Client and Event Driven Data Hub System at CDF

Ben Kilminster†, Kevin McFarland†, Tony Vaiciulis†
Hiroyuki Matsunaga*, Makoto Shimojima*

† *Department of Physics, University of Rochester, Rochester, NY 14627*
* *University of Tsukuba, Tsukuba, Ibaraki 305, Japan*

Abstract. The Consumer-Server Logger (CSL) system at the Collider Detector at Fermilab is a client and event driven data hub capable of receiving physics events from multiple connections, and logging them to multiple streams while distributing them to multiple online analysis programs (consumers). Its multiple-partitioned design allows data flowing through different paths of the detector sub-systems to be processed separately. The CSL system, using a set of internal memory buffers and message queues mapped to the location of events within its programs, and running on an SGI 2200 Server, is able to process at least the required 20 MB/s of constant event logging (75 Hz of 250 KB events) while also filtering up to 10 MB/s to consumers requesting specific types of events.

INTRODUCTION

CDF Run II Physics with the TeVatron

The Collider Detector at Fermilab (CDF) as well as Fermilab's proton/anti-proton particle acceleration facilites have undergone significant improvements in the pursuit of a better understanding of physics at the smallest scale. Due to the creation of the Main Injector and other improvements, the colliding proton/anti-proton beams in the Tevatron will deliver twenty times the luminosity of the previous data taking run. This means that there will be 7 million beam crossings per second in the CDF detector. CDF has risen to this challenge with faster electronics readout, and more extensive tracking, thus increasing the number of readout channels ten-fold. The combination of a better accelerator and detector improvements leads to a maximum 1 KHz digitization rate of 250 KB size events.

Data from the Detector

The data rate is reduced by a series of trigger systems, which use detector information to evaluate events. The third filter is a PC farm called Level 3 which uses event reconstruction software to lower the rate from 250 MB/s to a more desirable 20 MB/s to be logged for further analysis. The system responsible for this is the Consumer-Server/Logger (CSL). Its primary responsibilities are to (Figure 1):

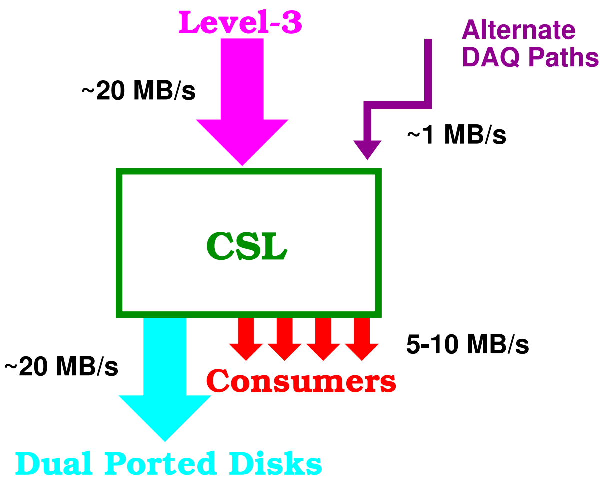

FIGURE 1. Rates in/out of CSL

1. Receive detector event data at a rate of 20 MB/s from multiple senders
2. Write events to multiple files depending on trigger information and event type.
3. Send up to fifty percent of the data to multiple external programs (consumers) which monitor the integrity of the data. Typically they monitor luminosity, trigger efficiencies, and detector channel activity. Consumers may select events based on trigger information.
4. Receive (independent from main data stream) a few MB/s of data from various detector sub-systems such as diagnostics data or calibration data.

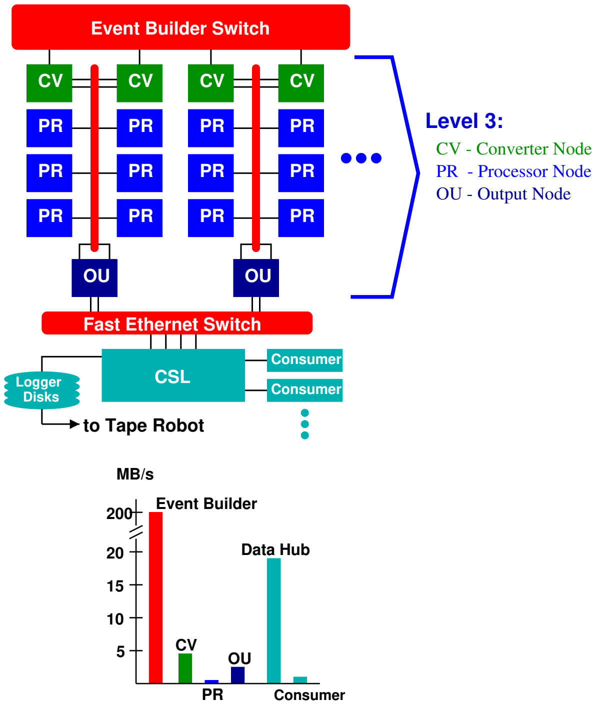

FIGURE 2. Assembled events are sent from the *event builder*, by means of an ATM switch, to *converter* nodes in Level 3, then distributed to available *processor* nodes running physics software (farm), collected and sent to the CSL by output nodes, then logged and distributed to *consumers*. The bottom chart shows typical rates through each component.

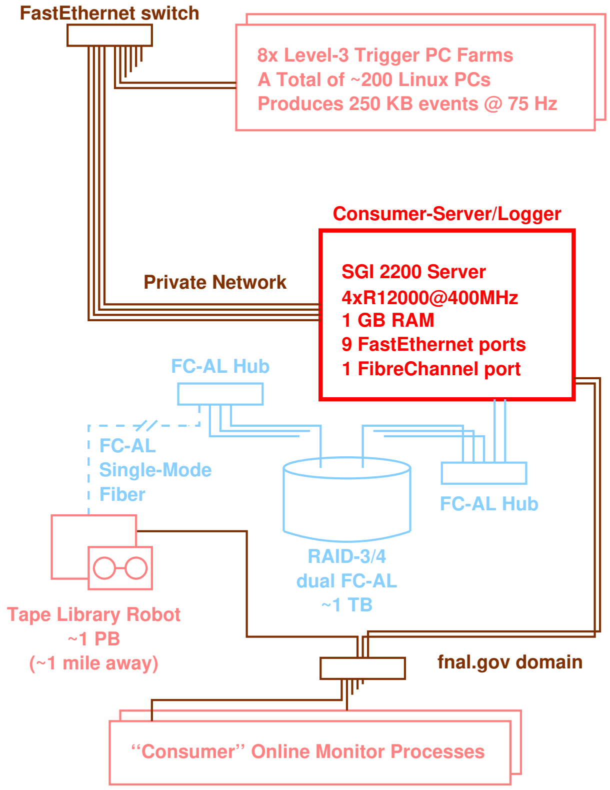

FIGURE 3. Files are written to 1 TB dual controlled/ported RAIDs

Figure 2 shows various data acquisition components, and their relative processing rates.

SYSTEM HARDWARE

Events processed by the Level 3 Farm are received by the CSL from a FastEthernet switch via four FastEthernet ports. The CSL runs on a single SGI 2200 Server with four 400 MHz R12000 processors. It logs data via its FibreChannel port to a series of dual ported RAIDs. Files are then read out from the arrays and sent by single-mode fiber to a robot which writes data to a 1 PB tape archive (Figure 3). Because the connection to the disk arrays is done with two independent fiber channel-arbitrated loops, file output from the CSL is not dependent on readout to tape. This, combined with the 1 TB RAID, mean that data may be stored successfully even if communication to the tape robot is down for as much eight hours.

CSL SOFTWARE

The CSL consists of independently running C programs which communicate through System V IPC shared memory segments and message queues. Collectively, these control the movement of events to logger and consumer output.

Typical operation is as follows (Figure 4): A receiver parent process waits for connections from an event sender. When a connection is made, a receiver child process is spawned to receive events. Consumer sending processes are also spawned for each connected consumer. After receiving an event, the receiver child writes it to one of a hundred shared memory buffers and puts a message on the logger message queue representing the buffer. The logger, after finding a buffer message in its queue, logs the event to the appropriate data files and puts the message on the distributor queue. The distributor checks a consumer request table and either puts the buffer message on the consumer-send queue or returns the message to the receiver queue. The consumer-send process looks for a message on its queue and sends the corresponding event to the appropriate consumers. After this the event has been fully processed and the message is returned to

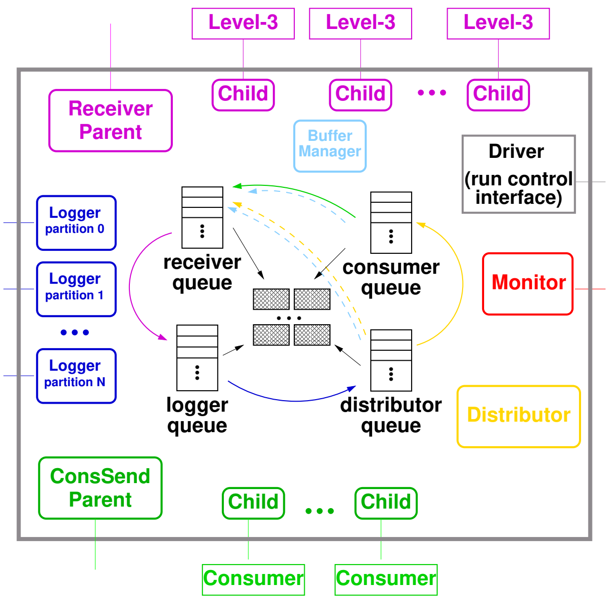

FIGURE 4. Software Design - Events are stored in shared memory, and messages are circulated through message queues from receiver to logger to distributor to consumer sender, and back to receiver. The Driver contols and communicates the CSL state, while the monitor records and sends out rate and process information.

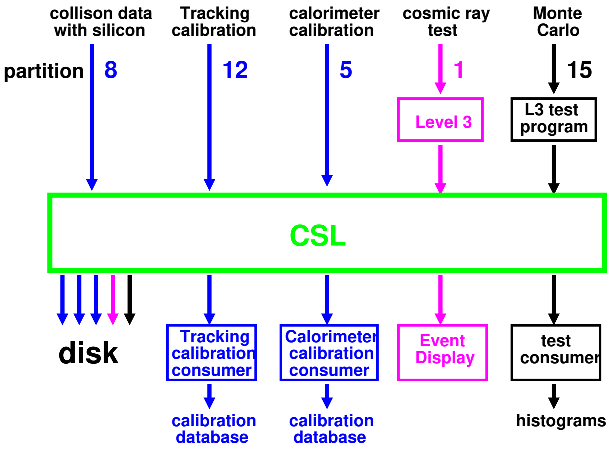

FIGURE 5. Data in the CSL may come from different paths of sub-systems called partitions. Here is an example of an early operations mode. To handle multiple input and output connections, receiver, logger, and consumer child processes are spawned off for each new partition within the CSL.

the receiver queue so that another event may be received. A buffer manager process watches over the receiver message queue to ensure that there are always buffers available for incoming events by removing messages from the consumer queue. In this way, only as many events are sent to consumers as can be done without interfering with the steady input/output rate of 20 MB/s as required.

Each process writes out rate and client information to shared memory and a central monitoring process assembles and sends the information via SmartSockets (www.talarian.com) to a java-based display running externally. A driver process controls the state of the CSL and the interface to the central run control system.

A partition is a sub-set of CDF detectors and/or data acquisition components. The CSL must handle events coming from different partitions. For robustness, a separate logger process handles each partition. Multiple receiver and consumer-send children may join a partition dynamically during the course of a run (Figure 5).

Since the CSL must log and send events to consumers based on physics information, yet not waste time probing their internal structure, events passing through Level 3 are "stamped" with a header structure containing needed event information. When the CSL receives the event, it reads and removes the header, and then adds the information to the event's representative message. Events can therefore be input as just a byte-input stream and written out in sequential ROOT format without actually decoding the event.

CSL STATUS

In a recent CDF commissioning run several months before the start of data taking, the fully operational CSL has exceeded rate specifications by logging 25 MB/s of average-sized events while also sending to consumers.

ACKNOWLEDGMENTS

We gratefully acknowledge contributions from the ROOT Team, CDF Task Force, CDF Online Group, and Rob Kennedy.