The DQM4hep project.

CHEP 2018 conference

Rémi Ete, Antoine Pingault

DESY

July 10, 2018
Main goals of DQM systems in HEP

- Evaluate data quality and alert users of possible anomalies
  - Are the data what you expect?
  - Are the data comparable to a previous set of data?
  - Online: quick feedback from (sub) detector

- Online and offline monitoring
  - Distributed system (TCP/IP)
  - Qtest automation
  - Event display
  - Visualization interface (Desktop, Web)

Data is the central concept in such systems. But ...

- Existing framework highly dependent on event data model
- Leads to duplicated software
- Test-beam setup → ad-hoc software solution

Development of a generic DQM software for any HEP experiment
The DQM4hep framework

Central ideas

Plugin system

- User’s logic encapsulated in **Plugins**
- Plugin libraries loaded at runtime
  \(\Rightarrow\) Plug user’s logic in the framework
- Plugin is non-intrusive
  - No class inheritance
  - No in-class definition (e.g. `ClassDef`)

Abstract event data model (EDM)

- No event data model \(\rightarrow\) abstracted and user defined
- Event streamer implemented as **Plugins**

Online analysis framework fully based on abstract EDM and plugin system!
The DQM4hep framework

Online VS offline

**Online**

- Interface to DAQ systems
  - DAQ data transfert (EventSource and EventStreamer)
  - DAQ run control commands/state/config (RunControlInterface)
- Online data processing
  - DAQ data monitoring (AnalysisModule)
  - Slow control monitoring (StandaloneModule)
  - DAQ data re-processing from file (EventReader)

**Offline**

- General purpose data monitoring
  - Data quality assertion and reporting (qtest, qreport)
  - Comparison with reference data (Chi2, Kolmogorov, etc...)

+ common visualization tools
Monitor element

- Holds two TObject objects (ROOT)
  - The main monitor object: TH1, TGraph, scalars, etc ...
  - An optional reference object

Quality test

- Implements the logic to test a monitor element
- Output a quality report (quality, flag, message, ...)
- Examples:
  - Expect rms of distribution to be below a threshold
  - Fit a gaussian on a graph and check if mean is within range
  - Perform Kolmogorov/Pearson test using a reference
Assessing data quality
The quality test runner

- Runs a series of quality tests
- Output quality test reports
  - Shown in shell
  - Write in json file
- XML input description
  - Configure quality tests to run
  - Describe monitor objects to read
  - Reference objects to attach (optional)
- Currently available qtests:
  - Kolmogorov test
  - Chi2 test
  - Exact ref compare test
  - Fit property within expected
  - Property below, within, above expected

Possible shell output:

<table>
<thead>
<tr>
<th>NAME</th>
<th>QTEST</th>
<th>STATUS</th>
<th>QUALITY</th>
<th>MESSAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>DblGaus_Mean15_RMS2_RMS5</td>
<td>MeanAround15Short</td>
<td>SUCCESS</td>
<td>0.998484</td>
<td>Expected 15, got 15.0019</td>
</tr>
<tr>
<td>Gaus_Mean10_RMS2</td>
<td>MeanAround10Long</td>
<td>SUCCESS</td>
<td>0.997348</td>
<td>Expected 10, got 10.0133</td>
</tr>
<tr>
<td>Gaus_Mean10.RMS2.bck</td>
<td>MeanAround10Short</td>
<td>FAILURE</td>
<td>0.388153</td>
<td>Expected 10, got 5.6458</td>
</tr>
</tbody>
</table>
The DQM4hep online architecture

DAQ data monitoring
The DQM4hep online architecture

Slow control data monitoring
The DQM4hep online architecture

Re-process DAQ data: file reader

DQM4HEP

Physicist
DQM Internal
DAQ
Shifter

DISK

Data analysis

Monitor element collectors

Shifters

R. Ete — DESY — July 10, 2018 — Page 9
Web monitoring interface browser. Ongoing ...
DQM4hep integration

Currently using DQM4hep

- CALICE SDHCAL online system
  - Hit map, GRPC HV/current, beam analysis, electronics performances
- CALICE AHCAL (quasi-)online system
  - Hit correlations, hit maps, SiPM currents, electronics performances

Future integration

- EUDAQ framework (+ beam telescopes)
- ILCS soft simulation data monitoring (Continous integration)
- DREAM calorimeter (Dual-readout calorimeter)
- DAMIC experiment (Dark matter)
Ongoing work on software

Latest version is v01-04-04. Used as proof of principle.

But, suffers from many things:

- Link to DAQ run control not possible. Run started manually...
- Clumsy ROOT Qt plugin installation. ROOT full installation often needed
- No separation between online and offline tools. All are online somehow...

New version coming soon!

- Moved to web visualization tools (js + JSROOT)
- Link to DAQ run control finally implemented
- Packages split into more granular sub-packages
  - DQMCore, DQMNet, DQMOnline, DQMVisualization, etc...

Next big steps:

- EUDAQ interface (AIDA2020)
- DESY slow control (AIDA2020)
Conclusion and outlook

Conclusion:

- A generic DQM software solution is being developed
- The abstract event interface allows different experiments to use it
- Prototypes/experiments already use it
  - CALICE SDHCAL
  - CALICE AHCAL
- Currently finishing last master-piece: the web monitoring front-end

Outlook:

- Next major step is EUDAQ integration
  - Will bring a full community as new users!
- New integrations coming soon
  - DAMIC experiment (Dark matter)
  - DREAM Calorimeter
- Always looking for new collaborations!
GitHub collaboration

https://github.com/dqm4hep

Documentation

https://dqm4hep.github.io/dqm4hep-doxygen/


Slack channel (Announcements, help, management)

https://dqm4hep.slack.com

Citation

http://doi.org/10.5281/zenodo.1012575

10.1109/NSSMIC.2016.8069668

Contact us!

- R. Ete (remi.ete@desy.de)
- A. Pingault (antoine.pingault@ugent.be)
- T. Coates (tc297@sussex.ac.uk)
Backups
The DQM4hep online system components

Provide monitoring of data recorded by the DAQ system.

Basic sub-components:

- **Run Control**
  - DQM application receiving commands/state/config from DAQ run control.
  - Forward it to DQM applications

- **Run Control Interface**
  - Interface to connect to DAQ run control, used by the DQM run control

- **Event Streamer**
  - Convert DAQ event structure ↔ binary

- **Event Source**
  - DQM component to be integrated into DAQ to send events to DQM

- **Event Collector**
  - Collect events from event sources and re-distribute to DQM applications

- **Module**
  - Analyse data from DAQ or other data source (e.g. slow control).
  - Produces monitor elements and run quality tests

- **Monitor Element Collector**
  - Collect monitor elements from modules and re-distribute them to shifters
Old Qt4 GUI monitoring interface
Old Qt4 GUI monitoring browser

![Old Qt4 GUI monitoring browser](image-url)