

Murrough Landon 5 November 2008

- Overview
- Tools
- Database
- Run Control and Module Services
- Usual Suspects
- To do list for the shutdown



- Main changes since the last joint meeting
 - actually the basic software has been fairly stable
 - moved to tdaq-01-09-01 (for collisions!)
 - major update of COOL database schema, mainly for PPM
 - added rate metering for PPM and CMM
 - see talk by Felix
 - more recently also reading PPM histograms
 - various improvements to monitoring
 - see talk by Taylor
 - further development of calibration procedures, mainly CPM
 - see talk by Joseph
 - many little changes
 - ROD related SW, hardware byteswapping, etc



- Mapping Tool
 - added display of rates from IS
 - recently also statistics from PPM rate histograms
- I1rates
 - tabular summary of many level 1 rates
 - no histograms, but faster and more detail than trigger presenter
- ACE
 - last batch of improvements from Alvin
 - better time range selection, handling of different types
 - will have less support in future, alas



Tools (2)

- I1chuck.py
 - still a work in progress
 - selective interface to DB for killing hot towers
 - update DB and allow on the fly changes (making a new lumi block)
 - latter not yet implemented
 - present version updates
 both PpmDeadChannels and
 PprChanCalib folders
 - better not to change LUT noise cut from calibration but add override value?

L1Calo Hot/Unwanted Channel Killer

Channel Specification
pp 5 ppm 0 chan 31
COOL ID 05100703 Read DB
Dead Channel Status
Bits that zero the LUT
🔄 Bad Adc 🔄 🔄 Bad Mcm
☐ FE Dead
📕 Noisy Tower
-Flags for layers with masked cells
□ Presampler □ A/Front □ BC/Middle □ D/Back
Comment Noise rate about 500Hz
Channel Calibration
Bits that zero the LUT
🔲 No Data 📑 Bad Data 📑 Bad Result 📑 Reserved
LUT Noise Cut 10
Commands
Update DB Zero LUT Now!



- Readout monitoring
 - rodmon application previously used to compare simulation and hardware in playback tests
 - recently extended to provide monitoring of ROD fragments publishing summaries of errors to IS
 - complementary to other monitoring producing histograms
 - should we envisage combining this IS info with other ROD information, making a better readout diagnostic panel?
 - eg combine/revamp/improve the L1Calo Busy panel



• PPM schema changed after summer discussions

- keep DAC scan results as floats
 - online calculation of correct DAC for expected pedestal
- but store other information as precalculated integers
 - ensures offline has identical values to those used online
- now have separate folder for dead/killed PPM inputs
 - those not due to internal L1Calo calibration problems
 - still bringing this slowly into use
- already realise need for modest further changes
 - noise cut from pedestal run
 - override via dead channels folder
- NB need stable schema before moving to use the production COOL database!

- though we could try using non-production Oracle DB



- Various other schema updates
 - extra run parameters attributes for some scans
 - bypass PPM LUT in playback (and set trivial FIR coefficients)
 - disable module status updates during CAM scans
 - common calibration error code attribute implemented
 - similar code for dead/killed channels
- (Not much) Validation
 - but no further work done on validation of calibrations
 - and still need mechanism for tracking module IDs
 - nor of code to extract/plot time history of values



• Recent L1Calo schema changes

- removed old OKS calibration
- cleaned up OKS run parameters
- added defined sets of TTCvi/LTP configurations
 - selected by user via the L1Calo RunPars IGUI panel
- Forthcoming TDAQ changes
 - "Check in" scheme for OKS database at point 1
 - No changes required outside point 1
 - It will no longer be possible just to write or edit new or updated files to the right directories - they will have to be checked in via a new CVS-like application
 - If valid, they will then be made available on a readonly filesystem (only writable by a OKS server)



- Run controllers now publish "full statistics"
 - used for rates (every 2s) and PPM histograms (less often)
 - additional objects for statistics of PPM histograms
 - responsibility in several packages:
 - statsL1Calo: containers for IS variables and ROOT histograms
 - module services: fills information using statsL1Calo API
 - rcL1Calo: takes care of publishing the results
- On-the-fly commands
 - To support on the fly changes of noise cuts, disabling of hot channels (anything else?)
 - Support for set of known commands in run controller
 - Passed to module services for implementation
 - External SW (scripts) needs to take care of changing lumi block



- We finally moved to hardware byteswapping!
- And have started talking about enabling bus errors...
- But we have not written much more documentation
- TDAQ versions
 - currently using tdaq-01-09-01 (01-09-00 now unusable due to improvements in monitoring
 - tdaq-02-00-00 is imminent (and 02-00-01 in February)
 - new version has yet more changes to the IGUI API
 - this time with more consultation than when tdaq-01-09-00 hit us



• Run control states reorganisation?

- We are currently using "substates", keeping the ancient "Load" and "Configure" steps as our internal states
- TDAQ implementation of substates has some problems
 - cannot restart our low level controllers, only top level L1Calo
 - error states are not properly recovered (needs TDAQ rewrite!)
- Would like to reorganise our module services code a bit so we can use single "Configure" state to do all configuration of module (apart from its connections)
- Use "Connect" transition for connections between modules
 - also for loading generated test vectors from the simulation?
- Good time to review, clean up and optimise (if possible)
- Need to check if there are really any "gotchas"
- Will also need time to test carefully



• Build/development environment

- Fix dependencies in CMT requirements files so we can do parallel makes (hope to improve elapsed time for builds)
- Should also make it easy to check out and build only one or two packages (not really the current model)
- Proper patch procedure for point 1
 - Original idea of tagged builds for point 1 completely undermined by little fixes and library rebuilds
 - useful, but means we never really know what SW we are running
 - Aim to make patches only from new tags
 - Disallow changes to header files (requires complete new build)



Errors and ERS

- The full ATLAS partition generates a huge amount of error messages from ERS
- But L1Calo is probably too quiet
 - Some important problems only get reported in verbose log files which may never be noticed
- We should review all this and ensure we do issue suitable error messages via ERS (so visible to the IGUI) for real problems
 - Long term intention has also been to convert HDMC and module service exceptions to the ERS exception framework

Bus errors

- We still ignore bus errors: should try reporting them!



- Automatic disabling of LAr/Tile inputs to L1Calo
 - The present L1Calo IGUI CaloMask panel allows individual
 LAr or Tile regions to be disabled from the trigger
 - But this must be done by hand (after the partition is booted)
 - After recent discussions with a few interested parties we propose to automatically disable LAr or Tile trigger inputs if their corresponding TTC partitions are disabled
 - Scheme uses resources and resource sets (needs some L1Calo SW)
 - http://indico.cern.ch/conferenceDisplay.py?confId=43097
- Automatic disabling of RoIB inputs
 - The same discussions also proposed a similar automatic scheme to disable RoIB inputs if eg L1Calo is disabled
 - Implementation requires SW work on the RoIB side
 - BUSY issues need RoIB firmware work (no effort available)



- Conditions database
 - So far we only read data from COOL
 - We also need to write conditions data for each run
 - IS parameters (still not done although fairly easy)
 - Module IDs eg at start of run
 - Timestamp when calibration data was read
 - This may be earlier than the run start and there might be database updates made between the configure step and start of run
- Archiving
 - Longer term archive for rates
 - Use of Monitoring Data Archive (MDA) for calibrations
- Application parameters
 - Take more application parameters from DB instead of hard coding them...



PPM Calibration Attributes in COOL

• Common

- ModuleId [UInt32]
- ErrorCode [UInt32]
- *ResultsTimeStamp [UInt63]

DAC/Pedestal

- DacOffset [Double]
- DacSlope [Double]
- DacCrossCor [Double]
- PedValue [UInt32]
- PedNoiseCut [UInt32]
- PedMean [Double]
- PedSigma [Double]
- PedNumEvents [UInt32]
- Coarse Timing (R/O Ptr)
 - SyncDelayData [Uchar]
 - SyncDelayBcid [UChar]

• Fine Timing (PHOS4)

- InDataNegedge [UChar]
- Phos4Delay [UChar]

• BCID

- InBcidNegedge [Uchar]
- ExtBcidThreshold [Uint16]
- SatBcidThreshLow [Uint16]
- SatBcidThreshHigh [Uint16]
- SatBcidLevel [Uint16]
- BcidEnergyRangeLow [Uint16]
- BcidEnergyRangeHigh [Uint16]
- BCID
 - FirCoeff* [UChar]
 - FirStartBit [Uchar]
- LUT
 - LutOffset [UInt16]
 - LutSlope [Uint16]
 - LutNoiseCut [Uint16]
 - LutStrategy [UChar]
 - LutParameters [BLOB]
- Notes
 - In *Results folder only