Level 1 Calorimeter Trigger Database

Murrough Landon – 3 February 2003

for the Level 1 Calorimeter Trigger group

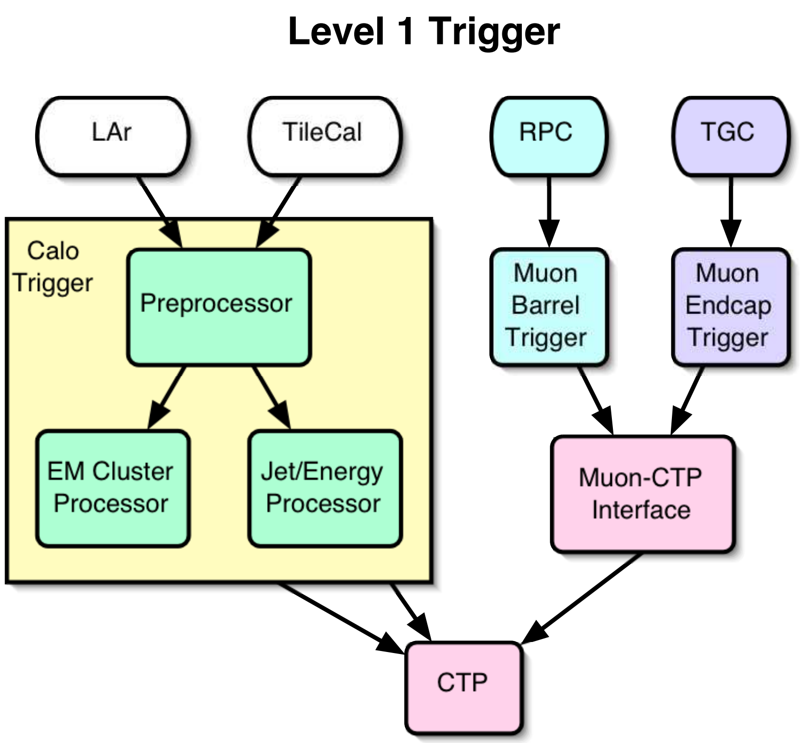
Overview

- Introduction
- Common Level 1 Requirements
- L1Calo: Configuration Data
- L1Calo: Conditions Data
- Archiving, access
- Summary

Introduction

Level 1 Trigger

- "Level 1" comprises the calorimeter trigger, the barrel (RPC) and endcap (TGC) muons triggers, the central trigger processor (CTP) the Muon CTP interface
- The database requirements of different "Level 1" subprojects will vary, though there are also many common features.
- This talk concentrates on the Level 1 calorimeter trigger



Level 1 Requirements (1)

Summary

- Some general requirements across the whole of level 1 were summarised last year in the two main areas
- Configuration data: *functionality required from the database service to configure the system for any given use*
- Conditions data: *all information that is to be kept in order to remember the configuration, behavior and performance of the system*

`http://cern.ch/Atlas/GROUPS/DAQTRIG/LEVEL1/software/level1_databases.html`

Level 1 Requirements (2)

Configuration Data

- Hierarchical schema reflecting the physical architecture
- Several selectable (sub)configurations/calibrations exist in parallel. Especially applies to the pool of trigger menus
- Versions of parallel configurations need to be archived and the last few must be easily accessible
- Resource management is applied when loading the current configuration
- Parameters flagged as modifiable (or not) after being submitted. The parameters can be either overwritten or a new configuration is generated
- Consistency of a configuration checked before it becomes active

Level 1 Requirements (3)

Access and Availability

- Acccess time is an issue especially at run start... Total configuration data is a few tens of Mb, but state transition times must be only a few seconds
- Action of loading a configuration triggers an update of the conditions DB
- Calibration data may be created in ROD crates, event filter or offline. It must be possible to store (and access) new configurations from all those sources
- Both configuration and conditions database must be accessible all the time (24/7) also outside running periods
- Access control needs to be flexible: only experts to change the configuration while normal users should be able to take new configurations

Level 1 Requirements (4)

Conditions Data

- Correlation with other databases (timestamp, run number)
- All aspects of the online configuration setup including calibration and trigger menu used
- Detector geometry including cabling, dead/hot channels
- Production data including history, when used, repairs
- Log of activity on the system (eg Online book keeper)
- Monitoring data: histograms (subset), DCS data, error conditions, run statistics, beam and machine conditions

Introduction

- Recent and current focus is on hardware testing
- We have been using and extending the present Online configuration database
- We are only starting to think about future conditions database requirements

L1Calo: Configuration Data (1)

Hardware Configuration

- Crates, modules and their subcomponents (down to detailed level)
- Cables from detector and between trigger modules
- Mappings of trigger towers to detector channels
- Total size $\sim$1 Mb, changes infrequently?
- One variant (TDAQ partition) for standard ATLAS running plus calibration and test variants, versions stored by run number
- Q: connection between online configuration and offline detector description databases?

L1Calo: Configuration Data (2)

Firmware

- Collections of firmware programs used by modules
- There will be many simultaneously valid variants for a given type of module, especially in the CTP where a pool of available trigger menu loads is required
- Database may hold pointers to binaries which are stored separately
- But archiving (and retrieval) is required with intervals of validity (probably by run number)
- Total size 10-50 Mb for L1Calo, changes infrequently (maybe frequently in the CTP)

L1Calo: Configuration Data (3)

Calibration

- Energy calibration: 8 Mb, changes daily?
- Pulse shape (BCID) calibration: 50 kb, changes occasionally?
- Timing calibration: 50 kb, changes rarely
- Dead and hot channel map: $<<$1 kb hopefully!
- All calibrations, and variants for tests, have version history probably by run number (which is when they are loaded)
- Calibrations may be updated either by online processes between normal runs or by offline processes at any time
- Dead/hot channel maps might get a burst of updates in a short period if a channel is “flaky” and seems to die and come back to life frequently, so versions should be time stamped

L1Calo: Configuration Data (4)

Miscellaneous

- Various other parameters to be loaded into the hardware to configure it into the correct mode for data taking: 1 kb, changes rarely?
- Test vector files: data used to check the trigger hardware. Some generated on the fly according to prescriptions, some physics data needs to be stored in bulk (many Gb?)

L1Calo: Conditions Data

What to store?

- The complete configuration used for each run
- Results of hardware monitoring (statistics, histograms)
- Results of trigger rates monitoring (statistics, histograms)
- Results of trigger decisions monitoring (statistics, histograms)
- Summary of monitoring data from DCS?

When and how?

- Monitoring data logged during run may need to be keyed by run number and event or time stamp within the run

Access Requirements (1)

Availability, completeness, test setups

- In ATLAS, fast 24/7 access to latest version of any database object
- Also desirable to have fast access to the previous version (however old) and all recent versions (last few days)
- Access to latest versions of all database objects needed at run control state transitions
- Where variants exist (eg firmware loads) the conditions DB must contain the details of all variants and record which variant was used for a given run
- Also need standalone implementation (local files, a la OKS?) for test setups at home labs, ie not requiring network access to remote database servers (but with the same API). May want local snapshot of part of a larger database. NB remote test rigs suffer from software decay, using old versions, etc

Access Requirements (2)

Update times and responsiveness

- Online run controllers: read $\sim$20 Mb at state transitions, new calibrations created offline only used at this point whenever they become available
- Online system: calibrations created online should be available for the next run start (may be immediately after the calibration run ends, ie seconds)
- Online monitoring (in ROD/trigger crates): update hot channel map at any time during a run. Hot channels should be suppressed by online “error handling” but other monitoring (EF, semi-offline) may want to know rapidly
- EF/offline monitoring amd simulation needs access to the calibration and other conditions used for a run to be available by the time they start processing that run. For the EF that could be very soon after run start, ie seconds

Archiving

Archiving vs "Conditions Database"?

- In both cases we want to store versions of chunk of data keyed by name with intervals of validity and perhaps arbitrary tags
- Data in Conditions DB expected to be used by offline reconstruction and analysis whereas archived data will not be used offline (probably)
- Online processes may like to see a similar interface? (with different implementations perhaps)
- Offline is clearly interested in calibrations, trigger menus and versions of firmware used, dead and hot channels, cablings (and miscablings), some aspects of the hardware configuration
- Offline is probably not interested in all hardware configuration details, run controllers, software aspects of the TDAQ partition, firmware binaries, etc
- But online software would like to see the complete current configuration as a single seamless whole

Miscellaneous

Random thoughts

- How does the Conditions DB treat the conditions for a sequence of runs or a run with multiple steps (pause/resume or checkpoints) which are part of a calibration scanning some parameter?

Online “Confdb” Wishlist

Already announced

- Read/write access to the configuration database, automatic generation of DALs, support for cabling and firmware

Desirable

- More detailed description of modules, integration with calibration data for modules. At the moment L1Calo has separate (old) module subcomponent description. Endcap muon trigger (TGC) have a specific solution. Are there common requirements across other detectors?

Effort...

Very limited!

- The L1Calo database effort (which is $<<$ 0.5 of me) is likely to be saturated with implementing common solutions to meet L1Calo specific needs
- We cannot offer any effort towards common database work in the near or medium term

Summary

- Online configuration should present a single unified view of the hardware (down to a detailed level), firmware and software setup together with the calibration, trigger menu and other data needed to configure the whole system
- The complete configuration used for each run should be recorded and be available to be restored
- Part of the complete configuration can be identified as of interest for offline processing (conditions database) while the rest may be archived in some fashion inaccessible to offline - but this division should not be apparent to online processes
- Home labs and other test setups need small local databases without empires of networks and servers
- Fast access to new online calibrations and new hot channel maps by EF monitoring