DQM4HEP
Status and prospects.

CHEF 2017 - Lyon

R. Ete, A. Pingault, T. Coates
DESY
October 5, 2017
Summary

• Introduction
• Framework presentation
• Experiments running with DQM4HEP
• EUDAQ / DQM4HEP interface
• Current status
• Ongoing and future work
DQM systems in HEP domain:

- Evaluate data quality and alert users of anomalies
  - Are the distribution what we expect?
  - Comparison between runs or old/new software version
  - Quick feedback from hundred of plots is challenging
- Provide online and offline analysis
  - Automated data quality tests, possibly with reference histograms
  - Distributed system for online analysis (data collectors)
  - Dedicated visualization interface (Qt, Web)
- Already developed for most of HEP experiments (i.e. AMORE or CMSSW)

But ... Based on experiment specific event format

- Not re-usable by other experiments
- Duplicated software
- Ad-hoc solution for test-beam setup monitoring

Development of a generic DQM software for any HEP experiment: DQM4HEP
Key points:

- **Standalone plugin system**
  - Plugin = C++ class in a shared library
  - Load shared library at runtime and hook plugin class

- **Generic event data model/format.** User needs to define:
  - Event model
  - Conversion Model ↔ Binary

More general features:

- Online analysis (API)
- Distributed system (TCP/IP)
- Data collectors: event and histogram collector servers
- Quality test tools: interface + quality test templates
- Visualization interface (histograms and quality tests)
Monitor element

- Wrap a ROOT TObject
- Optionally hold a ROOT TObject as reference

Quality test

- Implement the logic to test a monitor element
- Output a quality report (quality flag, success, etc)

One monitor element can be tested with many QTests, e.g:

- Kolmogorov test using a reference histogram
- Mean of histogram within an expected value

One QTest can be attached to many monitor elements, e.g:

- Test different histograms with the same gaussian distribution
DQM4HEP

Online architecture

**DATA FLOW**

- **DAQ**
  - Send event (TCP/IP)
  - Send/query event (TCP/IP)
  - External data source (e.g. Slow control)

- **Standalone modules**
  - Send event (TCP/IP)
  - Send/query event (TCP/IP)

- **Event collector servers**
  - Start/End of run (TCP/IP)

- **Analysis modules**
  - Send monitor elements (TCP/IP)

- **Monitor element collector servers**
  - Start/End of run (TCP/IP)

- **Run Control server**
  - Start/End of run (TCP/IP)

- **Remote process management**
  - (TCP/IP)

- **Job control servers**
  - (TCP/IP)
  - [START,STOP]
  - [STATUS]

- **Job control GUI**
  - [QUERY]
  - [UPDATE]

- **Monitoring GUIs**
  - [QUERY]
  - [UPDATE]

- **Tasks**
  - Developer
  - Engineer
  - Shifter
  - DQM - Internal

---

**R. Ete — DESY — October 5, 2017 — Page 6**
Analysis module

- Receive and process event (e.g. from DAQ)
- Book and fill histograms
- Process quality tests
- Send histogram and QReports to collectors with cycle structure
  - Every N events/seconds
  - User can reset histogram if needed at end of cycle

Standalone module

- Receive and process data from external source (e.g. slow control)
- Book and fill histograms
- Process quality tests
- Send histogram and QReports to collectors every N seconds
Start/stop/manage many processes on many hosts
DQM4HEP
Online monitoring interface (Qt Gui)

- Histograms organized in tree structure
- Plot many histograms at the same time
- Receive real time updates
- Browse histograms on collectors
- Histograms organized in tree structure
- Plot many histograms at the same time
- Receive real time updates
- Browse histograms on collectors
DQM4HEP used by different detectors in the CALICE collaboration.

SDHCAL online monitoring
- Hit maps
- Electronics rate
- Slow control: I, HV, LW, T, P
- GRPC efficiency, multiplicity

AHCAL online monitoring
- Hit maps
- Correlation with Telescope hits
- Electronics rate
DQM4HEP adopted as monitoring framework by AIDA2020 WP5:

Task 5.4 Development of data quality and slow control monitoring

Binding between the EUDAQ framework and DQM4HEP is ongoing.
ILD collaboration entering in a new MC production process.

Need for automatic data quality checks for simulated/reconstructed quantities.

Ongoing work to separate the main package (DQMC) into two different software

**dqm4hep-core**

- MonitorElement (ROOT)
- Quality test
- Event interface
- Streaming (xdrstream)
- Plugin management
- DB tools (MySQL)
- Logging (spdlog)

**dqm4hep-online**

- Modules (User classes, Online API)
- Event collector (server and client)
- Monitor element collector (server and client)
- Run control (server, client and external interface)
Current effort to provide an important set of quality test templates in core library

Users can also implement their own quality test(s)

- **Kolmogorov test** \((\text{hist} + \text{ref})\)
  - Mean withing range
  - Mean 90 within range
  - No data after limit
  - No data before limit
- **Fit function and check** \(\chi^2\)
  - Likelihood fit
  - Fraction of data after limit exceed
  - Fraction of data before limit exceed
  - RMS lower than
  - RMS 90 lower than
- RMS greater than
  - RMS 90 greater than
  - Mean lower than
  - Mean 90 lower than
  - Mean greater than
  - Mean 90 greater than
  - RMS within range
  - RMS 90 within range
  - **Fit function and check parameters within range**
  - Distance between two values

Incoming work: possibility to test any object in ROOT files using these quality tests
Conclusion

- Development of a new **generic framework** for data quality monitoring
- Used during test-beam by different detectors and **combination of sub-detectors**
- Current implementation works for online setup

Perspectives

- Refactoring of the framework to make it working for **offline data quality monitoring**
- Development of a EUDAQ binding for online data taking
- Development of quality test templates
GitHub collaboration
  
  ⚡️ https://github.com/dqm4hep

Installation package (v04-03-00)
  
  ⚡️ https://github.com/dqm4hep/dqm4hep

Slack channel (Announcements, issues, management)
  
  📨 https://dqm4hep.slack.com

Contact us!

- R. Ete (remi.ete@desy.de)
- A. Pingault (antoine.pingault@ugent.be)
- T. Coates (tc297@sussex.ac.uk)