The Data Quality Monitoring of the CMS Experiment: the Tracker Case

Suchandra Dutta
University and INFN Pisa, Italy

11th Topical Seminar on Innovative Particle and Radiation Detectors (IPRD08)
1-4th October, 2008 Siena, Italy

Suchandra Dutta
University and INFN Pisa, Italy
Outline

- Introduction
- CMS DQM Framework
- DQM Deployment
- Components of DQM
  - Producer (Tracker Example)
  - Quality Test
  - Consumer (Tracker Example)
- Data Certification
- Graphical User Interface
- Summary
Introduction

The Data Quality Monitoring (DQM) is one of the most important tools in an experiment

- Ensures
  - All sub-detectors, DAQ, Trigger components function correctly
  - Useful data recording
  - Very fast detection of problem in the early stage of data acquisition so that actions can be taken to cure it
- Must work efficiently throughout the lifetime of the experiment
### DQM Deployment

- **Online DQM**
  - Real time and uses fraction collected events
  - Monitors detector performance and efficiency

- **Offline DQM (During prompt reconstruction)**
  - Full statistics within a few hours of latency, uses best calibration
  - Detailed monitoring of detector performance and reconstruction algorithms
  - Data certification

-During data taking (Online DQM)
-Within a day/hours (Offline DQM)
-Within a few hours (CAF/control center)
CMS DQM Framework

- DQM Producer:
  - books/fills histograms and makes them available

- DQM Consumer:
  - Accesses histograms from source
  - Performs further analysis

- GUI:
  - Visualisation

The DQM application of all Sub-Detectors, Trigger, Calibration, Alignment, Analysis Objects in CMS based on this framework.
What do we Monitor

The DQM producer defines quantities we monitor

- Consists of histograms, strings, integers, floats
- They booked & then filled with information accessing from event
- In case of Tracker DQM
  - Information from different levels of reconstruction is accessed from the event which are
    - Raw data (readout and unpacking error)
    - Local reconstruction (Digi, Cluster, on/off track clusters)
    - Global track parameters
    - Residuals of hit (Alignment)
  - Histograms are arranged in tree like folder structure
    - Full Detector
      - layers/disks....
      - Detector modules
Quality Tester Tool

Automatic tool that allows the application of statistical tests on histograms to check the quality

- Histograms compared with reference (statistical test)
- Histograms are qualified checking mean, rms, etc. with reference values specified
- Finally alarms generated from test results
  - OK
  - Warning
  - Error
- The alarm levels can be adjusted depending on the need
- Quality tests (QTest) configured and attached to the histograms through xml files

Automatic determination of histogram quality is a complex and difficult task in such huge system of histograms. The QTest tool in DQM allows to configure and use it in a simple way
How Do We Monitor

- DQM consumer accesses histograms created in producer
- Performs analysis to create
  - summary results
  - alarm tree
  - synoptic views of the whole detector
- DQM output contains ALL histograms produced in producer and consumer
  - Some of these histograms are carefully selected by experts and laid out in an easily understandable way; Non-Expert/Shifter view
  - Experts first check the views from different layouts and synoptic views
    - In case of problem digs down through the structure to pinpoint the problem following the hierarchical structure
The Tracker Consumer

- Due to high granularity of detector modules in the Tracker
  - ~15K for SiStrip (~10 million channels)
  - ~1.5K for SiPixel (~67 million channels)

.....we have a large number of Histograms (~ 300K)
  - Not possible to check individually
  - Absolutely needed to spot problems

- Use detector level histograms to create summaries in the consumer
  - Mean/rms values from detector level histograms are used to create summary histogram
  - QTest results used to create overall status at different levels
    - producer, consumer used in the same process in online
    - In Offline producer is used during prompt reconstruction which are split in different processes. At harvesting stage they are merged and QTest, consumer used
Data Certification

- Defines the goodness of data to be used
- Design getting finalised
  - Based on online & offline DQM with overall status defined with histogram quality
  - Input from shifters also being used presently
- Soon will be combined with information from
  - DAQ
  - Detector-Control-System (DCS)
  - Automatic Error report system

Online DQM

- QTest Results
- Shift Histograms

Offline DQM

- Qtest Results
- Shift Histogram

Run Registry

- Detector Performance
- Reconstruction and Physics Performance

Expert Sign-off

Data Base
Graphical User Interface

- The Graphical User Interfaces (GUI) used in CMS DQM application are web based
  - Easily accessible from everywhere and no need for software installation, just a web browser is sufficient
- Asynchronous Javascript and XML (Ajax) is used
  an umbrella term for several mature, open source technologies that promote reusable patterns for web-based development
  - More responsive and natural than classical Web Browsers
  - Native desktop like look-and-feel
  - Well defined responsibilities for web server/client sides
  - Proper use of bandwidth
Ajax Model

- Asynchronous communication does not block the page
- Browser updates the specific part of the page
- Significantly less dead time
Available GUI(s)

- **CMS Web GUI**
  - A generic web based GUI developed for the whole CMS
  - Widely used to visualize DQM histograms
  - Very robust and capable of handing > 300K histograms in ~100 Miliseconds

- **Tracker Expert GUI**
  - First prototype deploying Ajax, developed for Tracker Experts which evolved over time
  - The DQM client is used as Web Server hence direct interaction with the client is the main advantage here
  - Can be used only Online during data taking
CMS Web GUI (Status Summary)

- Status for All CMS sub-detector systems
- Simple 2D map to represent overall status with QTest results
- The color scale:
  - Status : 95-100%
  - Status : 85-95%
  - Status : < 85%
Tracker GUI (Summary View)

Problems located in top level can be traced down to detector level following tree like structure.
Tracker GUI (Alarm View)

Alarms represented in tree like structure to trace back the faulty detector modules

Error
2D map of tracker layers and disks

Each pixel represents a detector module

The map is painted with generated alarm

The map is active (Ajax), histograms of that modules can be viewed
Synoptic View of Tracker (Readout)

- Same alarm information shown in readout view to understand if problems are correlated
- Active map hence histogram drawing is possible
It is very important to study the detector behavior in medium (few days) to long term (few weeks or months)

- Summary information from DQM Histograms are stored in database corresponding to each run number
- They can be plotted later as a function of run number at any given interval
Summary

- CMS DQM framework and tools are being successfully deployed in online and offline.
- All sub-detector groups have developed DQM producers and consumers based on the framework.
  - Extremely useful Quality Test tool enables automatic qualification of histograms.
- Powerful web based GUI in place and being used widely.
  - There is a tracker specific GUI for experts.
- Data certification tool is being developed.

DQM is an indispensable part of an experiment. In CMS it is in place & getting more and more matured with Cosmic Data. Taking exercises to be perfectly ready for collision!
Back Up Slides
Traditional Web Model

- Synchronous communication
- Browser does a full refresh as the server sends the updated HTML
- Long dead time
Tracker Web GUI

1. Web Interface (Action)

2. Function call (Javascript)

3. HTTP transport

4. Web Server

5. XML/Image data

6. Response

7. DHTML/CSS

- Ajax engine:
  - XMLHttpRequest
  - XML
  - Document object model (DOM)

- Asynchronous update is the key here

Request Interpreter

Information Extractor (back end)