Online Database Developments

Murrough Landon
17 October 2007

- Overview
- OKS database status and plans
- COOL database developments
- Validating calibrations
- Tools
- Summary

Overview: online databases

- OKS
  - Configuration of the TDAQ and online system
    - Description of partition, segments, SW, applications, HW
    - Hardware includes crates, modules and cables connecting them
    - Presently also includes "static" module settings, run types, test vector specifications (so called "DataGenRecipes")
    - Also has old style calibration data and trigger menu
- COOL/CORAL
  - New trigger configuration DB (CORAL)
  - Calibration data (partly done)
  - Definition of run actions
    - May (partly) replace OKS run types?
    - Would also need specification of test vectors
  - Should also have configuration choices for modules

OKS Database Status

- Point 1 segments reorganised (fewer controllers)
  - Simulation is a "controlled application" (no segment)
  - Separate segments for monitoring and calibration
  - All ROS applications in a single segment
- OKS configuration of GNAM processes now works
- SW and FW descriptions part of the SW release
  - Now using automatically generated software description
  - Hand edited firmware description moved to L1CaloRelease

OKS Database Plans

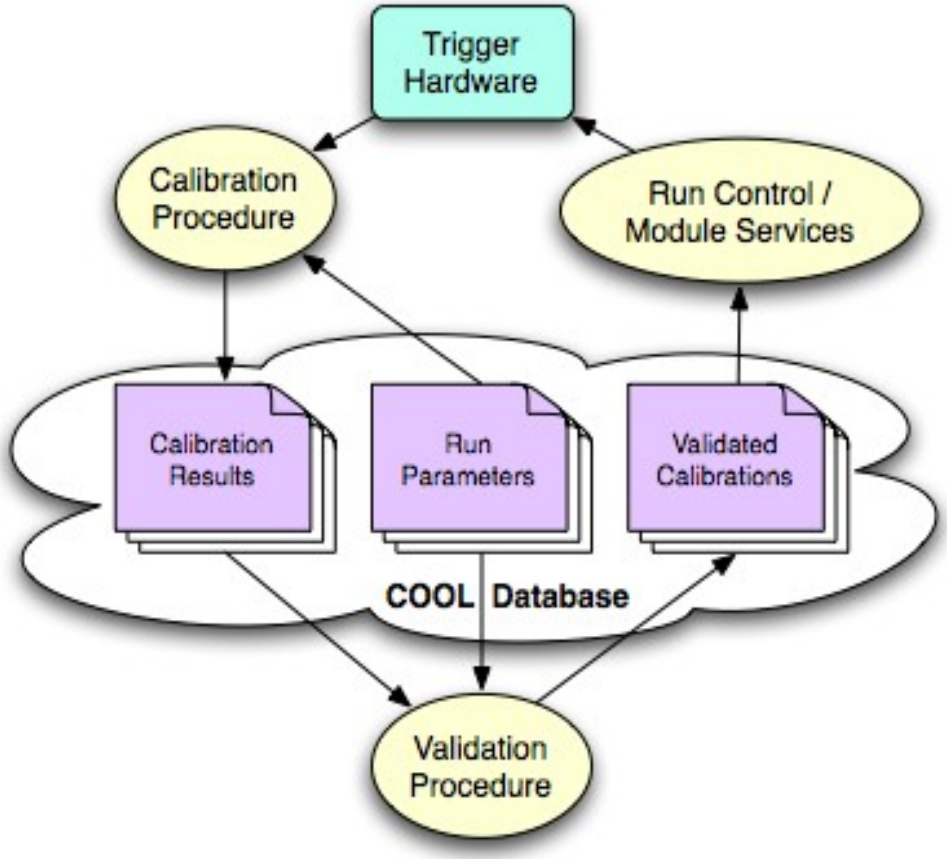
- Generation script still needs to be finished
  - Module settings, separate cables files
- Make our Modules and Crates into Resources
  - Enable/disable from Segments & Resources panel
  - Can then remove L1Calo ModPars panel from IGUI
- Use Cable classes in tdaq-01-08-xx schema
  - Allow automatic ROD/ROS connection
  - Delayed due to error in tdaq-01-08-01 schema
- Use TDAQ/LAr tool to extract cables from TC DB
  - Assuming such a tool gets developed and is usable
- Still intend to slim down run types in OKS

COOL Database Overview

- We already have several different categories of data
- Folders of results from each calibration run
  - One (or more) folder for each different type of run
    - May have module/submodule/channel level results from one run
- Folders of validated calibrations
  - Validation process may combine results from several different types of calibration run
  - One folder for the complete set of calibration data for a particular module type
- Folders of run parameters
  - One folder per type of calibration (or test, or physics) run
    - Definition of multistep run sequence
    - Analysis parameters (eg desired pedestal, quality thresholds, etc)
    - Could also include criteria for validating a new calibration?

Calibration Procedures and Folders

- Calibration procedures
  - controlled by run pars
  - write to Results folders
- Run control
  - uses Validated folders
    - may use Results folders if requested by run pars (PPM) and suitable SW is written
- Validation
  - controlled by run pars?
  - copy/merge results to validated calibration



COOL Validated Calibration Folders

- CmmCalib, CpmCalib, JemCalib:
  - all values for one module (ttcrx, readout)
  - for CMM: backplane and cable timings
- CpChipCalib: phases for each CpChip
- SerialiserCalib, JemInputCalib:
  - phases and delays for input channels
  - also, at present, masks for disabling JEM input channels
- CpmDeadChannels, JemDeadChannels
  - list of channels to disable (keyed by eta,phi,layer)
- PPM still to be finalised
  - Set of "validated" folders was defined (PPM,MCM,Channel)
    - But now only using these to provide default values
  - Currently combining values from Results folders

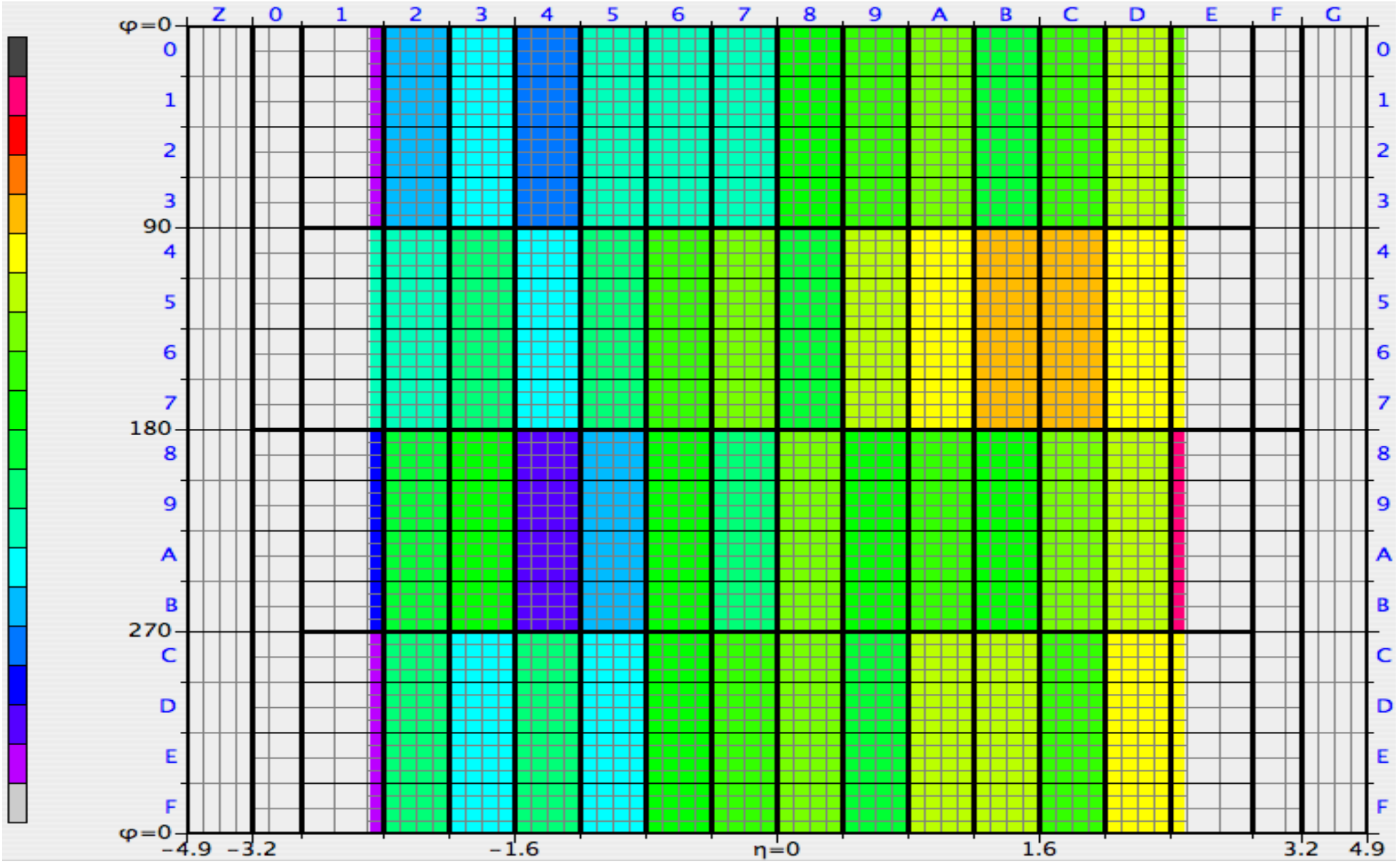
Validating Calibrations

- Validation Procedure (same/similar for all modules)
  - Set of criteria to apply
  - Update whole calibration at once or channel by channel
    - Former probably needed for readout pointers, real time delays
    - Latter probably fine for PPM DAC settings
  - Where to do it?
    - Copy offline calibration results to online DB and validate online?
- Validation Framework (C++)
  - Common software framework if possible
  - Read run parameters and process folders
    - May need custom classes in some cases
    - Default behaviour otherwise
- What we have right now…
  - Blind merge of results to validated folder ("useCalib")

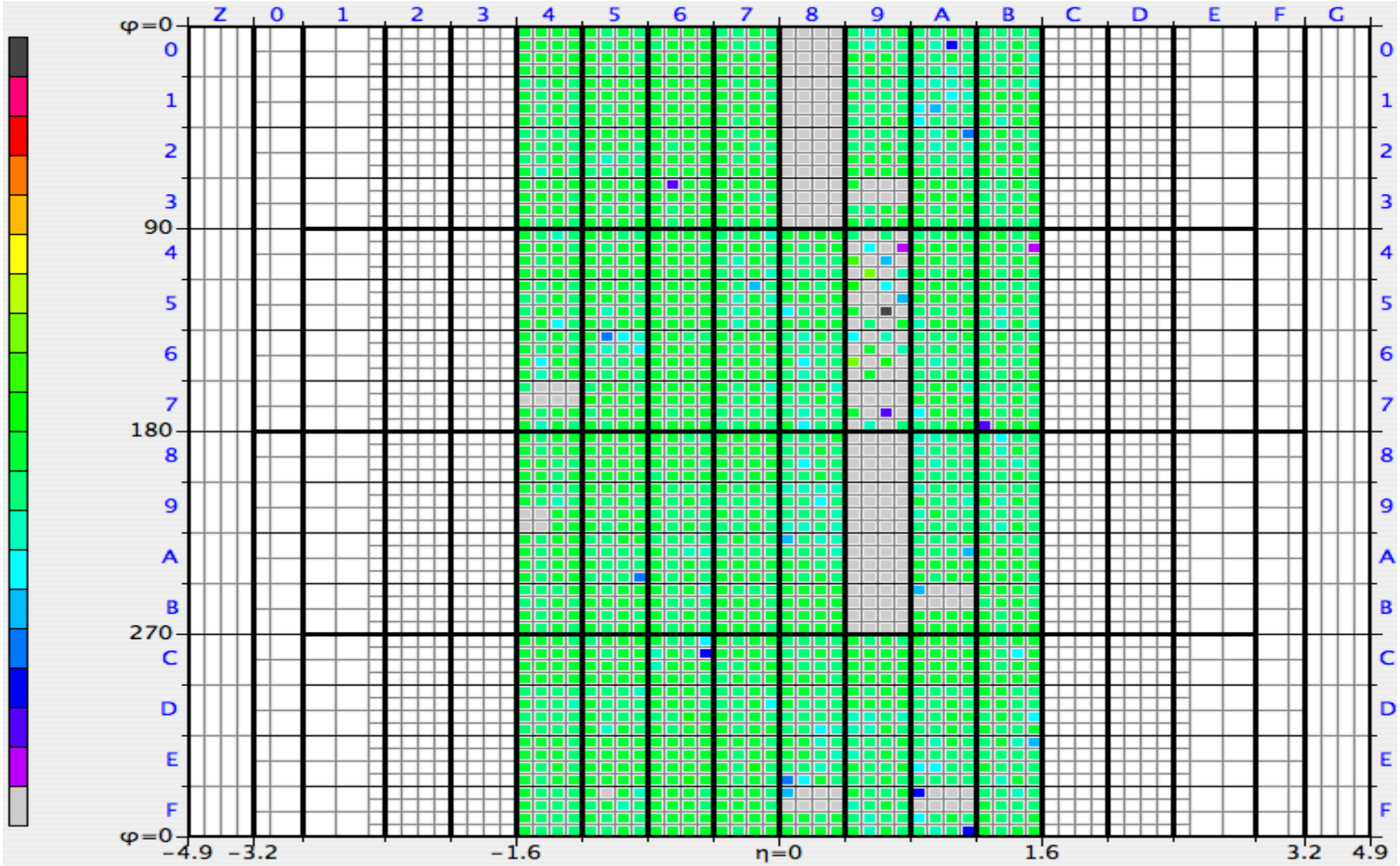
COOL Tools

- Standard tools (KTIDBExplorer etc)
  - But KTIDBExplorer doesnt support SQLite
- New L1Calo COOL editor from Alvin...
- Meanwhile an array of python scripts
  - listfolders.py: list all our folders (and their attributes)
  - dumpfolder.py: show contents of a folder
    - optionally in format for loadfolder.py or l1calomap tool (XML)
  - loadfolder.py: load a folder with contents of a file
    - python version of testInitCool from Florian with same syntax
  - dropfolder.py: start again from scratch (use with care!!!)
  - createfolders.py: create complete set of empty folders
    - will be needed for Oracle as "writer" account is not the "owner"
- Basic visualisation via l1calomap tool (via XML file)
- Still need tools for trends, analysis, etc

l1calomap: CpmCalib/ttcrxPhase1



l1calomap: PprDacScanResults/Slope



Summary

- Some OKS changes
  - More to come
- Good progress on storing calibrations to COOL
  - But still not finished
    - Test latest JEM work
    - Finalise PPM folders
    - Dead channels discussion
  - Make wider use of run parameters folders
    - Eg CPM, JEM calibration scans
  - Need more optimisation (esp PPM) when reading DB
    - Speed up run control, simulation and other processes
  - Move active DB from SQLite to Oracle for offline
- Need to develop validation procedures
- Still need more work on tools