Overview of CDF's L2 Trigger

- General trigger system details
  - Front end HW implements 4 buffers for Level 1 event data
    - Each buffer holds detector information from one event
    - data : L1 bits, Muon, Calorimeter, Track, Electron, SVT
  - Data sent to L2 system on L1 accept (~40kHz)
    - ~20us delay btw arrival of non-SVT & SVT data at L2
    - *Deadtime* occurs when all 4 HW buffers are occupied
  - The L2 system ...
    - Runs trigger algorithms on L1 data & forms a decision bitmask
    - must return decisions in *event order* (~300Hz)
- The upgrade will use ...
  - 2 cards (non-SVT,SVT) w/ filar FW to receive data from L1
  - 1 card w/ solar FW to transmit L2 trigger decision
  - Fast dual processor 64bit AMD Opteron(s)

Possible Upgrade System Configurations

- 4 buffers → 1 CPU
  - Serial event processing
    - Block on the non-SVT filar's fifostat reg, then on the SVT filar's
    - Run L2 algorithms on the entire event, then process the next
  - Interleaved event processing
    - Poll both filars, possibly readout non-SVT filar several times before SVT data arrives.
    - Can run non-SVT algorithms for several events while waiting for SVT data.
- 4 buffers → 4 CPUs
  - Independent event processing
    - Each CPU processes serially, won't see another event until it finishes processing its current event.

S32PCI64 Usage in the L2 Upgrade

- Our software adapted from CERN code
  - Filarscope: Jun 14, 02
  - Solarscope: Apr. 2, 03
  - io_rcc: Dec. 12, 01
  - cmem_rcc: Jun. 7, 02
- Software changes
  - Modified filar/solar code to support multiple devices
    - Took advantage of data structures for unused channels
  - io_rcc and cmem_rcc drivers ported to linux 2.6
    - Use the new kernel driver build process
    - Minor compatibility changes to the code
  - Ported again to linux 2.6 / AMD64 platform
    - Requires changes in memory mapping and VMA's

Wish List

- FW modifications to reduce latency
  - One-time configuration of filar reqfifo memory addresses?
  - Fifostatus in SW rather than a PCI register?
    - Implement using a separate SW buffer, to be accessed like the data?
  - Data transfer to specific SW buffers based on S-Link header bits?
    - Allows data from our 4 L1 buffers to be directed to different memory locations
    - We wouldn't have to access the data to determine which buffer it belongs to (important in the 1 CPU, interleaved scenario)
- A 4-channel FILAR
  - When can we expect it?
  - SW interface shouldn't be too different from what we're using now?
- FW glitch?
  - Sometimes solar's opstat register tell us no slots free