CHEP'07 VICTORIA, BC
International Conference on Computing in High Energy and Nuclear Physics
2-7 Sept 2007 Victoria BC Canada

Data acquisition software for the CMS strip tracker

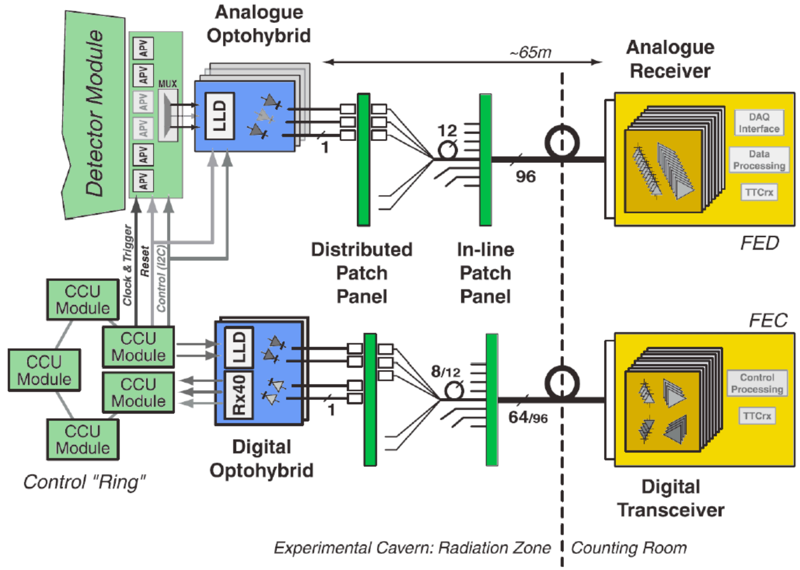
R Bainbridge[1], G Beaulieu[2], S Bel[3], J Cole[4], N Cripps[1], C Delaere[3], F Drouhin[5], J Fulcher[1], A Giassi[6], D Giordano[7], L Gross[8], K Hahn[9], S Mersi[10], L Mirabito[11], M Nikolic[12], S Tkaczyk[13] and M Wingham[1]

[1] Blackett Laboratory, Imperial College London, UK
[2] Institut de Physique Nucleaire de Lyon, France
[3] CERN, Geneva, Switzerland
[4] Rutherford Appleton Laboratory, UK
[5] Universite de Haute Alsace, France
[6] INFN, Sezione di Pisa, Italy
[7] INFN & Università di Bari, Italy
[8] Institut Pluridisciplinaire Hubert Curien, CNRS, France
[9] Massachusetts Institute of Technology, USA
[10] INFN & Università di Firenze, Italy
[11] Institut de Physique Nucleaire de Lyon, France
[12] University of California, USA
[13] Fermi National Accelerator Laboratory, USA

E-mail: Robert.Bainbridge@cern.ch, Laurent.Mirabito@cern.ch

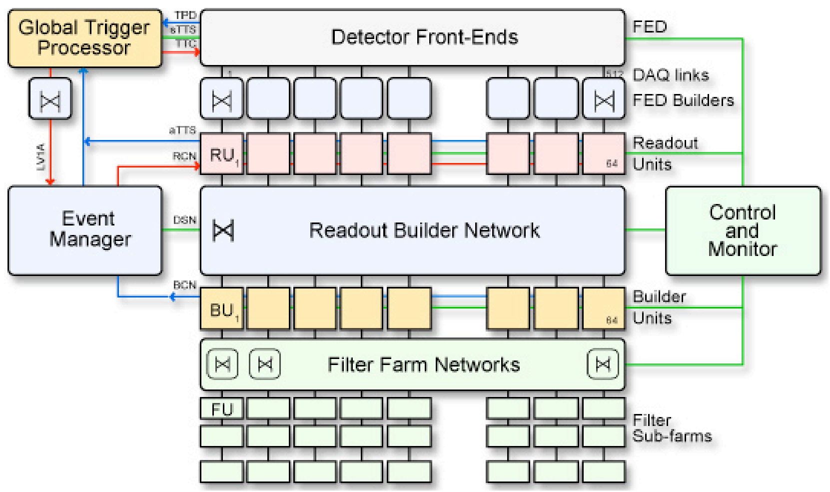
Strip tracker data acquisition system

- Strip tracker has unprecedented size and complexity
  - Active area of silicon is >200 m^2
  - 10M readout channels
  - 15k front-end modules



- Control system
  - Distributes clock, L1 triggers and slow control
  - 40 off-detector Front-End Controllers (FECs)
  - 300 "control rings" with redundancy

- Readout system
  - 15k micro-strip sensors with various topologies
  - 76k analogue APV25 front-end chip (0.25 um)
  - 38k analogue optical links
  - 440 off-detector Front-End Drivers (FEDs)
    - ADC, zero-suppression and formatting
    - S-Link readout (global) and VME readout (local)

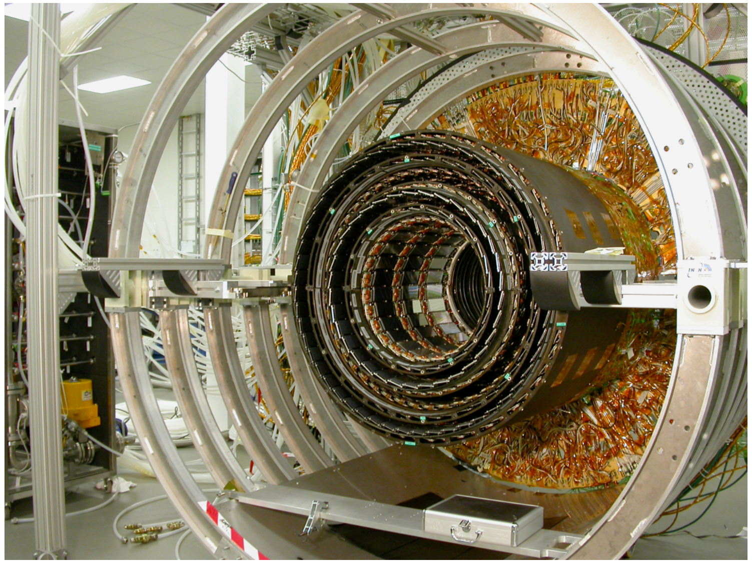


- CMS Event Builder
  - Receives data from 700 FEDs via S-Link
  - Two-stage event building, 100Gbit/s throughput
  - DAQ comprises 8 "slices" (system scaled up over time)
  - Filter Farm provides High Level Trigger and monitoring

Recent “slice tests” with the strip tracker

- Magnet Test Cosmic Challenge (Autumn 06)
  - Commission and operate all sub-detectors within 4T field
  - Commission trigger system (using cosmic rays)
  - Data taking using global DAQ
  - Cosmic ray reconstruction within offline software fwk (See: D. Kcira, abstract 301)
  - Some milestones achieved for strip tracker DAQ:
    - Integration and operation of Tk DAQ with all of CMS
    - Commissioned system of 100k channels
    - Collected several million events using cosmic trigger

- Tracker Integration Facility (Feb 07 - July 07)
  - Large clean room environment for final integration
  - Infrastructure and services to operate **1.5M channels**
  - Large cooling capacity for “cold tests” up to -15 ^{o}C
  - Commissioning of sub-detector prior to P5 installation
  - Exercise aspects of service infrastructure (power, cooling)
  - Tracker Analysis Centre (on Meyrin site)
    - Operation of DAQ and Detector Control System
    - Online monitoring and offline analysis
    - Promote collaboration b/w online/offline communities
    - TAC will be maintained during LHC running



CMS software frameworks

- XDAQ is the online software framework for the CMS DAQ and provides:
  - A fast communication protocol for peer-to-peer messaging between registered processes
  - A slower communication protocol for configuration of registered processes
  - A finite-state machine schema, event building and memory management tools

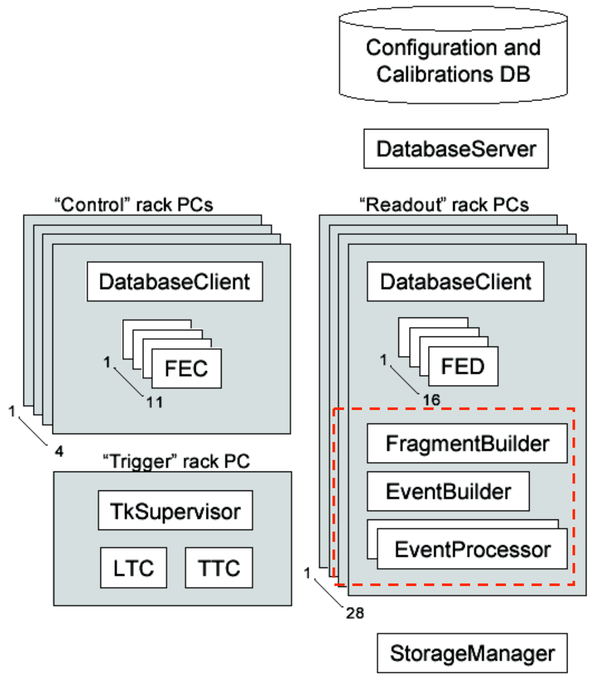
- CMSSW is the offline software framework and provides:
  - Simulation, data handling, reconstruction, analysis tools and event selection (HLT) software
  - Services and tools such as: geometry, conditions database, monitoring framework (DQM)

- Strip tracker DAQ software uses both XDAQ and CMSSW frameworks
  - Hardware-related applications (control and configuration) developed within the XDAQ framework
  - Data handling and commissioning analysis components have been developed within CMSSW

- Motivation for DAQ software to use CMSSW (in addition to XDAQ)
  - CMSSW provides many useful services and tools (which will be supported for lifetime of the expt)
  - Allows for scalable design, through parallel event processing and distributed monitoring
  - CMSSW-based commissioning analysis can be used in both local and global data-taking modes
    - Track-based calibration procedures will use global mode
    - VME readout (~5 Hz) versus S-Link readout (4 kHz)
    - Local computing resources (tens of rack PCs) versus global computing farm (1000 CPUs)

Data acquisition software

- DAQ software comprises tracker-specific XDAQ applications
  - Allow configuration and control of hardware components
  - Top-level “tracker supervisor” steers data acquisition



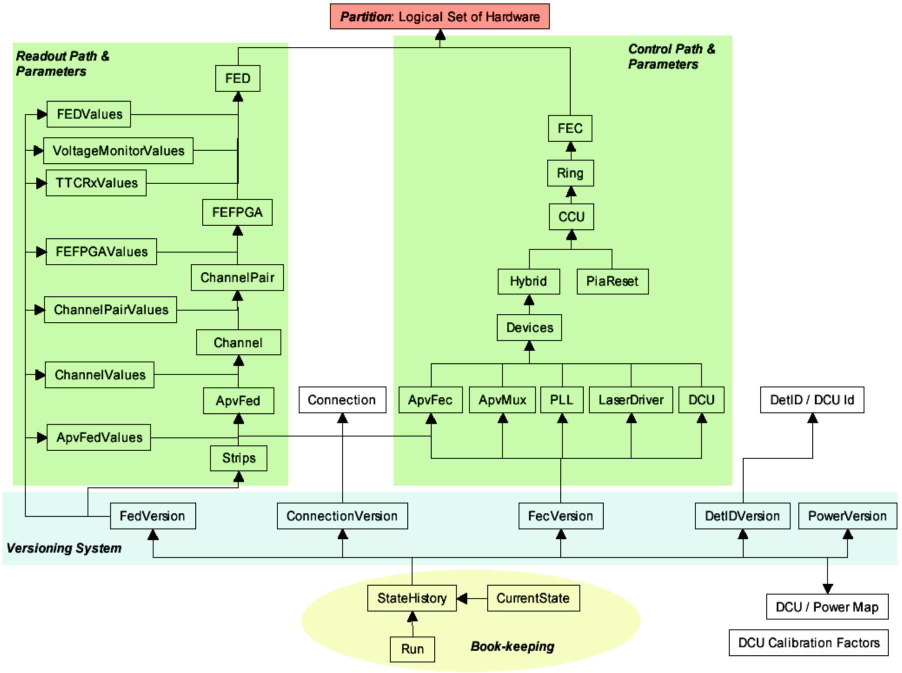
- DAQ services and tools
  - Configuration database (server-client “caching” mechanism)
  - Error diagnostic software (handles error states, problem solving)
  - Detector Control System (power supplies, environmental info…)
  - CMS Event Builder tools provided by XDAQ fwk and EvF group
  - CMS Run Control (defining tracker-specific state machine)
  - CMS Storage Manager (provides mechanism to write to disk)

- Offline software framework (CMSSW) used in online environment
  - EventProcessor app provides “hook” b/w XDAQ and CMSSW
  - EventBuilder forwards complete events to CMSSW via EventProcessor
  - CMSSW provides commissioning analysis, track reconstruction…

- “Local DAQ” is used during detector commissioning and calibration
  - VME readout, local event building on rack PCs via Gbit ethernet
  - Plan to re-use same architecture and tools for *monitoring* during physics

Configuration database

- Online configuration and calibration database
  - Holds hardware configurations and calibrations
  - Organized according to (trigger) partitions
  - Versioning system provided for all tables
  - Complete history recorded in “run table”
  - Data volumes:
    - Control components: up to 1.7 MB
    - Readout components: up to 66 MB
  - Presently optimizing download performances
  - Developing client-server “cache” mechanism

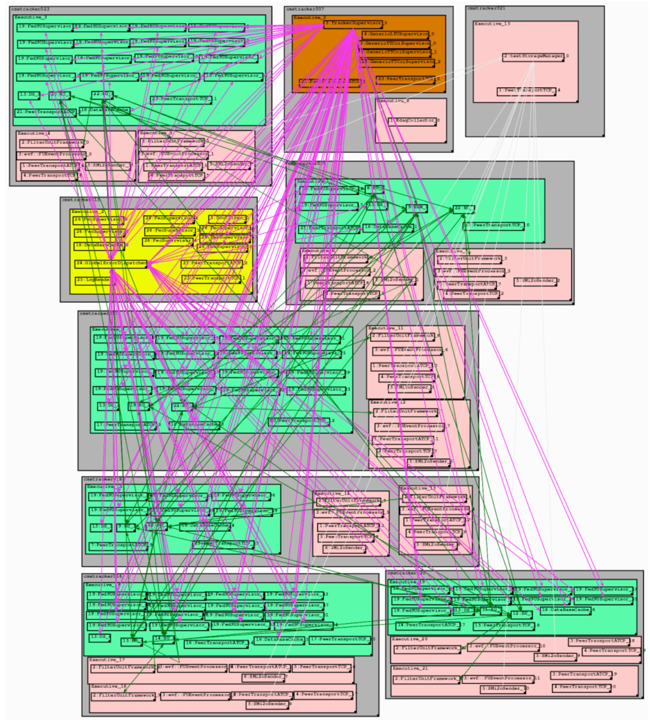


- Database model
  - Oracle relational database
  - C++ API wraps underlying SQL queries
  - Data formatted according to XML schema
  - “Static” table used to hold geometry information

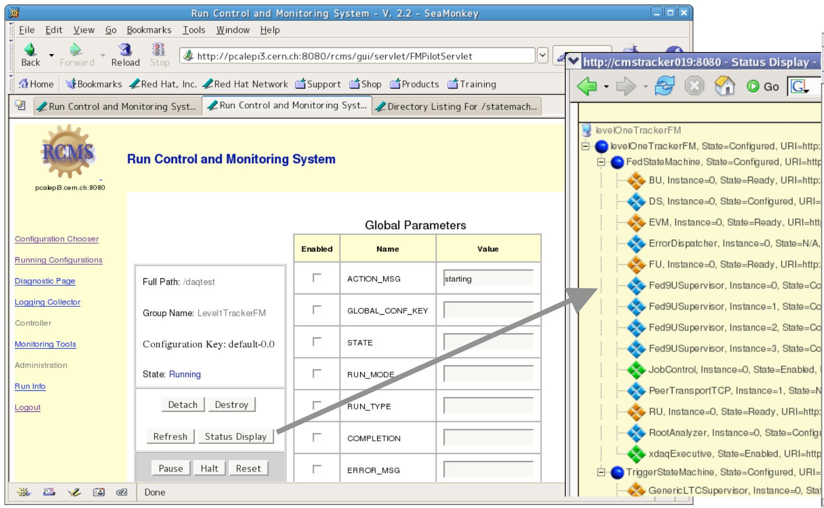
- Online-to-Offline (O2O) transfer tool
  - Subset of configurations/calibrations are also *conditions* data used by offline reconstruction (eg, noise)
  - O2O tools moves this data subset to CMS conditions DB, which is fully integrated with CMSSW fwk
  - “Run table” used to define “intervals of validity” in order to synchronize conditions with event data

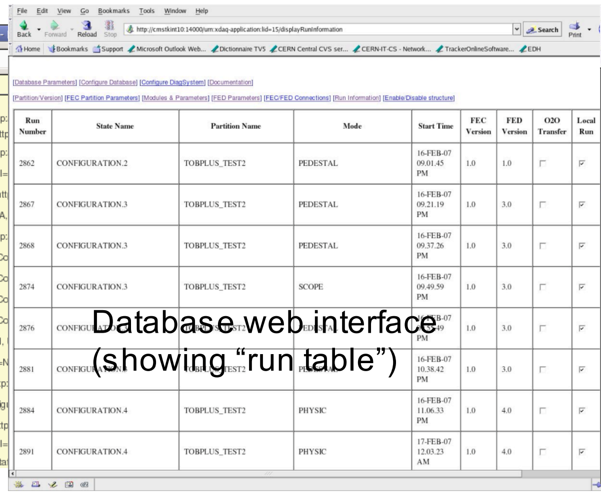
Run control, monitoring and configuration

- CMS Run Control and Monitoring System (RCMS)
  – Allows definition of tracker-specific “function manager”
  – Controls FSM of all XDAQ apps and monitors state transitions
  – Web interface displaying run information and possible states
  – RCMS was standard user interface during TIF activities

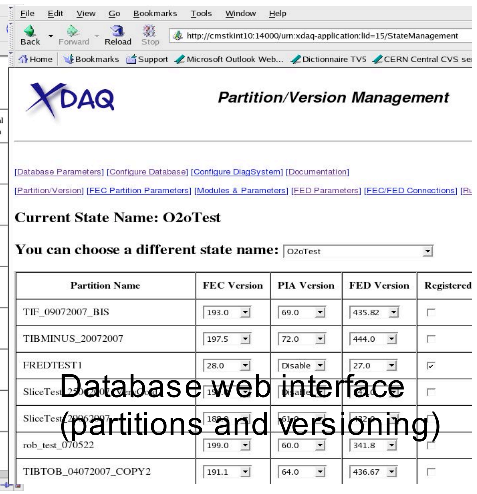


- Building DAQ configurations
  – DAQ configurations built using simple graphical tool: DuckCAD
  – DuckCAD provides editors to define all aspects of configuration
    – Hosts, networks, applications, function managers…
  – Configurations stored in RCMS DB and loaded via web interface





Database web interface (showing “run table”)

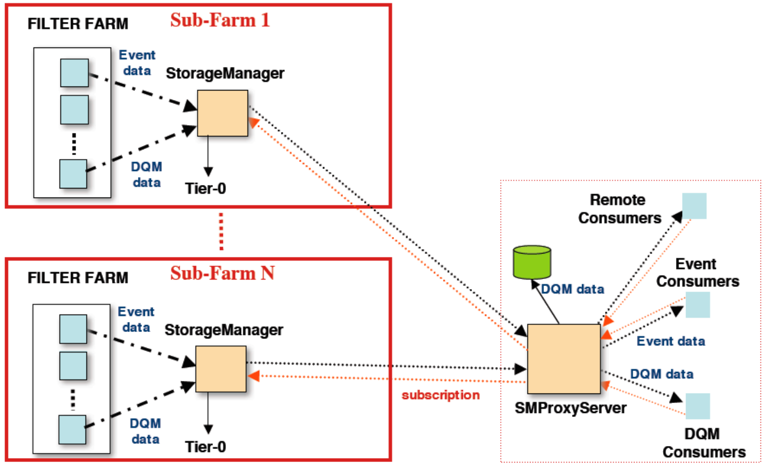


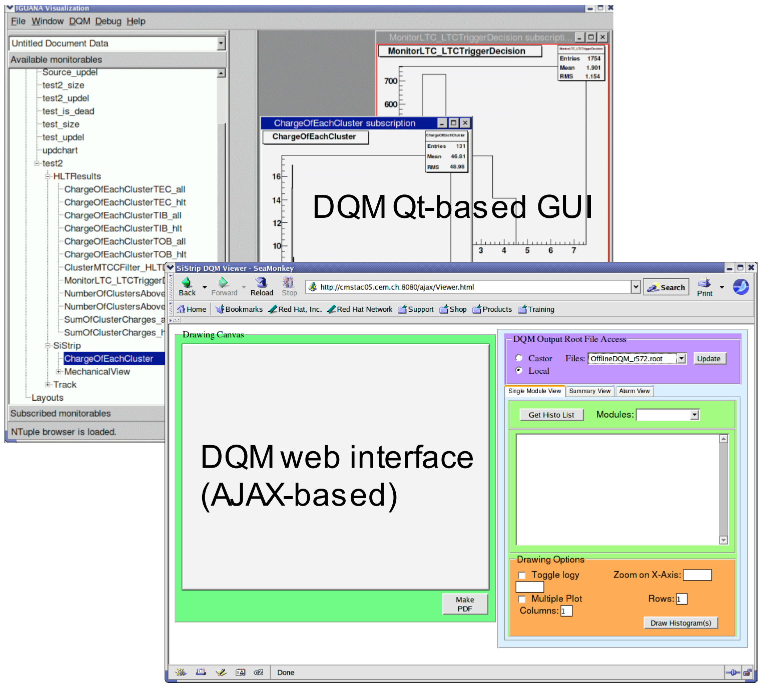
Database web interface (partitions and versioning)

Data storage and online monitoring

- Data storage using the CMS Storage Manager
  - Provides mechanism for writing-to-disk
  - Outputs "intermediate" binary format (for performances)
    - Conversion to EDM format done offline
  - Used for first time during TIF activities
  - Also acts as "histogram server" for online DQM monitoring

- Online monitoring
  - Several streams, using both XDAQ and CMSSW fwks
  - XDAQ-based monitoring primarily concerns HW scalars
  - CMSSW "DQM" monitoring based on event data
  - Online monitoring performed at TIF (& Fermilab ROC)
  - Qt-based GUI and AJAX-based interfaces for DQM





Refs: L. Sexton-Kennedy, abstract 367 (Storage manager)
N. De Filippis, abstract 237 (Data flow)
S. Mersi, abstract 342 (Hardware monitoring)
D. Giordano, abstract 434 (DQM monitoring)

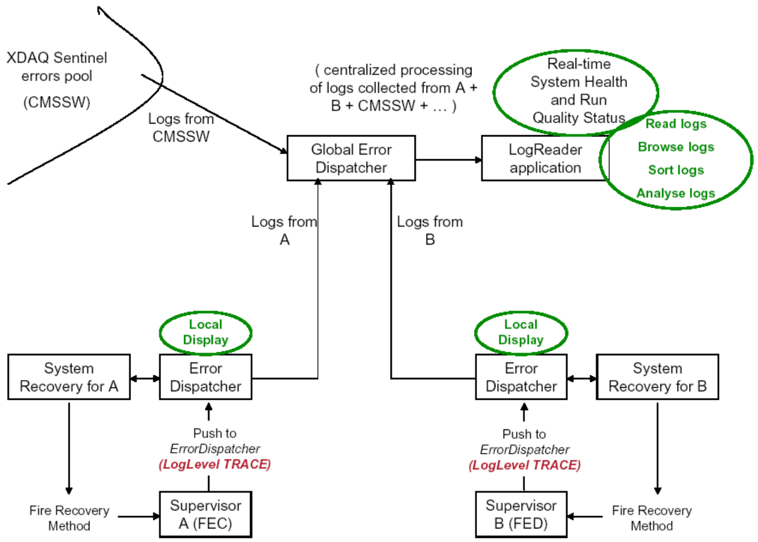
Error diagnostic system

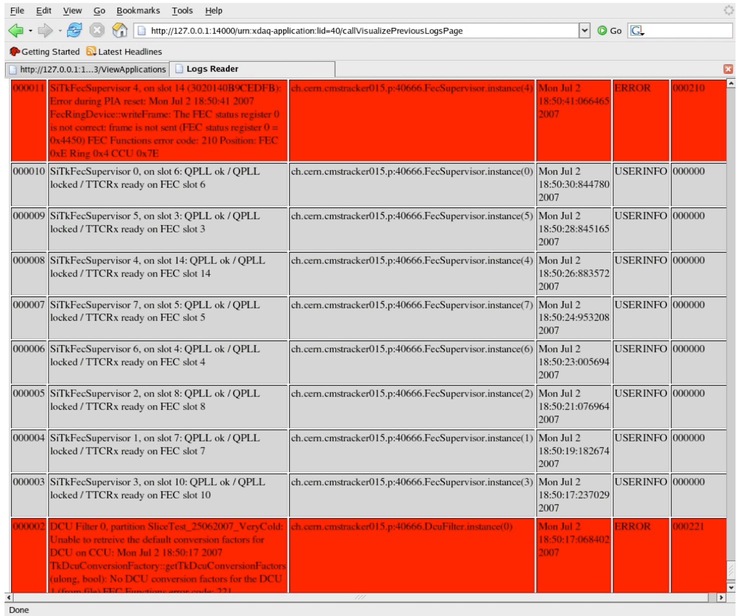
- Overview
  - Complex sub-detector, many potential error sources
  - Handles both configuration and runtime errors
  - Provides human-readable description (what, where, why)
  - Will be interfaced to dedicated error conditions database
  - Mid-term goal: integrate with CMSSW and DCS
  - Long-term goal: error analysis and automatic recovery

- Design and architecture
  - Developed within XDAQ framework
  - DAQ applications communicate with “error dispatchers”
  - Error dispatchers collect and process error messages
    - Hierarchical “tree” arrangement: flexible and scalable
    - Tree typically matches logical structure of HW
  - I/O: other dispatchers, LogReader, file, errors DB, Sentinel, Chainsaw, XMAS…

- Slice tests
  - Integrated and tested with DAQ software at the TIF
  - Stress tests using large-scale (10% of final) system
  - Performance tuning required (pre-processing of errors)

See: S. Mersi, abstract 342





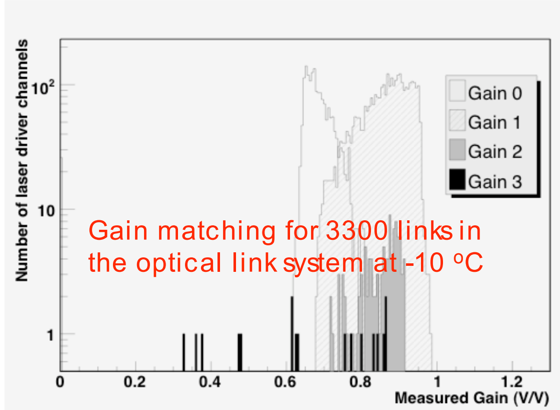
000011	SiTkFecSupervisor 4, on slot 14 (3020140B9CEDFB): Error during PIA reset: Mon Jul 2 18:50:41 2007 FecRingDevice::writeFrame: The FEC status register 0 is not correct: frame is not sent (FEC status register 0 = 0x4450) FEC Functions error code: 210 Position: FEC 0xE Ring 0x4 CCU 0x7E	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(4)	Mon Jul 2 18:50:41:066465 2007	ERROR	000210
000010	SiTkFecSupervisor 0, on slot 6: QPLL ok / QPLL locked / TTCRx ready on FEC slot 6	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(0)	Mon Jul 2 18:50:30:844780 2007	USERINFO	000000
000009	SiTkFecSupervisor 5, on slot 3: QPLL ok / QPLL locked / TTCRx ready on FEC slot 3	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(5)	Mon Jul 2 18:50:28:845165 2007	USERINFO	000000
000008	SiTkFecSupervisor 4, on slot 14: QPLL ok / QPLL locked / TTCRx ready on FEC slot 14	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(4)	Mon Jul 2 18:50:26:883572 2007	USERINFO	000000
000007	SiTkFecSupervisor 7, on slot 5: QPLL ok / QPLL locked / TTCRx ready on FEC slot 5	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(7)	Mon Jul 2 18:50:24:953208 2007	USERINFO	000000
000006	SiTkFecSupervisor 6, on slot 4: QPLL ok / QPLL locked / TTCRx ready on FEC slot 4	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(6)	Mon Jul 2 18:50:23:005694 2007	USERINFO	000000
000005	SiTkFecSupervisor 2, on slot 8: QPLL ok / QPLL locked / TTCRx ready on FEC slot 8	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(2)	Mon Jul 2 18:50:21:076964 2007	USERINFO	000000
000004	SiTkFecSupervisor 1, on slot 7: QPLL ok / QPLL locked / TTCRx ready on FEC slot 7	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(1)	Mon Jul 2 18:50:19:182674 2007	USERINFO	000000
000003	SiTkFecSupervisor 3, on slot 10: QPLL ok / QPLL locked / TTCRx ready on FEC slot 10	ch.cern.cmstracker015.p:40666.FecSupervisor.instance(3)	Mon Jul 2 18:50:17:237029 2007	USERINFO	000000
000002	DCU Filter 0, partition SliceTest_25062007_VeryCold: Unable to retreive the default conversion factors for DCU on CCU: Mon Jul 2 18:50:17 2007 TkDcuConversionFactory::getTkDcuConversionFactors (ulong, bool): No DCU conversion factors for the DCU	ch.cern.cmstracker015.p:40666.DcuFilter.instance(0)	Mon Jul 2 18:50:17:068402 2007	ERROR	000221

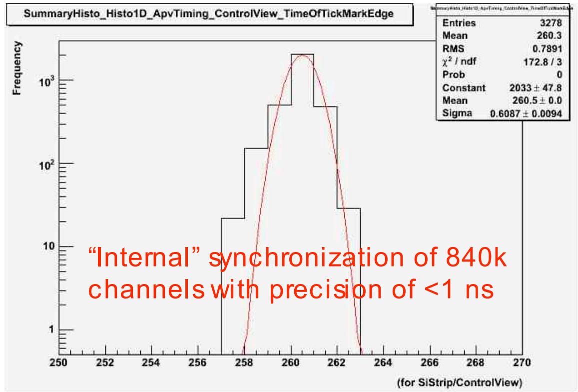
Detector commissioning procedures

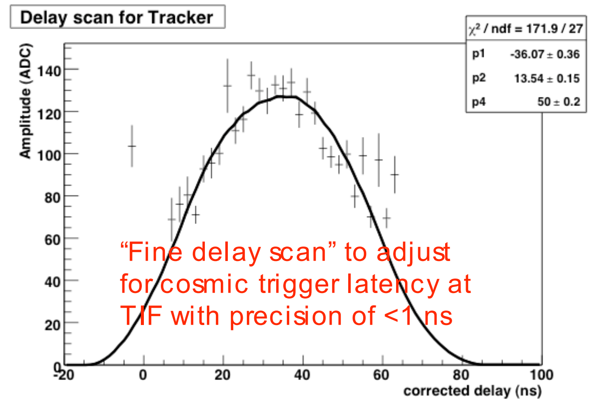
- Commissioning procedures steered by strip tracker DAQ
  - Dedicated DAQ "loops" to reconstruct calibration pulses, timing delay curves and other features of the APV25 data stream
  - Determine configuration parameters and calibration constants…
  - …which are uploaded to strip tracker configuration database
    - Eg, gains, noise, timing delays…
  - Heavily used during Start-Up to validate detector performances

- Commissioning procedures
  - Detect hardware components of the four trigger partitions
  - Determine connectivity b/w front-end, power supplies and FEDs
  - Tuning of all front-end APV25 chips (bias, gain)
  - Gain matching and calibration of readout optical links (38k fibres)
  - Determine calibration constants for FED zero-suppression algorithm
  - "Internal" synchronization of front-end (with ns precision)
  - HV scans to determine depletion voltages (req. interfacing to DCS)
  - Track-based analyses:
    - Tuning and calibration of analogue pulse shape of APV25
    - "Fine delay scan" to synchronize to LHC collisions

See: R. Bainbridge, CHEP 2006, abstract 311

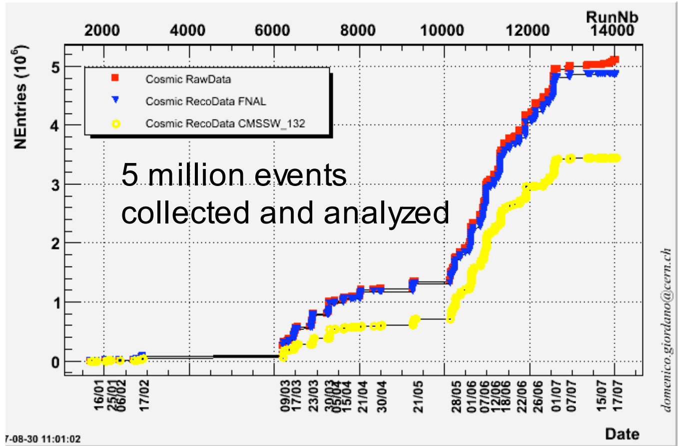




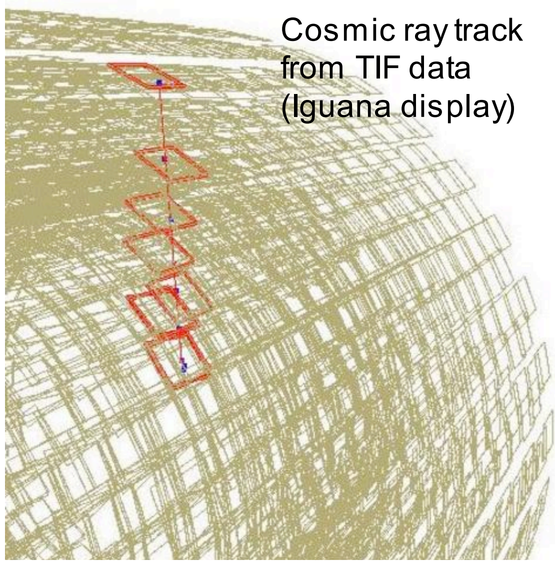


Slice test experiences and milestones

- Integration (and validation) of several new services
  - CMS Run Control, Event Builder, Storage Manager
  - Configuration and calibrations database
  - Error diagnostic system
  - DAQ integrated with Detector Control System and its DB
  - Closer coupling with offline analysis (O2O, monitoring)



- Validation of final software architecture
  - Scalable design (performances ~independent of channel count)
  - Memory and CPU usage easily handled by standard rack PC
  - Trigger rates of few Hz in "local DAQ" mode (VME readout)
    - Event size for largest system was 3.3 MB
    - VME throughput of ~10 MB/s



Cosmic ray track from TIF data (Iguana display)

- Detector commissioning and operation
  - Commissioned system of **1.6M channels** (~15% of final system)
    - Detecting HW connectivity to measuring trigger latency **in few hrs**
  - Long-term stable operation of DAQ during cosmic runs
    - Running with multiple partitions (inner+, outer+, both end-caps)
  - 5 million events collected on disk (using VME readout)
    - Ongoing studies: detector performance, tracking and alignment
  - 100kHz performance studies using "DAQ column" (S-Link readout)

Lessons learned from the slice tests

- Software build and release management was difficult
  - No central support provided for XDAQ developments, strip tracker learned the hard way
    - In contrast with CMSSW, for which multi-platform builds and releases are handled centrally
  - Parallel activities of code development, release validation (debugging) and DAQ operation at the TIF
    - Rack PCs accessed central installations (for multiple release versions) via NFS
    - Releases managed using CVS to tag “stable”, “beta” and “development” versions
    - Allowed easy switching b/w releases as focus shifted b/w data-taking and DAQ development
  - Now also package the DAQ software as RPMs to aid deployment of local installations
    - This method likely to be used at Point 5, to suppress susceptibility network problems
  - DAQ software releases compiled using the 'distcc' distributed compiler and a 20-PC farm (quick!)
  - XDAQ and CMSSW frameworks are now closely coupled for strip tracker DAQ
    - Shared libraries for database access and FED readout (root, xerces, oracle, XDAQ…)
    - DAQ software therefore sensitive to both online and offline release schedules
    - Critical issue for forthcoming “global runs” and “checkout” period of strip tracker at Point 5

- Outstanding software issues
  - Long initialization of software during “Configure” just prior to start-of-run
    - New database client-server caching system required
  - Integration of Error Diagnostic system and CMSSW remains to be done
  - Building of DAQ configurations very painful (complicated, time-consuming, prone to corruption…)

Conclusions

- Huge progress made in last year, thanks to experiences at TIF and during MTCC
- Large-scale slice tests demonstrate software is stable and architecture is scalable
- DAQ software is ready to commission and operate the strip tracker at Point 5
- Next major challenge: DAQ software is still an expert-only system and this must change!