Event Building at CDF", CDF Note 4101.

[8] G. Bauer *et al.*, "Analysis Framework for the Run II Level-3 Trigger", CDF Note 4344.

[9] `http://www-ols.fnal.gov/pcfarms/sc97final.html`

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