Databases (SW Meeting)

Murrough Landon
8 November 2006

- Overview of ATLAS databases
- Status of recent work
- Proposals for calibration databases
- Connectivity

Database Glossary

- COOL: Conditions Database for LHC
  - API to tables where each row has interval of validity, ie history of changes with time (and multiple versions of that)
- CORAL: COmmon Relational Access Layer
  - API to generic relational database with implementations for Oracle, MySQL, SQLite and Frontier (http)
- POOL: Persistent Objects for LHC
  - Unique tokens (and catalogues?) for distributed files
- SEAL: low level services
- OKS: Object Kernel System
  - Object oriented DB used in online/DAQ software

ATLAS Databases

- Equipment and installation
  - MTF (objects), Rack Wizard (locations), Cable DB
- DAQ Configuration (OKS)
  - Configuration of modules, online applications
  - Also L1Calo cabling and run types (at least for now)
- Conditions Database (COOL)
  - Calibrations (results and next one to load)
  - Record of user choices and status info from each run
- DCS configuration and archive
  - Also has link to COOL
- Trigger configuration
  - API hiding relational DB behind it (using CORAL)
  - Now provided as DetCommon ATHENA package :-(

Recent Activity

- MTF
  - Just restarted impetus to enter our equipment
  - Barbro will try to do this
  - SW improvements in tools over the last year
  - Better web interface (AtGlance) to search and download
- Configuration DB (and online status) archiving
  - Recent TDAQ releases include two new tools
  - oks2coral to archive OKS database to relational database
  - oks2cool to store configurations in COOL (Athena access)
  - Torbjörn has started looking at these for us
    - First using SQLite which may be most suitable for test rigs
  - We should then look at using cdi (conditions database interface) to store our volatile "IS" data

Overview of Status and Plans

- Present status
  - At present much L1Calo information used online is in the OKS configuration database: HW configuration, cabling, calibration, trigger menu, run types, etc.
  - Exceptions: PPM configuration & calibration in pure XML
  - Also some “flat” data files used by a few applications
- Intentions
  - Trigger menu will move to new CORAL based configuration
  - Calibration data should move to COOL (PPM & others)
    - Also PPM configuration as that is integrated with calibration data
  - Consider moving other L1Calo configuration data to COOL
    - Or are new OKS configuration archiving tools enough?
  - Description of connectivity? (See later)

Calibration Database (1)

- Reminder about COOL
  - Data kept in folders, with unix filename like hierarchy
    - Directories referred to as "foldersets"
  - Folders may contain many rows identified by channel ID
    - Presently only integer IDs, in future also have channel names?
  - History of updates to channel data using interval of validity (IOV) either as period of time or range of run numbers
  - Multiversion folders may have several tagged versions for the same or overlapping IOVs (mainly used offline)
  - API returns sets of rows valid at a particular time (eg now)
  - Implemented in relational DB (using CORAL) but API hides this – though may allow some relational queries in future

Calibration Database (2)

- General proposal for use of COOL folders
  - Follow suggestion in Normans diagram
  - Results for all calibration runs stored in one set of folders
  - Separate validation procedures copy data to another set of folders which are used to configure the system
    - However present use of sequence of multistep runs in analogue signal testing makes it necessary to use the most recent results automatically to run the next stage in the procedure
  - Separate folder for each type of component to be loaded
    - Eg PprChannel, PprMCM, PPM, Serialiser, CpChip, CPM, etc
      - Separate Ppr folders for configuration, timing & energy calibrations
  - Channel IDs coded with crate, module, submodule, channel
    - HW identifier best for online to quickly select data from one crate
    - But might not be so natural for offline?

Calibration Database (3)

- Suggested folder structure
  - Existing convention for top level “foldersets”, eg /TDAQ
  - Next level down free for us, eg “/l1calo”
  - Then type of data “/calibration”, “/configuration”
  - Then level for simultaneously valid types
    - Eg “/physics”, “/cosmics”, “/test”, “/brucesplayground”
    - This follows COOL API philosophy rather than pure relational DB style of putting this as a column in a single folder
  - For calibration, subfolders for “/validated” or “/allRuns”

Calibration Database (4)

- Suggested folder structure (continued)
  - And finally folders for different types of data:
    - "/Cmm", "/Cpm", "/Serialiser", "/CpChip", "/Jem", "/JemInput", "/Ppm", "/PprMcmConfig", "/PprMcmEnergy", "/PprMcmTiming", "/PprChanConfig", "/PprChanEnergy", "/PprChanTiming"
  - Separation of PPM folders according to different types of calibration (timing, energy) updating COOL at different frequencies
- Structure of "validated" and "allRuns" folders
  - Propose that these be identical for ease of copying
    - This would be a requirement if we do need to use the "AllRuns" folders in run control as part of sequence of multistep runs

Calibration Database (5)

- Complete calibration loop: so far
  - Richard has code to store CMM calibration results in COOL
    - He has also read back and compared to previous validated data
    - And checked out the available browsing tools (KTIDB, COOL_IO)
- Missing link has been use by run control
  - I have now added code to L1CaloDatabase to read CMM, CPM & JEM calibration data from COOL folders
  - Fills same object structures as used up to now for OKS
    - But would like to encourage move away from these to "facade" methods in DbXXX classes for each module
  - Can choose COOL vs OKS via IGUI
  - Not quite complete and tested even for CMM/CPM/JEM
  - Also need to add filling of the PPM calib/config structures

Calibration Database (6)

- Next Steps
  - Proposed DB “experts week” at CERN in week of 27 Nov
    - Or delay until 11 December?
  - Aim to converge on folder structure
  - Our run controllers should read calibration (and PPM configuration) data from COOL folders written by calibration procedures and validation process
  - Ensure CMM model calibration works in full
- Further steps
  - Tools required: copying across folders, browsing, etc
  - Storing histograms with POOL tokens (in CASTOR?)

Calibration DB: Column Names?

Hardware oriented vs analysis oriented column names

BpTimingNN	ReadOffWinNN
CblTimingNN	n/a
PipeDelay	n/a
TtcrxPhase1	AverageBestPosition
TtcrxPhase2	n/a
DaqOffsetXX	n/a
RoiOffset	n/a
n/a	ModuleIdB
n/a	RootLocation
n/a	DataStatus

Configuration Database

- PPM configuration
  - Part of integration object structure, so moves to COOL
  - Still have PpmWatch as tool to edit it
- Other module configuration data
  - Do the same thing - and provide editing tools? Timescale?
  - Or leave in OKS (which has some editing tools) and rely on the oks2cool and oks2coral as archive?
- Hardware configuration
  - General TDAQ plan is to derive this from TC installation databases, possibly in connection with PartitionMaker tool
  - Not something we have yet looked at
    - Still need simple editable HW configuration for test rigs

Connectivity

- Present L1Calo status
  - Digital cabling described in OKS
  - Norman started description of analogue cabling including pin to pin patch panel interconnections in relational DB
    - Source is Excel spreadsheets used in cabling document
    - Not yet used in online or offline code
- ATLAS and TDAQ status and plans
  - Assume TC cabling database as master
    - Problem with limited access for updates is known...
  - Extract DAQ cabling into OKS (L1Calo-like schema)
  - Other detectors (eg muons) extracting TC database into relational tables for use online and offline
    - Normans DB would need Cabling and MTF/Rack Wizard?

AOB

- ERS and Exceptions
  - Now being used in new DB code
  - l1calo::Exception base class to be caught
    - Sets of subclasses defined in different packages db/coolL1Calo
  - Gradually use this more
    - Also for info/debug message streams rather than cout/cerr?
- VME Byteswapping
  - Change to hardware byteswapping finally??