The D0 Level 3 High Rate Distributed DAQ System

B. Angstadt, G. Brooijmans, D. Chapin, D. Charak, M. Clements, S. Fuess,A. Haas, R. Hauser, D. Leichtman, S.E.K. Mattingly, A. Kulyavtsev, M. Mulders, P. Padley, D. Petravick, R. Rechenmacher, G. Watts, D. Zhang

Overview

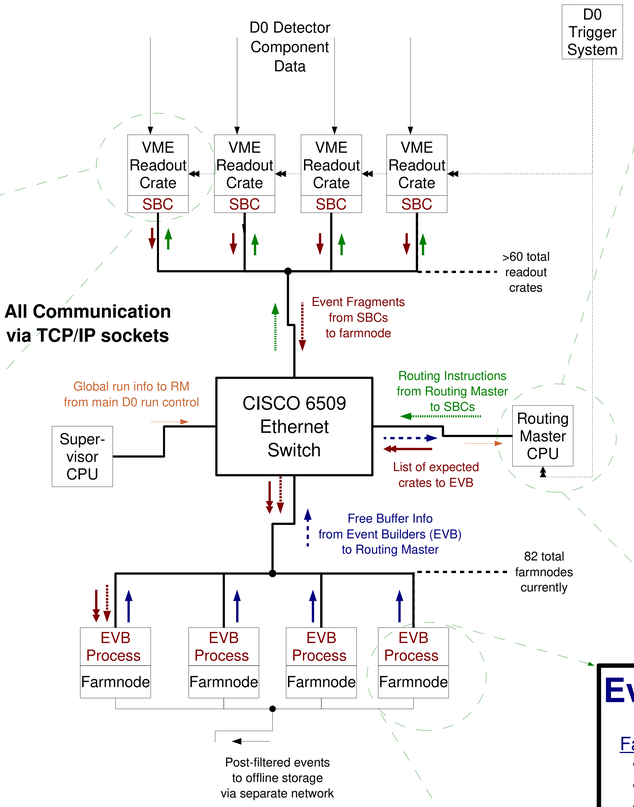
The D0 Detector and Trigger System

The D0 detector is one of two experiments studying high energy proton-antiproton collisions at Fermi National Accelerator Laboratory's Tevatron collider. Up to 7.5 millions occur every second inside the detector, of which about 50 are interesting enough to save for later analysis. The reduction in rate is achieved via a three-level trigger system. The first two levels are based on custom electronics, work on a subset of the full detector data, and reduce the rate down to about 1kHz. The third level consists of a commodity processor farm, currently 82 dual-processor nodes, which applies sophisticated filters to the full detector data (~250kB/event) and reduces the rate down to about 50Hz.

The D0 Level 3 Data Acquistion System

At the time of the second-level trigger decission, the complete data for an event is distributed in 1kB to 20kB fragments among 63 VME crates, each containg 1 to 10 VME modules. A commodity **Single Board Computer** (SBC) in each VME crate reads out the VME modules and buffers the resulting event fragment which is sent over ethernet to a farm node specified by routing instructions received (also over ethernet) from the **Routing Master** (RM) process. The RM chooses a farm node based on the type of first/second-level trigger that fired for the event and the number of available buffers among farm nodes capable of processing that type of event. On the farm node, event fragments are received and collated by an **Event Builder** (EVB) process, which passes the complete event to a third-level trigger filter process also running on the farm node. For each event, the EVB receives a list of crates to expect from the RM. Events that pass the third-level trigger are sent to a storage facility via a seperate network.

With the exception of the first/second-level trigger information, TCP/IP socket connections are used for all communiction within the system using a dedicated ethernet network built around a Cisco 6509 ethernet switch (2Gb/s, 112MB shared output buffer).



Routing Master (RM)

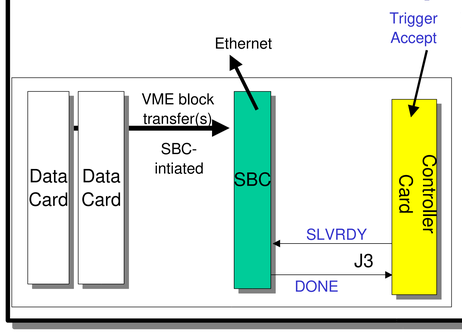
- RM is a software process running on an SBC in a dedicated VME crate
- Actively connects to Supervisor process
  - Receives global run information at start of run
    - Multiple simultaneous runs are possible
    - For each type of second-level trigger decission RM must know:
      - list of crates expected to readout
      - group of target farmnodes
- Pulls trigger decision (bitmask) and event number from D0 trigger system over VME
- Receives socket connections from Event Builders and SBC processes
- On every trigger accept determines which farmnode receives that event
  - Generates and sends route information in the form of Event Tags (ET) to each SBC
  - Sends expected crate list to chosen farmnode

Routing Decission (Event Tag generation)
- Based on free buffers available on farmnode
  - Round-robin among set of farmnodes with most free buffers
- Event Tag contains event number and target farmnode
  - Only 12 bytes of information
  - Queued and sent to SBC

Event Tag Queueing
- Separate software ET queue for *each* SBC connection
- Tags collated and sent as a group to SBC when 10 tags in queue or after 250ms timeout
- 10 tags grouped into one 128 byte packet
  - Necessary to reduce ethernet overhead at designed 1kHz event rate
- Requires SBC to buffer at least 10 event fragments

Trigger Hardware Interface
- INPUT: hardware FIFO on custom VME board(s)
  - Programmable depth--max 128
  - Each row contains 16bit event number and 128 bit fired-trigger mask
  - Read by RM process with single VME reads through UniverseII driver
    - Polled every 10ms
- OUTPUT: 128 bit disable mask to D0 Trigger (to apply "backpressure")
  - Individual triggers disabled via VME accessible registers by RM process when no free EVB buffers
  - Safeguard: all bits automatically disabled if FIFO overflows

VME Readout Crate Example



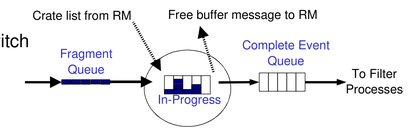
Typical Reaout Crate
- 1-10 data cards (max 20kB per card)
- Total event fragment size from 1kB to 20kB
- Custom controller card
  - Receives D0 trigger accept
  - Triggers SBC to read data via SLVRDY line on J3
  - Receives DONE line handshake from SBC on J3
    - Asserted by SBC when block transfers are complete

Event Building at the Farmnode

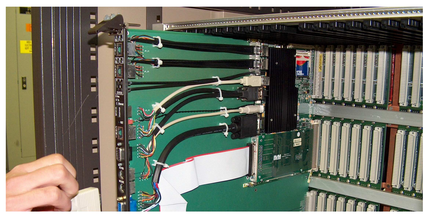
Farmnode Specs
- 48 Dual 1GHz PIII and 34 Dual 2GHz Athlon
- 1 GB memory, Dual ethernet, Linux 2.4.9
- One Event Builder (EVB) process per node
- Two Level3 filter processes per node

Event Building
- Collate event fragments into shared memory buffer based on event number as fragments arrive
- Multiple events built simultaneously
- Event complete (or incomplete with 1s timeout) based on crate list from RM
- Complete events are pulled by L3 filter processes
- Advertise free buffers to RM via socket connection after every in/complete event

Free Buffer Messages
- EVB has 80 events buffers
- But only up to a max of 3 buffers are advertised to the RM
  - This limits number of events in flight through the ethernet switch
    - 82 nodes * 3 events * 250kB/event =~ 60MB
    - Cisco 6509 has 112MB output buffer



Single Board Computer (SBC)



Stats (VMIC 7750)
- Single Processor PIII 1GHz, 128MB RAM
- Dual 100Mb Ethernet
- 128MB Flash Disk
- UniverseII PCI<->VME Interface
  - driver supplied by vendor
- One PMC slot -- filled with digital IO card (DIO)

Custom Linux Driver Module
- DIO card interface
  - Drive DONE line output
  - Interrupt handler (trigger on SLVRDY edge)
  - Drive front panel LEDs
- Universe II driver interface
  - Programs UniverseII DMA engine
    - Initiates VME block transfer -> PCI DMA
- Manages a 12MB buffer (bigphysarea) of event fragments
  - 51 fragments total can be buffered
  - No-copy interface with userlevel process

Event Matching and Routing in the Userlevel Process
- Userlevel process queries driver for next fragments's info
- Event Tags (ET) from RM are queued
- Fragment's event# matched with head ET event#
  - If match, send to farmnode specified in ET
  - If mismatch, drop fragment or ET

Linux 2.4.20 kernel
- Custom kernel trace patch
  - Circular log buffer
  - Integerated with userlevel logging
- Bigphysarea patch
  - Contiguous kernel memory
- Default scheduler

Custom 9U VME Extender
- DIO card -> VME J3 lines
- Front panel LED packs
  - Event#, J3 signals

Custom Linux Distribution
- Size ~33MB
- Net-boot or flash disk

Ethernet Operation
- Userlevel network send call is non-blocking
  - rely on large (1MB) tcp/ip window for buffering
- Dual ethernet operation gives ~24MB/s
  - Each EVB makes two TCP/IP connections
  - Userlevel process calls bind_to_socket and alternates sending between sockets
  - Both interfaces in same subnet
    - Requires small custom kernel patch to force only one interface to respond to arp request

Performance

SBC
- CPU usage
  - At 1kHz event rate, crates with 20kB event fragments can approach 80-90% CPU usage
  - Optimizations under investigation
- DIO card
  - SLVRDY-DONE handshake tested at 15kHz

Ethernet Switch
- Cisco 6509 has seen up to 30% utilization
  - Corresponds to D0 runs with 300kB/event at 400Hz
- Can accommodate up to 200 farmnodes in the future
  - Need to reduce EVB advertised buffers to two
    - 200 nodes * 2 buffers/node * 250kB/event = 100MB

Routing Master
- Routing decission takes up to 700us (<500us avg)
  - Hardware FIFO read ~20us per event
  - Adequate for 1kHz event rate, but algorithm implementation could be optimized
- Network latency reduced by Event Tag grouping
  - Tested up to 2kHz with ~30 crates

Event Builder
- Consumes <15% CPU on slowest farmnode variety
- Algorithm and memory management not fully optimized