

A Final Review of the Performance of the CDF Run II Data Acquisition System

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2012 J. Phys.: Conf. Ser. 396 012004

(<http://iopscience.iop.org/1742-6596/396/1/012004>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 69.47.189.46

The article was downloaded on 17/12/2012 at 19:18

Please note that [terms and conditions apply](#).

A Final Review of the Performance of the CDF Run II Data Acquisition System

W Badgett, *for the CDF Collaboration*

Fermi National Accelerator Laboratory, P.O. Box 500, Batavia, IL 60510, U.S.A.

E-mail: badgett@fnal.gov

Abstract. The CDF Collider Detector at Fermilab ceased data collection on September 30, 2011 after over twenty-five years of operation. We review the performance of the CDF Run II data acquisition systems over the last ten of these years while recording nearly 10 inverse femtobarns of proton-antiproton collisions with a high degree of efficiency – exceeding 83%. Technology choices in the online control and configuration systems and front-end embedded processing have impacted the efficiency and quality of the data accumulated by CDF, and have had to perform over a large range of instantaneous luminosity values and trigger rates. We identify significant sources of problems and successes. In particular, we present our experience computing and acquiring data in a radiation environment, and attempt to correlate system technical faults with radiation dose rate and technology choices.

1. Introduction

The Tevatron proton-antiproton collider at Fermilab ceased operations on September 30, 2011 after more than twenty-five years of operation. For most of that time, the Tevatron ran as the highest energy particle collider in the world and was only recently superseded in energy reach by the LHC proton-proton collider at the CERN laboratory. The CDF (Collider Detector at Fermilab) experiment observed and recorded the particle collisions produced by the Tevatron over this lengthy span of time with a high degree of efficiency. Central to the CDF experiment were its trigger and data acquisition systems, which selected and recorded interesting collisions. In the year 2002, CDF began its Run II data taking period after an extensive mid-life overhaul of the detector and electronics. Now that the CDF data taking operations have completed, we take this opportunity to explore some aspects of the performance of the trigger and data acquisition systems over this period.

Many possible measurements of performance exist, and we select only a few here to discuss as amongst the most interesting. We briefly review the design of the trigger and data acquisition systems. Following that will be a numerical look at the performance, at detector records, and at the efficiency of CDF as a whole. In section four we look at event rates and our effort to maximize the physics output of the experiment by adjusting them in real time, a process named “dynamic prescaling”. During the ten years of running we made a number of upgrades to improve the performance, which we cover in section five. In section six we observe the more qualitative aspects of the various technology components of the software control systems used. We also digress briefly in section six to have a look back at the evolution of control computing in high energy physics experiments. Section seven looks in

details at the most pernicious causes of inefficiency in the experiment – “single event upsets” involving computing elements running in a radiation environment. Conclusions and a plea for common frameworks for physics experiments come last.

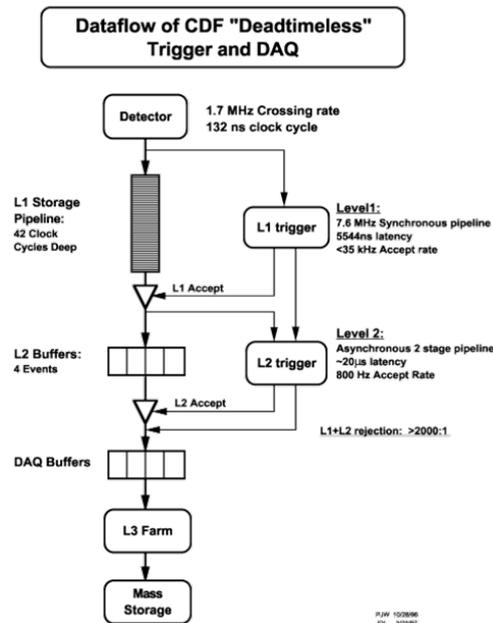


Figure 1: Diagram of Trigger and Data Acquisition Design

2. Trigger and Data Acquisition Design Overview

The trigger and data acquisition systems have been described many times in detail elsewhere [1], we briefly review the system here. Figure 1 indicates the flow of the detector data from the front end to the final mass storage. CDF had a three level trigger and data acquisition system, with each level successively reducing the raw crossing rate such that the final Level 3 output rate is at a manageable level for offline storage.

The Level 1 trigger system was synchronous to the beam crossings in the Tevatron collider. Level 1 made a “deadtimeless” decision in the span of 42 bunch crossings, all the while storing the detector data in a data acquisition pipeline while the decision was made. Level 1 operated at a beam crossing rate of 132 nanoseconds, although in the end the Tevatron only provided beam crossings at 396 nanosecond intervals. Level 1 consisted entirely of specialized custom hardware.

After the Level 1 decision is made and is positive, the detector data are moved to an intermediate Level 2 buffer on the custom readout boards, while a more detailed Level 2 decision is made. Level 2 was a hybrid of dedicated custom hardware and software decision making modules. Level 2 had access to more detailed data than Level 1, but still not the full detector data set.

With the Level 2 accept decision, all detector elements were read out and sent to an event builder and a Level 3 processing farm. Level 3 consisted entirely of commercial processors running offline style event reconstruction with full access to all of the detector data record. After a Level 3 accept, the event record is cached on a local disk array, and then sent to an offline mass tape storage system where more leisurely reconstruction and analysis was made.

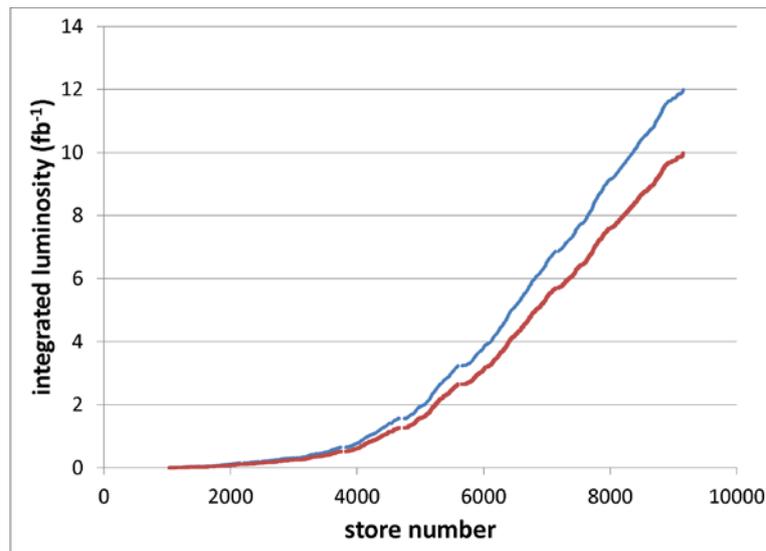


Figure 2: Integrated Luminosity, 2002 - 2012

3. Records, Performance, Data Taken

The best measure of the success of a collider experiment is the luminosity delivered and recorded over the course of the experiment. More luminosity means more collisions seen and recorded, which in turn means greater sensitivity to rare physics processes. In Figure 2 we plot the amount of luminosity delivered (blue) by the Tevatron and that recorded (red) to mass storage by CDF. The horizontal axis shows the store number of the Tevatron, where one store is one proton-antiproton beam injection session. The store number correlates highly with time, and the terminus of the plot data is September 30, 2011. The total delivered by the Tevatron was approximately 12 fb^{-1} and recorded by CDF was 10 fb^{-1} . The instantaneous luminosity peaked at about $433 \times 10^{30} \text{ cm}^{-2} \text{ sec}^{-1}$.

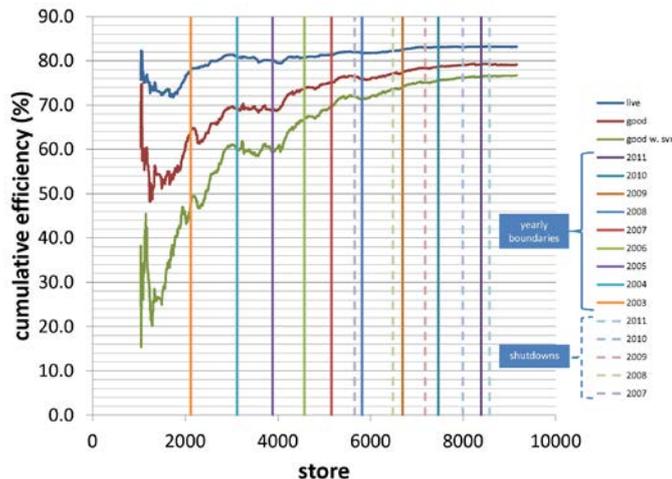


Figure 3: Cumulative Data Taking Efficiency

The ratio between the delivered and recorded data is the essential measure of the efficiency of the experiment's operation. Dividing the recorded by delivered curve from Figure 2 yields the upper blue curve in Figure 3 – the cumulative data efficiency for the full span of data. As expected, operational efficiency dips early in the run as kinks are worked out, and then asymptotically levels off towards the end of the run. Since the luminosity profile of the Tevatron gave low luminosities initially, and high ones in later years, the early inefficiency washes out of the total ratio.

The averaged efficiency in the end is about 83%. The primary causes of the lack of improvement during the later years were irreducible systemic problems. These problems were primarily single event upsets due to operating parts of the computing and electronics inside the collision hall of CDF, and thus inside a radiation environment.

3.1 Event Collection Numbers

The number of collisions processed at each level is shown in Table 1 for each of the system levels. An astounding 258 trillion beam crossings were processed by the Level 1 trigger at the input rate of 1.7 MHz. The first level trigger managed to reduce the rate by culling uninteresting events by a factor of about 100, yielding an approximate input to level 2 of 16 kHz. Level 2 further reduced the background rates by another factor of 50, leaving the Level 3 processing farm's job to reduce by a factor of merely 4. The low input rate to the Level 3 farm allowed a leisurely processing time in which all detector and physics objects could be reconstructed. The numbers indicated in Table 1 are the averages for the entire run, while the instantaneous rates varied a great deal.

Table 1: Events Processed at Each Level

Level	Events	Rate	Rejection Factor
Beam Crossings	258,224,721,879,337	1.7 MHz	
Level 1 Accepts	2,382,669,811,547	15.7 kHz	108.4
Level 2 Accepts	50,250,240,098	330.8 Hz	47.4
Level 3 Accepts	12,086,169,337	79.6 Hz	4.2
“to tape”	~150kB per event		

3.2 Throughput Limitations

Due to finite data processing times, each processing level possesses an inherent maximum rate that it can deliver to the subsequent level. Table 2 lists the averages and limitations approached during the stable period are the end of the run. These maxima represent another important metric of the performance of the system, essentially meeting or exceeding the design specifications.

Although the Level 1 trigger decision does not nominally incur deadtime, its output rate must be limited such that Level 2 has adequate time to make its own decision before the Level 1 accepts start to pile up. Four Level 2 buffers were available while the Level 2 trigger processors made a decision. If the buffers filled, further Level 1 accepts were not allowed to trigger, thus causing deadtime due to this Level 2 decision time. The Level 2 processing time typically ranged around an optimal time of 40 microseconds per event, in parallel, with a long positive tail. Significant contributions to this time came from the Silicon Tracker displaced vertex finder and the Calorimeter Shower Maximum detector. Both subsystems provided new information at Level 2 not previously available at Level 1. Another critical limitation on the Level 1 rate was that the deadtime fraction must be strictly limited to below the few percent level so that resonances on the Silicon Tracker wire bonds would not form. The Silicon Tracker readout line wire bonds were located within a magnetic field; when readout signals came at regular intervals of a specific frequency on the bonds, physical vibrations would occur, causing the bonds to break [2]. Resonant frequencies varied greatly depending on the location of the bonds, and ranged from 14 kHz to 112 kHz. The lower end of that range corresponded to the maximum Level 1 accept rate, when operating at saturated conditions.

The Level 2 output rate was limited by the hit processing and readout times on the central tracking wire chamber. The hit occupancy of this detector in Run II greatly exceeded that predicted by an extrapolation of occupancies from the Run I period. The scaling of occupancies with increasing luminosity exceeded the quadratic function expected. The time to digitize the chamber wire hits and

read them out over the VME bus determined the maximum rate obtainable of about 1 kHz at the highest luminosities experienced in Run II. If high luminosities had been maintained or exceeded, a significant reworking of the data acquisition would have been necessary. A secondary limitation was the VME bus readout speed. Many of the front-end VME readout crates had eighteen cards to read out in sequence over the bus, before the data were shipped via high speed serial link to the event builder buffers. Crates generally had hard limits of 200 to 400 micro-seconds readout time.

The limitation of the Level 3 was not intrinsic to itself, but rather to the ability of the offline production systems to process, store and analyse the data in the fully asynchronous offline world. Without the offline resource limitation, the Level 3 trigger could have simply passed the data through without additional rejection. However, additional Level 3 physics cuts significantly improved the purity of the data records in permanent storage.

Table 2: Throughput Maxima for Each Level, Latter Part of Run

Level	Average Rate	Maximum Rate	Limitations
1	17.6 kHz	40 -50 kHz	L2 Processing time; vertex finding; shower max data transfer; danger of wire bond damaga
2	439 Hz	950 Hz	TDC Hit Readout of central tracker; VME bus data transfer
3	105 Hz	400 Hz+	Offline computing resource

3.3 Performance Breakdown

To optimize operational efficiency, one must understand the sources of inefficiency. The global inefficiency can be broken down into two parts, as indicated in Table 3. From the total delivered luminosity of 12 pb^{-1} the losses of approximately 2 pb^{-1} are split into intrinsic deadtime and system faults and conditions. Intrinsic deadtime follows from the finite capabilities of the trigger and data acquisition to cope with given rates, if those rates cause back pressure in the data flow. The second category includes system malfunctions, high voltage trips and conditions such as high beam losses which prevented the system from operating normally.

The intrinsic deadtime depends not only on the capability of the system to handle specific rates, but also on the choices of physics content of the data stream. Trigger rates may be reduced by increasing physics thresholds, for example, but at the cost of losing physics content. A balance must be reached which maximizes the desired physics output without incurring too much deadtime. The following sections discuss this further.

Table 3: Luminosity Delivered and Recorded with Main Categories of Loss

Category	Hours	Luminosity, pb^{-1}	Luminosity, %
Total Delivered	42,255	11,994.6	100%
Total Recorded	36,490	9,997.4	83.2
Intrinsic Deadtime	1,344	782.0	6.5%
System faults & conditions	4,420	1,179.4	9.8%

The system faults and conditions category, on the other hand, must be minimized at any cost. As the easiest problems are solved early in the run, we are left with rare problems that are difficult to diagnose and solve. The top few categories of problems are shown in Table 4. The “events” column counts the number of times the problem cropped up. The “HV Problems” category groups together the top three subdetectors’ high voltage systems, all of which were located in the collision hall.

However, high voltage problems occurring at the beginning of a store are included in the “Startup” category. Since the category assignments involve human judgement, there may be overlap. The main subdetectors causing high voltage issues were the plug calorimeter, central drift chamber and silicon vertex detector. All three subdetectors had similar failure rates and used similar CAEN power supply and control equipment [3]. The second category of “Startup” represents the time setting up the detector for data taking after the Tevatron has declared the beams ready for physics. As noted above, in many cases, high voltage and readout problems are included within this category, too. The downtime induced by the next two most frequent categories, “L2 trigger” and “Event Builder”, formed part of the motivation for upgrades to these systems, as mentioned later in this article.

The calorimeter shower-maximum detector and silicon tracker readout problems share important issues with the high voltage systems: They are complex systems located in the collision hall, operating in a radiation field. Taken together, the collision hall based readout and high voltage systems cause the leading amount of system downtime. We believe the underlying cause are the so-called “single event upsets” in which a high energy particle traverses the electronics in question and induces a malfunction. Such malfunctions often require system resets and subsequent reconfigurations, during which the detector is not taking data. To test this hypothesis, we performed a study of the calorimeter shower-maximum readout, as described later in this article. To have alleviated the problems caused by single event upsets would have necessitated a major redesign effort.

The complexity of the experiment is shown by the number of categories of system malfunction, most of which do not cause major downtimes, but collectively count towards the 9.8% total downtime.

Table 4: Downtime Categories with most Luminosity Lost

Category	Events	Hours	Lost Lumi pb ⁻¹	Lumi, %
HV Problems	2970	598	200.3	1.667%
Startup	1899	265	197.1	1.643%
L2 Trigger	1115	253	51.6	0.430%
Event Builder	944	274	49.6	0.413%
Show Max R/O	534	139	48.4	0.404%
Silicon R/O	872	153	44.5	0.371%
Beam Losses	692	227	44.4	0.371%
18 More Categories with 0.1% – 0.3% luminosity losses				
24 More Categories < 0.1%				

4. Luminosity, Event Rates and Maximizing Physics Output

The Tevatron provided luminosity over a very wide dynamic range, as indicated in Figure 4, which plots the instantaneous luminosity as a function of real time for the highest luminosity store. The y-axis units are in $\times 10^{30} \text{ cm}^{-2} \text{ sec}^{-1}$ while the x-axis is local time. The lifetime of the intensity is very short at the beginning, while flattening out two or three hours later.

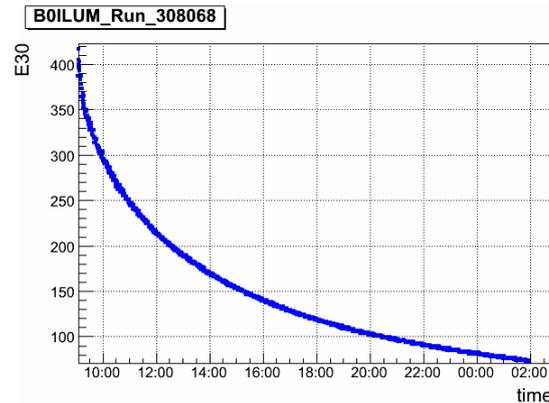


Figure 4: Instantaneous Luminosity vs. Time, Highest Luminosity Store

Rapidly falling luminosity represents a challenge to data collection. At the beginning of the store we have high luminosity and thus high trigger rates which saturate the throughput of the system. A short time later, the luminosity and the trigger rates fall, leaving the system with excess capacity. For the same high luminosity store, see Figure 5 to show the evolution of the intrinsic deadtime as a function of real time. The y-axis is percentage dead, where one hundred percent means the system is entirely saturated, and zero percent means the system is free enough to process every single bunch crossing that comes by. At the beginning of the store, the deadtime exceeds thirty-five percent, indicating the system was near the edge of capacity.

Even a small increase in luminosity would have caused the experiment to become inoperable. During the run, times with luminosity above $4 \times 10^{32} \text{ cm}^{-2} \text{ sec}^{-1}$ were rare; if they had not been, significant changes would have been required for continued operation. The important bottle-neck was the long TDC hit processing times; some inner layer(s) of the central tracking wire chamber may have had to be disabled, with their exponentially increasing occupancy.

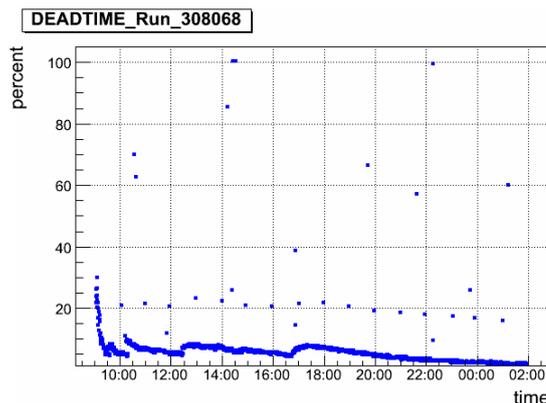


Figure 5: Deadtime vs. Time, Highest Luminosity Store

4.1 Maximizing Throughput and Physics

As luminosity decreases during a Tevatron store, excess bandwidth becomes free and can be used to increase the acceptance of high rate physics processes. Not only does the luminosity decrease rapidly, but the event size decreases as well, indicated in Figure 6. Detector hit occupancies decline as interaction probabilities decrease. High rate trigger paths that were impossible to enable at the highest luminosity become possible at lower points on the luminosity curve. These processes include, for

example, all-hadronic b quark decays indicated by a displaced vertex, and low transverse momentum muon trigger paths. The trigger paths are gradually enabled by decreasing the prescale value of the trigger path. The prescale mechanism vetoes trigger paths on a periodic basis in order to limit their rate; by decreasing the prescale value, the trigger rate increases. As triggers are enabled, the occupancy and data size of the events also changes. This process we call “dynamic prescaling” has extended the physics reach to include a plethora of bottom quark measurements, and even plays a role in the Higgs boson search.

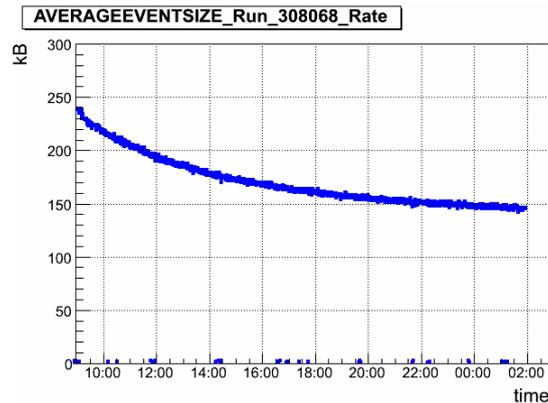


Figure 6: Event Size vs. Real Time, Highest Luminosity Store

Figure 7 shows the effect of dynamic prescaling on the Level 1 accept trigger rate, looking at the same record luminosity Tevatron store. Again, real time is on the x-axis; the Level 1 accept trigger rate is on the y-axis, measured in triggers per second. A variety of real-time mechanisms are implemented to control the trigger rate. One example is a “saw-tooth” mechanism near the time 16:00, where the trigger rate is gradually increased in steps as the actual rate declines over time. Checks were made before the change that predicted what the rate should be after that proposed change. If the predicted rate is within safety boundaries, the change is made. After the change, the resulting increased actual rate is checked for the same safety constraints; if safety margins are exceeded, the change is rolled back. In parallel, other applications checked the trigger rate and the deadtime to be within prescribed boundaries; if the boundaries were exceeded, the data taking would be paused in order that the malfunction could be repaired.

As luminosity further decreases, other algorithms kick in. Around 17:00 in Figure 7 the fractional prescale algorithm puts a high rate trigger on prescale values sliding from 2 to 1. The so-called “über-prescale” algorithm allows triggers to fire on coincidence with a free Level 2 buffer. The luminosity-enable algorithm allows triggers to be enabled at a prescribed value of the instantaneous luminosity. All of these algorithms are automated, and by about 18:00 in Figure 7, the remainder of all trigger algorithms are enabled to run out the remainder of the Tevatron store. As such, we maximize the utility of the experiment in terms of physics output.

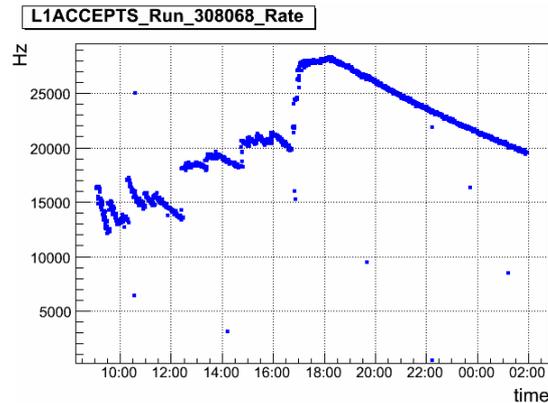


Figure 7: Level 1 Trigger Accept Rate vs. Time, Highest Luminosity Store

5. Trigger and Data Acquisition Upgrade

The design of the CDF Run II trigger and data acquisition straddles the eras of parallel bus readout and high speed serial links, and CDF Run II contained both types of systems. Over the years, increasingly faster and cheaper serial links have made parallel bus readout less attractive. The CDF system extensively uses serial links, but the first stage of readout after the Level 2 trigger accept begins with reading out front-end cards, one by one on each VME bus. The first central Level 2 processor also used a custom readout bus to transfer data from various subsystems into the central processor.

All the CDF upgrades implemented during Run II took advantage of the increasing speed, decreasing cost, and commoditization of serial links. The new links provided not only greater throughput but also increased reliability, especially when replacing parallel links. The original ATM and ScramNet [1] based event builder was replaced with a dedicated, private gigabit Ethernet switch. The first central Level 2 trigger processor replaced a custom parallel bus with multiple serial sLinks [4] into commercial computers. The Level 2 calorimeter trigger upgrade replaced many parallel data links with serial ones, giving increased reliability and flexibility in trigger algorithms. A subsection of the Silicon Vertex trigger used a similar strategy. At Level 2, and other locations, increasingly large FPGA chips provided another incremental upgrade path.

6. Data Acquisition Software Control Systems

As in most modern high energy experiments, the CDF control system was divided into two parts: the lower level software that talked directly to the electronics hardware, and the higher level software that talked to the user and distributed commands to the lower level. We describe briefly the technology and structure of both levels, and observe qualitative features of our experience with them.

6.1 High Level Control Software Experience

The top level control and monitoring software were primarily stand-alone Java applications running on Scientific Linux 4 on commercial PCs. This mode provided excellent system stability for the Run Control and various monitoring applications. The Linux computers were virtually impervious to crashing. Since communications to the rest of the system were through standard network interfaces, there was no special hardware nor device drivers in these machines. Such simplicity presumably provided the stability. Often the Linux PCs would run for eighteen months or more without a crash or shutdown. The shutdowns that did occur were for regularly scheduled Linux updates.

The Java language proved well suited to the applications. The ease of programming in Java greatly outweighed any technical advantages of languages such as C++, although the control applications are not as real-time critical as other applications. The object-oriented nature of Java suited the detector applications, since detectors have subdetectors, which have sub-subdetectors, all of which lends to object inheritance models in the software. Also one imagines a generic readout card, with child

instances of ADC or TDC cards, and further grandchildren of subdetector-specific cards. Drawbacks of Java included the mysterious and infamous crashes of the Java Virtual Machine (JVM), with cryptic dumps of the JVM essentials. The trace files were sent dutifully to the authors of the JVM, but with little success in interpreting them. In the end, mysterious Linux kernel panics had been replaced by mysterious JVM crashes. Mercifully, the Java control applications were fast and simple to restart.

Another strength of Java was the ease of creating and running multiple threads in a single virtual machine, allowing real-time applications of unlimited complexity and parallelism. Typically the central Run Control application would have approximately twelve concurrent permanent threads, with many ephemeral temporary threads started as needed. On the negative side, so many threads risked collisions and deadlocks, and may explain the rare hang ups of the user interface over the years.

Persistency was provided by an Oracle database running on a SunOS server. Originally starting with version 8, we ended up at last with 10, and even 11 after the fact. The database was used extensively for configuration and conditions. The reliability and extensibility of the database was well worth the additional expense, when compared to our early experience with freeware databases. Figure 8 plots the growth of the database over time, with the final size about 800 GB. The plots break down the sizes of the constituent database schemas, with some tiny like the Trigger configuration database, and others much larger, like the run and slow controls databases. Each application included Java and C++ object oriented wrapper classes for API access.

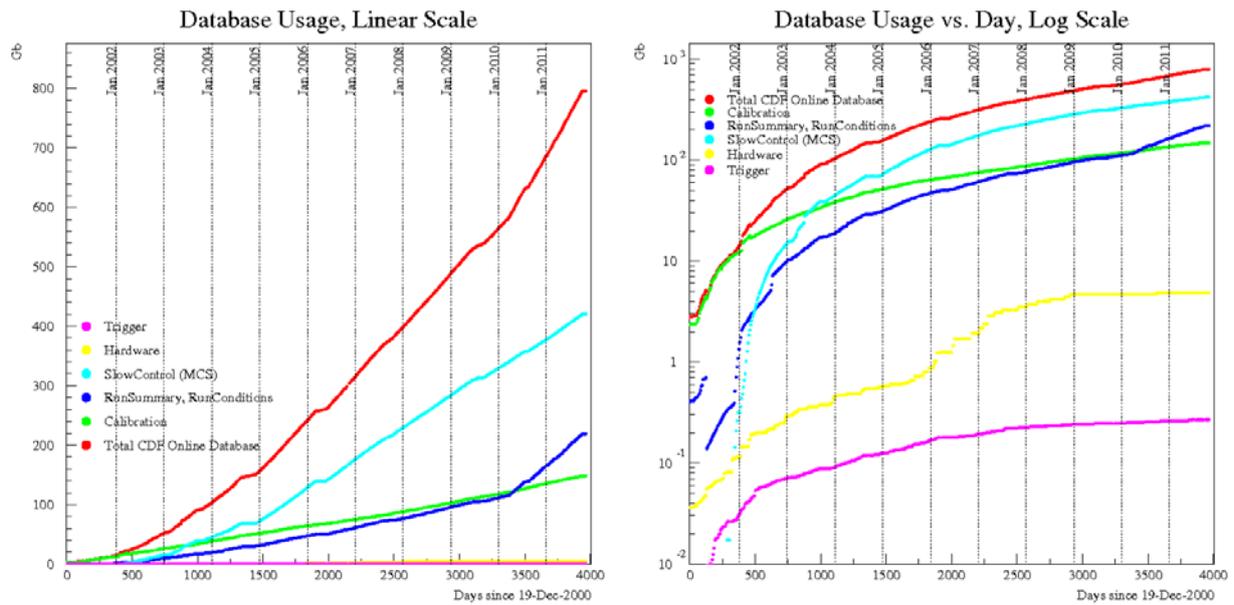


Figure 8: Database Size vs. Time

The Java applications all ran within a protected, private network. Extensive web-based monitoring tools were also developed, with the intention of providing a remote spy portal into the online workings, without the need for direct access to the private network. In recent years various high energy experiments have endeavoured to also put the run control and command applications on a web interface, with sophisticated use of javascript and dynamic HTML. Since these direct control applications will never be accessible outside the experiment's nominal private network, one wonders the need for the extra layer of complexity.

6.2 Low Level Hardware Control Software

The second distinct part of the online control software is the front-end software talking directly to the electronics hardware. For this purpose, we used Motorola MVME crate processors embedded in the VME readout crates, running *C* in the *VxWorks* operating system. For a majority of the crates, the crate processor conducted the readout of the data over the VME bus. For a typically heavily loaded crate, the readout time was on the order of 200-400 microseconds, which was a significant limitation on the readout latency, as noted above.

The experience with real-time operating system *VxWorks* was generally good. In spite of the somewhat eccentric design as compared to other fully fledged systems, *VxWorks* allowed one to know almost everything that was happening on the processor. Such knowledge can be critical when dealing with balky electronics devices. In this situation, Java would not be feasible, and *C* provides a necessarily low level contact with the hardware. However, as buses and VME go away, the use of embedded processors in this context may also wane.

6.3 RunControl Through the Ages

We digress briefly here to look back at high and low level control applications over the years, per the author's own experience. The bifurcation between high and low level software did not exist in the CDF Run I period. The Run I system used DEC VaxStations to run RunControl and was originally developed and used in the mid to late 1980's. The Vax was closely coupled to the hardware through a specialized device on its I/O bus, connecting to the Fastbus network. See Figure 9 for a look at Run I RunControl user interface, designed to run on a minimal VT100 text terminal.

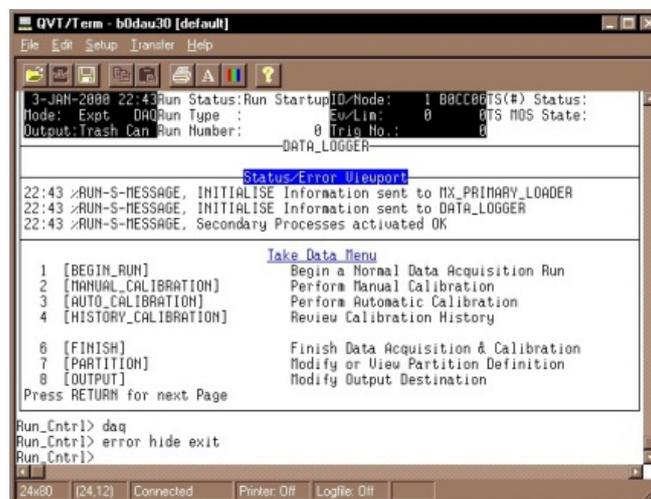


Figure 9: CDF Run I RunControl [5]

This RunControl had close contact with the front-end hardware, while also providing the user access to performing runs and basic monitoring, all within a 24 line by 80 character screen. When terminals with graphics capabilities appeared, a first attempt at a GUI can be seen in Figure 10. The left shows the Fastbus network of readout crates, while the right image shows a graphical representation of the detector elements, from which the user could select individual elements for inclusion in the readout. With the primitive state of pixel graphics at the time, the rendering of the right hand image could take up to ten minutes, and woe be to the graduate student who selected this graphical option during a physics run.

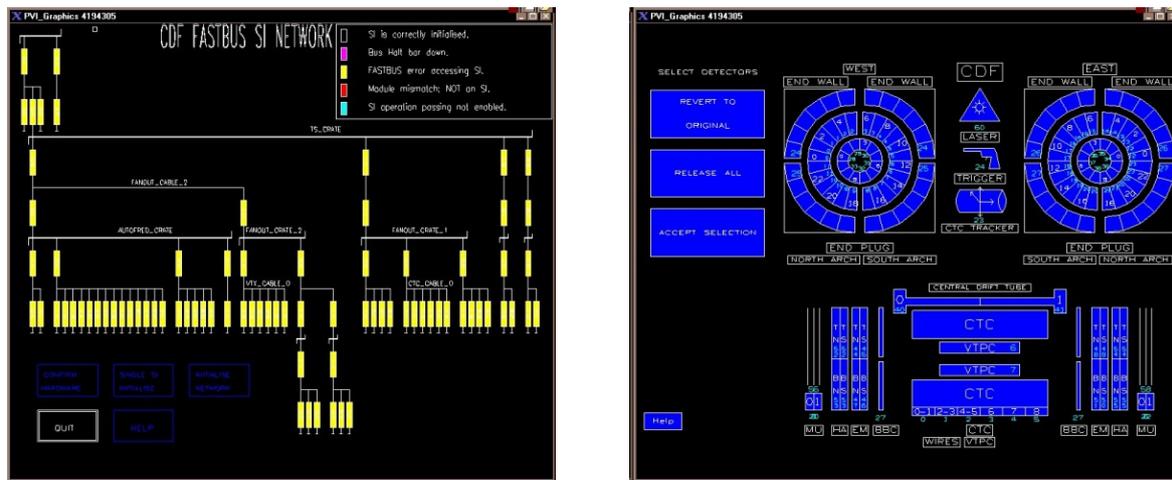


Figure 10: CDF Run I Fastbus Network Diagram and RunControl Detector Selection GUI

During the mid-1990's, the bifurcation between high level and low level was demonstrated in the Zeus experiment at DESY. The RunControl GUI [6], shown in Figure 11, was highly disconnected from the front-end hardware and the various subdetectors. Knowledge of the subdetectors was strictly limited. The subdetectors even had to write their own independent RunControl applications if they wished to have local data runs such as calibrations. The version shown is the second Zeus RunControl, written in TCL/TK. The front-end software was written in C and the Occam language. The front-end crates had Inmos Transputer processors, multiply connected to other crate processors by a network of serial links. Occam was designed to make parallel processing a built-in part of the language, with easy communications between local and remote threads through the serial links. As history repeats itself, the multiple serial connections resemble the increasingly popular ATCA crate architecture today.

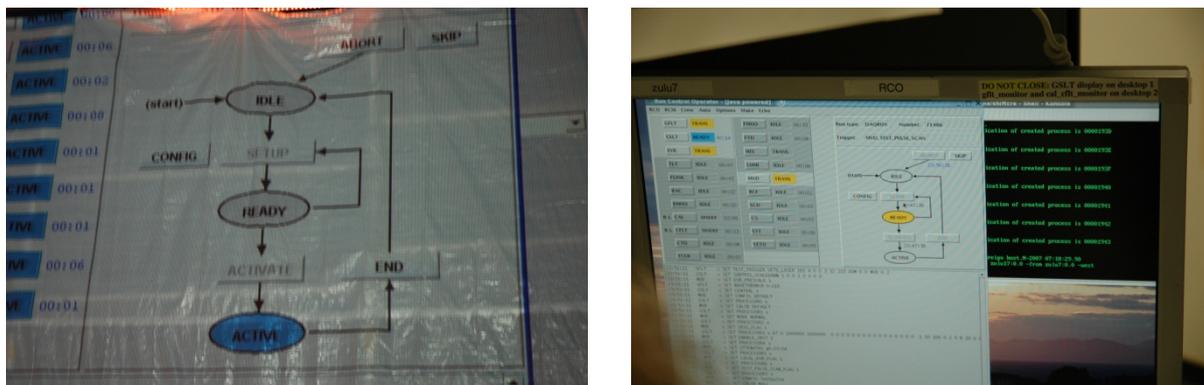


Figure 11: Zeus (second) RunControl Graphic User Interface

Next, in Figure 12 we show the CDF Run II RunControl GUI. The Zeus RunControl clearly inspired the graphical design of the state manager diagram, however the functionality went beyond the Zeus RunControl. The system also had the distinction between high and low level control, but by using Java in this case we have built-in extensibility. The standard physics state diagram on the left shows the flow of actions and states as the system builds towards a data taking run. In addition, subdetectors may create their own specific diagrams by extending the base class and adding or even subtracting features, as shown on the right for a calibration state diagram.

Figure 13 shows one example of a Java monitoring application: the high voltage inhibit control panel. This panel was one of several running in the CDF control room and monitoring the status of the detector. This status panel had a counterpart web page, as did all of the control room applications.

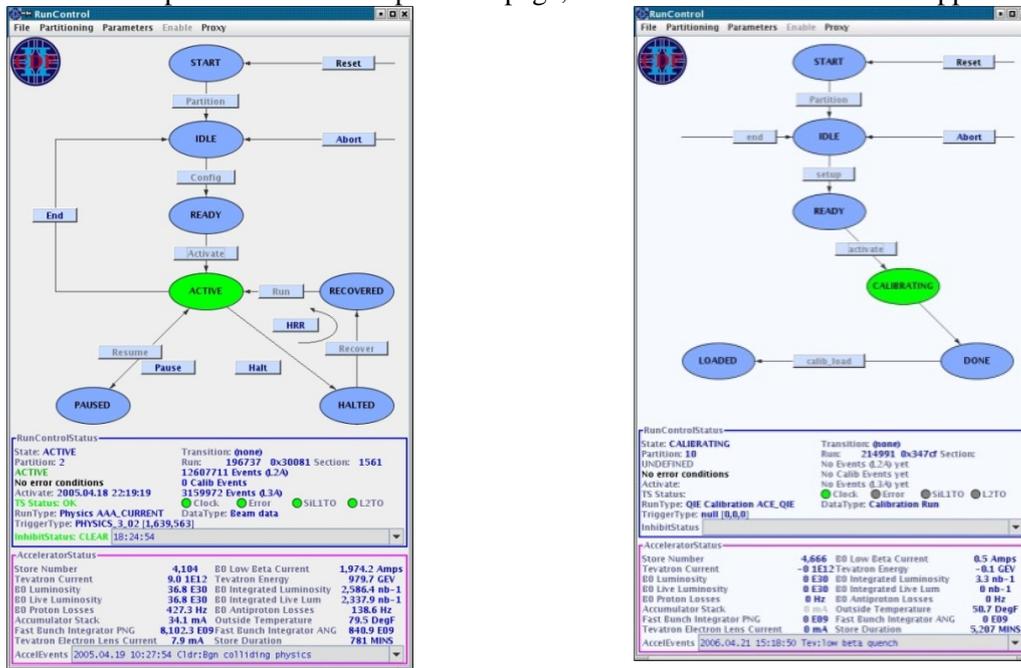


Figure 12: CDF Run II RunControl Graphical User Interfaces

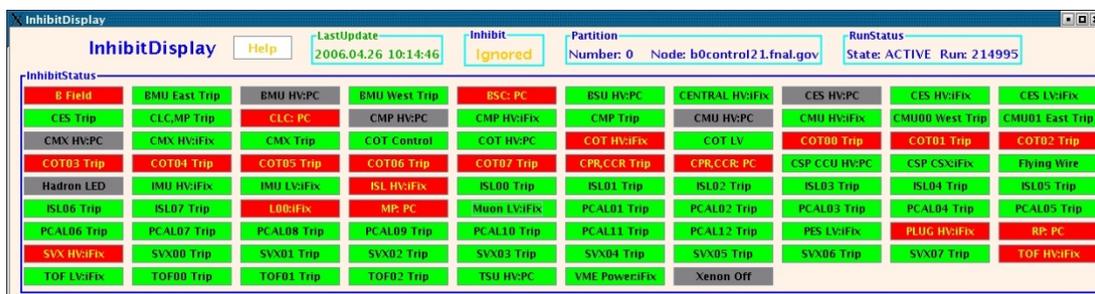


Figure 13: Detector High Voltage Inhibit Control Panel

Finally, in Figure 14 we show the RunControl of the CMS experiment at CERN. The CMS RunControl [7] divides into two parts: a Tomcat Java based back-end, and a Javascript and dynamic HTML front-end for the web-based user display shown in the figure. While many browsers could in principle run the display, in practice Firefox on Scientific Linux is used in the CMS control room. Subdetectors may inherit the classes in the back-end to extend to their own purposes. Javascript and dynamic HTML add another layer of complexity, giving a new risk of browser crashes to operations.

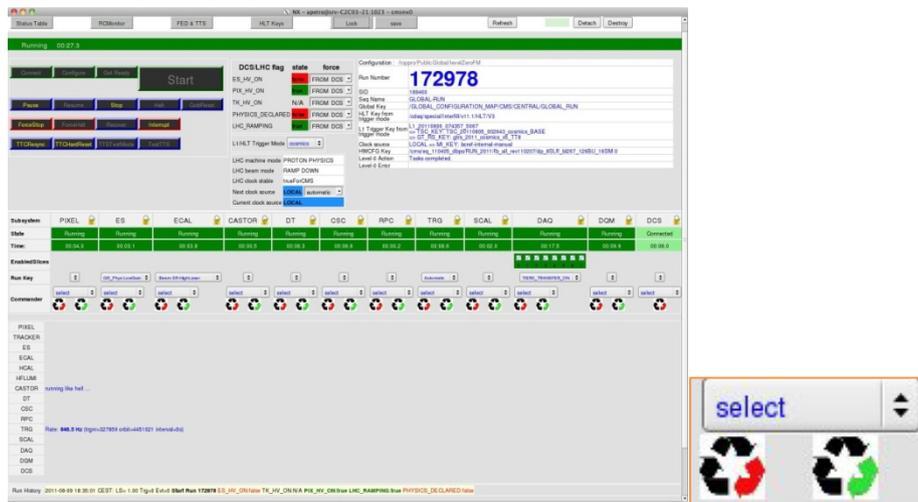


Figure 14: CMS Level 0 RunControl Web Based GUI

The CMS RunControl also divorces itself from direct hardware contact. The C++ language is used to access the electronics controller in a software package known as xDAQ [8]. In the case of CMS, there is no embedded processor in the VME crates, but rather a CAEN VME to PCI bridge [3] which connects to a Linux PC with a special PCI card. Due to increased throughput requirements at CMS, the VME bus is not used for data transfer, only for configuration. Slink [4] serial links are used to pump out the data directly from the front-end driver card to the event builder, demonstrating the evolution to all-serial link systems mentioned above.

7. Computing in a Radiation Environment

As mentioned above, the most difficult challenge in operating CDF smoothly was the presence of readout computers and high voltage control systems inside the collision hall. We must answer the questions, “Are the single event upsets related to the radiation field?” and “If yes, can we do anything to alleviate them?” The radiation field while running beams in the Tevatron is significant. The distribution of radiation is shown in Figure 15. The top plot indicates radiation dosage during normal colliding beam running of about 1 pb^{-1} of integrated luminosity. The bottom plot shows the radiation dosage during one occurrence of high proton beam losses. The views are from the top of the detector; the trackers are in the interior and the calorimeters farther from the colliding beam locus at the center of the plots.

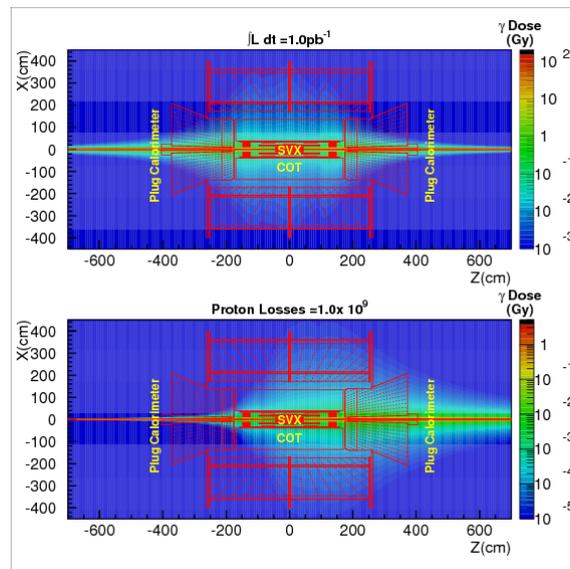
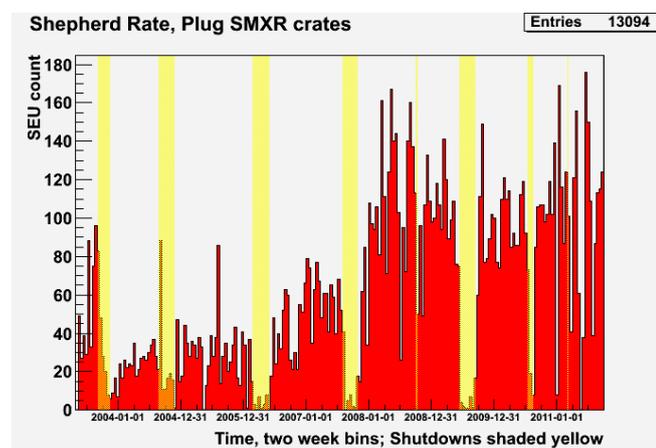


Figure 15: Radiation Dose around the CDF Detector [9]

For this study, we examine the rates of single event upsets in the plug calorimeter readout crates and show they are related to beam. These twelve crates are located just beyond the plug calorimeter shown in Figure 15 in groups of three in each of the four corners, in the blue region. The radiation sensors used to create the dose plot are not sensitive to the dose at this location, but there must be some low, non-zero rate throughout the area. To demonstrate this point, see the two plots in Figure 16, which together compare the single event upset rate in the Plug Calorimeter Shower Max crates vs. time, and the peak instantaneous luminosity vs. time. Roughly half way through the run, the Tevatron began to deliver consistently higher luminosity. Coincident with this time, the single event upset rate also increased significantly. Note also the yellow bands in the upper plots – these are Tevatron shutdown periods. During these times, the readout proceeded without significant upsets. Clearly the upset rates are colliding beam related. Through the dynamic prescale mechanism, data rates were kept approximately constant, so we expect such effects factored out to first order. Whether beam losses also cause upsets is more difficult to determine, since the detector readout was prohibited during high loss periods. The proton and anti-proton beam currents were highly asymmetric, with the protons several times more intense than the anti-protons. Therefore one might suspect an upset rate asymmetry on opposite sides of the detector; however, averaging the rates shows no asymmetry. Unlike beam losses, the radiation field from collisions, as seen in Figure 15, is forward-backward symmetric.



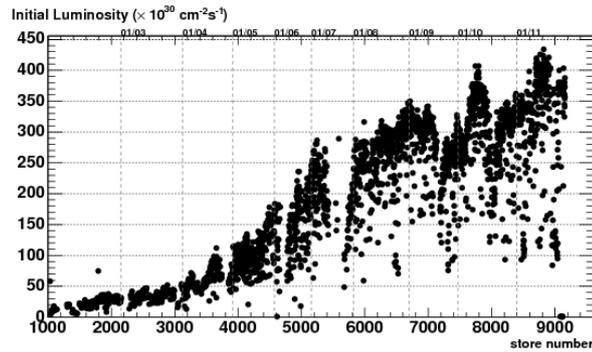


Figure 16: Single Event Upsets vs. Time and Peak Instantaneous Luminosity vs. Time

The upsets normally forced a reset and complete reconfiguration of the readout crate involved. This normally took one to two minutes, thus contributing significantly to downtime. This downtime leads to the second question, “Can we alleviate the problem?” The plug calorimeter readout crates come in two flavors: One has only plug calorimeter readout; the second has both plug calorimeter readout and plug shower-maximum detector readout. The shower-maximum readout is significantly more complicated than the calorimeter readout. The shower max requires various per channel calibration corrections and zero suppression, all involving significant computing and memory resources in the crate processor. The calorimeter data are passed downstream with no processing.

Figure 17 demonstrates the disparity in upset rates vs. type of crate. Crate numbers 0, 3, 6 and 9 are the calorimeter-only readout crates; the other crate numbers also have the complex shower-maximum readout. All the calorimeter-only crates have much smaller upset rates; however, they are not zero. One additional difference between the two crates is the type of embedded processor. The calorimeter-only crates use the Motorola MVME2300 while the others use the faster MVME2400. The difference could perhaps explain the upset rate differential. The MVME2300 may be less susceptible to upsets due to hardware design or materials used, or due to a slower clock speed. During the second period shown on Figure 17, we replaced one of the shower-maximum crate’s MVME2400 processor with an MVME2300 for a period of eight months (blue square data points). The rates were compared with the same eight month period in the prior year (red circle data points) with an equivalent amount of luminosity delivered. The green box show the rates for the experimental crate number 10. With the slower MVME2300 processor, we clearly see a significant reduction in upset rates. However, they do not go to zero nor to the low level of the calorimeter-only readout crates.

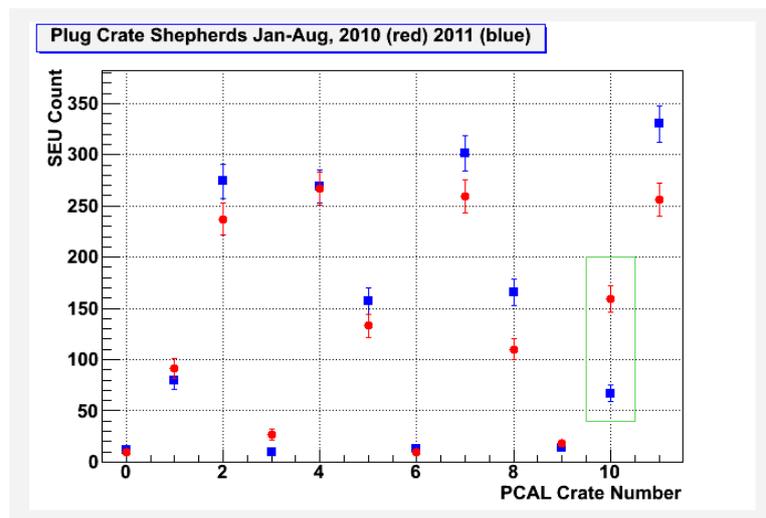


Figure 17: Single Event Upset Rate vs. Plug Calorimeter Readout Crate Number

The study of the crate processor indicates that it is possible to mitigate the effects of computing in a radiation environment. In this case the penalty was running with a slow crate processor, thus increasing the readout latency and reducing the maximum Level 2 trigger rate. The trade-off of reduced downtime was not worth the increased intrinsic deadtime. The rate was also not brought to zero. The extraordinary length of time – eight months – needed to show a significant change in the upset rate shows the extreme difficulty of improving the situation in situ. For a single readout crate, the rates are low, so the effect would unlikely to be seen at a test beam. But with one hundred crates and many high voltage controllers in the collision hall, the total effect bears down significantly on the efficiency of the experiment.

To conclude, we note that the simple readout crates without much computer processing had by far the lowest rate of upsets. Therefore the lesson learned is not to put complex computing inside a radiation environment.

8. Plea for Common Tools and Conclusions

The online software of high energy experiments that run large, complex detectors is itself large and complex. Across experiments, the functions and needs are all very similar. We would all benefit from common solutions. Such things as network message passing, error logging, event state machines and monitoring tools are all examples of common needs that should have a common solution. The high energy physics offline analysis world has examples of successful common tools, such as Root, Geant, Monte Carlos, etc. This indicates some form of collaboration should be possible, and the author invites readers to participate in such a collaboration. Some critical mass of people would be needed, but perhaps not too many to weigh progress down.

8.1 Conclusion

[The CDF data collection concluded on September 30, 2011 with twenty-five years of running and great success and efficiency during Run II. We look forward to the next generation of experiments and their upgrades. The trigger and data acquisition of the future may contain ATCA crates, Graphics Processing Units, $\times 10^n$ gigabit Ethernet or other new tools. The evolution of control software could bring surprises and new platforms. We hope there will be many more high energy experiments in the future to take advantage of these advances.

9. Acknowledgement

Work described in this document was performed at Fermi National Accelerator Laboratory, operated by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the United States Department of Energy.

References

- [1] Meyer A, *The CDF Data Acquisition System for Run II*, Proceedings of the Computing in High Energy Physics Conference 2001, Beijing, China, September 2001.
- [2] Bolla G, et al, *Wire-bond failures Induced by resonant vibrations in the CDF silicon detector*, Nuclear Instrumentation Methods A518 (1-2) 2004.
- [3] CAEN Electronic Instrumentation, <http://www.caen.it>
- [4] van der Bij H, *S-LINK, a Data Link Interface Specification for the LHC Era*, Proceedings of the Beaune 97 Xth IEEE Real Time Conference, Beaune, France, September 1997.
- [5] Patrick J, *private communication*, August 2011.
- [6] Youngman C, Milewski J, Boscherini D, *private communication*, August 2011.
- [7] Sakulin H, *Operational Experience with the CMS Data Acquisition System*, Proceedings of the Computing in High Energy Physics Conference 2012, New York City, New York, U.S.A., May 2012.
- [8] Brigljevic V, et al, *Using XDAQ in application scenarios of the CMS Experiment*, Proceedings of the Computing in High Energy Physics Conference, La Jolla, California, U.S.A., March 2003.
- [9] Kordas K, et al, *Measurement of the Radiation Field Surrounding the Collider Detector at Fermilab*, Proceedings of IEEE/NSS-MIC 2003 Conference, Portland, Oregon, U.S.A., October 2003.