DB: Run Control and Others

Murrough Landon
22 May 2006

- Context
- Trigger Menu
- Connectivity
- HW, FW, SW configuration
- OKS Archiving
- Run Control
- Organisation

Online Database Context

- Online DB Task Force
  - Reviewed actual/planned use of DBs online
  - See various talks by Steinar Stapnes
  - Standard solutions recommended, custom solutions possible but subdetectors have to commit to maintain them for the lifetime of ATLAS
- Development/Production Servers
  - Devdb10 available for development
  - New DB application (eg custom solution) needs to be benchmarked before moving to production server
  - Production schema changes only by DB administrator?
  - ATLAS does now have some in-house Oracle experts

Trigger Menu

- Database
  - CORAL application with many "write once" tables and complex web of relationships from top level Trigger Menu to L1Calo thresholds (and muon, HLT, etc)
  - Used trigger menus are marked and linked in COOL
- Tools
  - C++ library with API, similar(?) to our existing one
    - Should not be hard to migrate – but not tried so far
    - NB Alan recently requested addition of FADC scale, so PPMs will now also need info from triggr menu
  - Java GUI to edit the DB, expert and shift level
  - Still being developed, but Johannes Haller is moving to DESY (still working on ATLAS TDAQ though)

Connectivity

- Online requirements and considerations
  - Granularity expected by crate controller and online simulation is at the level of cables (ie not pins)
    - Each cable is an object with relationships to module objects at each end and names (string) of their connectors
    - Modules are objects in hardware configuration
    - Connector names are "known" to the code
  - Assume one query to get all relevant cables
    - Eg all of a given type, or those connected to a given crate
    - Or could continue to use OKS as cache
  - DAQ group may describe Slink/Busy cables in future
    - Nice if we could converge on a single Online approach....?
  - Editing: presently by experts using text editor
    - Other implementation needs better tools (not plain SQL!)

Hardware Configuration

- OKS: now and in future?
  - Now using OKS extended hardware schema to describe our crates and modules (also some hardware parameters)
  - Used in crate controllers, simulation, IGUI, scripts, ...
  - Prefer to keep OKS implementation, even if only as cache
    - Could remove all hardware settings to COOL/CORAL?
    - But need links between HW modules and COOL/CORAL data
      - Settings, run plans, test vectors, etc
- Link to TC databases?
  - Some discussion in TDAQ of extracting HW configuration from TC installation DB – at least for PC farms
  - But we still need local non-TC description for test rigs

Firmware/Software Configuration

- Software description
  - TDAQ now use CMT macros to generate description of SW configuration, ie binaries built by CMT, scripts etc
  - We still maintain it by hand, but I would like to change this
- Firmware configuration
  - Could also look at generating FW description in similar way from macro definitions in CMT requirements file
  - I would also like to move FW binaries out of the software CVS repository one day, eg into COOL?
    - NB this is nothing to do with storing firmware source!
    - Would need simple-to-use tool for firmware designers to put new versions of firmware binaries into the database
    - Software releases should still have links to which firmware is needed with the corresponding module services version

OKS Archiving

- Oks2coral
  - Low level relational backend to OKS
  - Pure CORAL, but saved partitions linked in COOL
  - Can use it instead of existing XML files or RDB
    - Can access offline access, but need same DAL as used online
  - Or to (incrementally) save and retrieve partitions
- Oks2cool
  - Stores (subset of) OKS database in COOL folders
  - Retrieval still under development (problems with type mapping and schema evolution)
  - Easier access offline (no need for DAL)
- Both
  - Partition will be saved using both tools at start of run

Run Control: Overview

- Use of databases
  - RC uses HW (modules&cables), FW, SW descriptions
  - Some hardware settings (choices not calibrations)
  - Calibration and trigger menus
  - Description of run types (or run plans)
- Current organisation
  - Single OKS partition defines everything except calibration and trigger menu which come from separate files (allowed list defined the partition DB)
  - Data presented as scalar values or objects either through L1CaloDatabase object or DbXXX object for each module type
  - In RCD, singleton class handles all DB access

RC: Moving to COOL/CORAL (1)

- Trigger Menu
  - Query new trigger menu API instead of reading OKS file and then just fill the old objects?
  - Or just use new trigger menu objects which should have similar API (to be checked)
- Calibration
  - Two separate empires: PPM and the rest
  - Could migrate PPM separately from the rest, but would prefer to do CMM, CPM, JEM together
    - Implies defining all PPM or "rest" tables at once
    - Also need the tools to fill/browse the new tables
  - Easiest path, fill old objects from new DB queries
    - For each COOL table, query for all "channels" in one crate at once (whole system for simulation)

RC: Moving to COOL/CORAL (2)

- Non calibration settings
  - Unlike existing calibration objects, these are all separate scalar settings in DbXXX objects
    - Bit more tedious to fill from results of DB query
    - Might involve more work on DbXXX internals
  - There was (largely unused?) idea to include some such parameters into run type definitions
    - Might imply connection with new run plans
- Run plans
  - Present run type objects define

RC: Moving to COOL/CORAL (3)

- Run types and run plans
  - Present run type objects define:
    - Test vectors (data gen recipes)
    - Sequencer parameters (N.steps, parameter to change)
    - Run type dependent module settings (eg N.slices years ago)
  - Read as object structure
  - Sequencer parameters used to update calibration objects (which are available via DbXXX) and PpmCal structures at beginStep transitions
  - Probably quite some work to migrate...

Run Control: DB Summary

- Requirements
  - Efficient readonly bulk queries
    - All data for one crate for crate controller
    - All data in the system for simulation
- Proposals
  - RC issues queries based on types of modules in crate
    - Using a few common classes shared with other packages
      - Hide whether implementation is COOL, CORAL or mixture
      - Keep details of folder structure in one place for RO and RW access
  - Get results as vectors of CORAL AttributeLists
  - Unpack into existing or modified L1CaloDB objects

Run Control: Workplan

- Short term (end July?)
  - Try new trigger menu ASAP
  - Develop alternative "reader" for non-PPM calibration objects via COOL/CORAL queries
    - Already separated existing reader into Oks-only subclass
    - NB need custom crate + CPM/JEM/CMM at CERN to test it
  - Develop COOL/CORAL "reader" for PpmCal structures (with help from Florian?)
- Medium term (autumn)
  - Not sure
    - Non-calibration settings? Run plans?

Online DB Organisation

- People, places and packages
  - The different strands of existing L1Calo online DB developments are quite tightly coupled
  - Not easy to decouple and coordinate work remotely
  - More DB experts weeks, working together in one place would help
  - Need to avoid overlap/duplication of code or classes