L2 Global
Status and Opportunities

James T. Linnemann
Michigan State University
London Workshop
January 6-8, 2000
L2 Trigger

- 10 KHz L1 out to 1 KHz L2 out
  - 128 L2 decision bits, 1:1 with L1
  - few % deadtime

- Global Processor selects events
  - threshold for object
  - matching objects from different detectors
  - cuts on quality
  - kinematic variables (but Zv=0…probably)

- Objects from single-detector preprocessors
L2 Preprocessors

- Cal em  TT clustered em pt, $\eta$, $\phi$, quality (L1)
- Cal jet  TT clustered jet pt, $\eta$, $\phi$, quality (L1)
  - Ç no ICD
  - Ç large tiles only from tower seeds
- CFT   pt, $\phi$, quality (no uv so no $\eta$)
  - Ç tags with CPS
  - Ç STT: confirmed CFT tracks; impact parameter-oriented lists
- FPS   E, $\eta$, $\phi$, quality (upstream*downsteam)
- Mu    pt, $\eta$, $\phi$, quality
- No vertex position information ($\eta_{detector}$)
L2 Global

VME: < 1 KHZ
L3, Monitoring to TCC via DP Mem

Magic Bus > 10 KHz
L2G Inputs So ... Sn
Admin/Worker Comm
L2 Answer to L2 HW FW

Scaler gates
L1, L2 HWFW

SCL (Includes L1 Qualifiers)
L2Global Hardware

- Alpha processor for Worker
- Another for Administrator
- Add more Workers if can’t keep up
  - hidden from user software
L2 Global Software

- Above user level: buffer handling, monitoring
  - I/Ogen: a great deal of data access and formatting hidden from user

- Above user level: script interpretation
  - L2 script runs iff its L1 fires
    - speed, and efficiency independent of other triggers
  - 128 bits 1:1 with L1 for now
    - fanout later if needed (would need pre-scaling)
  - interface with user code defined
  - script runner implemented
  - parameter initialization still not implemented

- Working on adding to simulation
L2 Infrastructure

- 60 nsec time budget for L2G overall
  - tight coding required
  - budget per algorithm less crisp--but shorter!
- C++ with static allocation, no user I/O
  - see L2 coding guidelines; error logger allowed
- Considerable effort made to hide complexities
  - buffers hidden; IOGen; classes for h, j
- modified Linux (eases debugging)
  - Linux turned off when run starts
    - maybe even during run downloading
L2 Global Algorithms

- Some rough C++ code exists (timing tests)
  - not using real interfaces, environment
- Now ready to begin drafts of final algorithms
- L2Global algorithms differ from L2 Preprocessors algorithms:
  - don’t run every event
  - many accept multiple input object types
  - variable input parameters from script (instances)
- Dylan & Roger’s talks for details
  - writing, and adding to simulation
Doing a L2 Algorithm
includes liaison with ID group!

- Coding is demanding, *but a small part*
  - MSU: technical assistance, or even final rewrite if necessary
- Verify algorithm *(each new version)*
  - in simulation, before going online
    - test data, automated release tests
  - in test stand (larger data volumes); analyze results offline
- Monitor Online
  - make L3 EXAMINE histograms
  - compare simulation v.s. actual results
- Tune cuts *(less relevant for simpler algorithms)*
  - as part of ID Group: define baseline cut values
    - possibly different tunings for different physics topics
- Measure, parameterize turn-on curve *(mostly indep. of physics)*
  - efficiency v.s. Et offline / Thresh L2 (resolution, tails, relative scale)
  - parameterize v.s. Et, $L$; put in parameterized MC
3 Types of L2G Object Algorithms

- **Match**
  - create by matching preprocessor objects
    - $\zeta$ threshold, # objects, quality cuts
  - $e, g, m$
  - jets, $t$ (?)

- **Cut**
  - # CFT tracks with $p_T > x$
  - # tracks with impact significance $> x$

- **Relation**
  - Transverse Mass $> x$
  - $D_j < x$
Match

- Relatively complex:
  - multiple object type matches (but known in advance)
  - more cuts to study
  - will require interaction with preprocessor experts

- electromagnetic e, g:
  - matches among cal, ctt/stt, cps, fps
  - exact relation of e, g code to be determined

- matches among mu, ctt/stt (vary with quality req’d)
  - slow mover tag from scintillator timing?

- jets: is this just a cut on L2CAL output? Tracks too?

- ID group should define needs
  - or just further cuts on jets?
Cutting Tools

- Simplest
  - single object type, known in advance
  - mostly coding, technical verification
  - still, cut studies, turn-on curve measurement
    probably by preprocessor group

- Etmiss

- Tracks
  - CFT? Require STT confirm?

- Tracks with impact significance

- Jets (if just calorimeter jets)

- t? (even if cal-only, new cut studies needed)
Relations: t 2 L2G objects from previous tools in script

- Complexity similar to cutter but no preprocessor team
- Different input: pointers to passed object lists
- Objects of different generic type
  - Not known in advance: technical issues
- Rapidity range: a degenerate case (n=1?)
- Transverse mass, invariant mass
  - Need $Z_v$ or an approximation strategy
  - Studies, and parameterization needed
- $D_h$, $D_j$, and $D_R$ (=isolation)
  - $<x, >x$, and $180^\circ$ variants (Jet co-linearity, etc.)
- $H_T$ (from jets)
Speculative: not clear when/if done

- Zvertex
  - Define, then cut on?
  - Use in Et definition?
  - L0? Potentially STT ZV?
  - # primary vertices? Same issues
  - Luminosity-dependent efficiency--lots of studies!

- Displaced Vertex
  - x-y vertex finder among displaced tracks?
Responsibilities and Opportunities I

**Matching**

- MSU will do e, g tools:
  - probably most complex
  - with SUNYSB for PS, low Pt; UIC on Cal
- need **volunteers for mtool**
- Jet, and t tools
  - volunteers needed for both
  - also need **crisper definitions** from/with ID groups
  - are these a small or large amount of work?
Responsibilities and Opportunities II

Cutting and Relational

- MSU: simple # CFT tracks > Thresh: sample cutting
- MSU will do transverse mass: sample relational
  - SUNYSB collaborates on invariant mass tool, esp. low mass

- Need volunteers for rest of cutting and relational tools

- If no volunteers mid-Feb, MSU will code 1\textsuperscript{st} versions
  - so simulations can proceed
- Even if MSU codes 1\textsuperscript{st} versions, may still need help
  - cut tuning, parameterization, monitoring