The DQM4hep project.

CHEP 2018 conference

Rémi Ete, Antoine Pingault

DESY

July 10, 2018

Data quality monitoring software

in a nutshell ...

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
  $\Longrightarrow$ 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 → abstracted and user defined
- Event streamer implemented as **Plugins**

Online analysis framework fully based on astract 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

The DQM4hep building blocks

Monitor elements and QTests

Monitor element

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

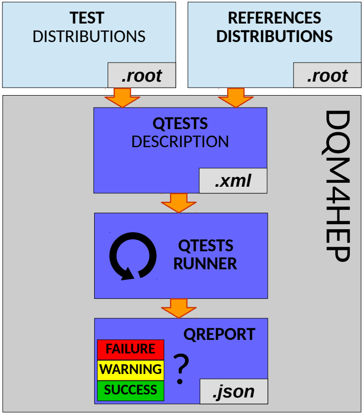
Quality test

- 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

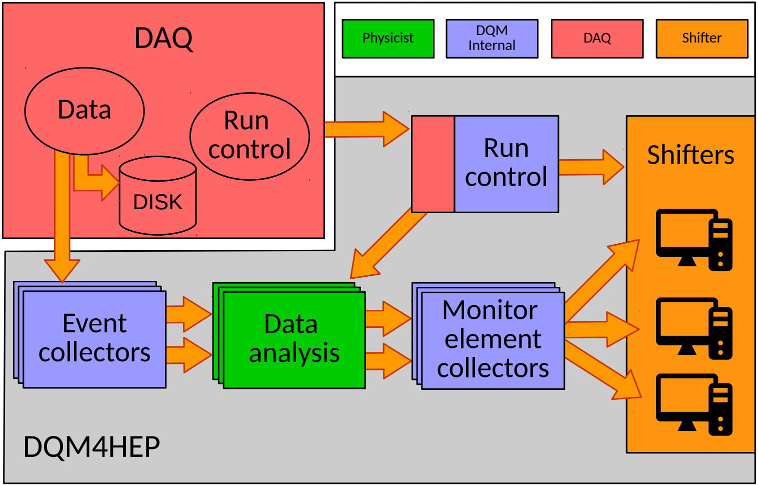


Possible shell output:

```
NAME                       QTEST                STATUS    QUALITY    MESSAGE
DblGaus_Mean15_RMS2_RMS5   MeanAround15Short    SUCCESS   0.998484   Expected 15, got 15.0019
Gaus_Mean10_RMS2           MeanAround10Long     SUCCESS   0.997348   Expected 10, got 10.0133
Gaus_Mean10_RMS2_bck       MeanAround10Short    FAILURE   0.388153   Expected 10, got 5.6458
```

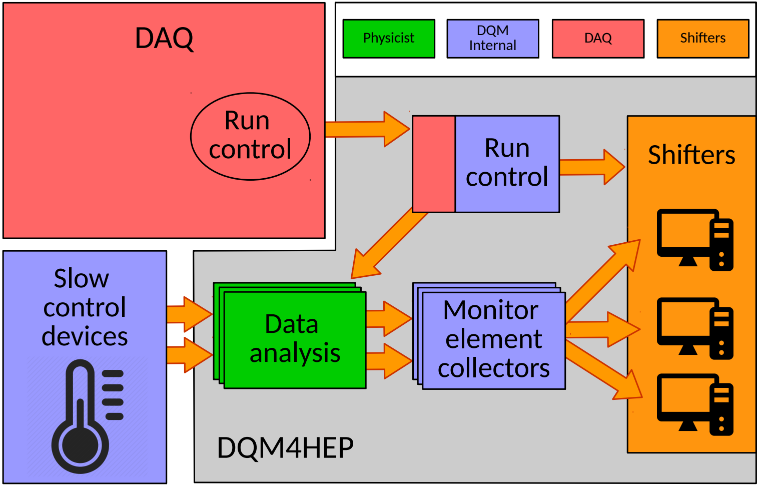
The DQM4hep online architecture

DAQ data monitoring



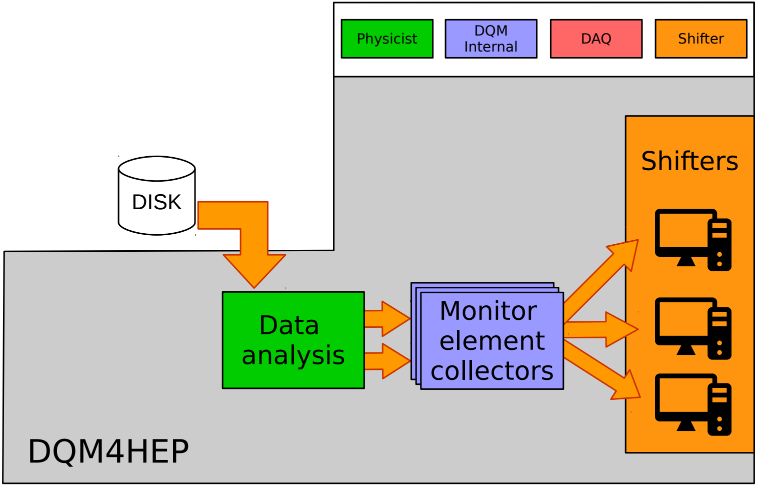
The DQM4hep online architecture

Slow control data monitoring

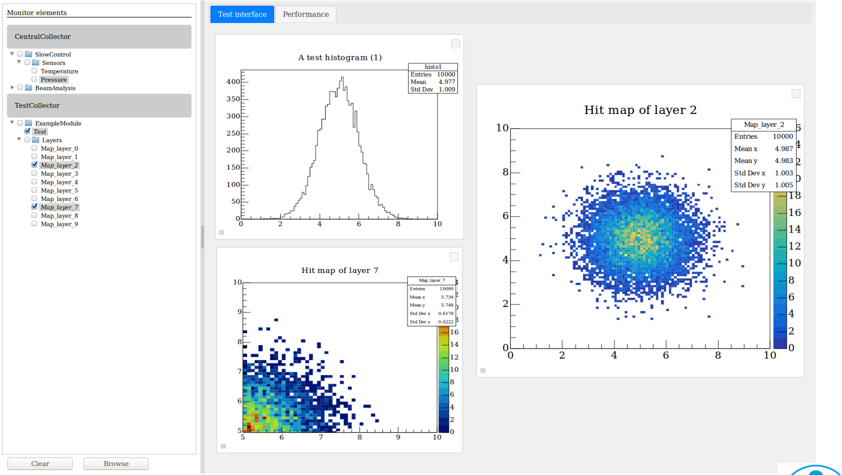


The DQM4hep online architecture

Re-process DAQ data: file reader



Web monitoring interface. Ongoing ...



Monitor elements

CentralCollector

- ▼ □ SlowControl
  - ▼ □ Sensors
    - □ Temperature
    - □ Pressure
- ▶ □ BeamAnalysis

TestCollector

- ▼ □ ExampleModule
  - ☑ *Test*
  - ▼ □ Layers
    - □ Map_layer_0
    - □ Map_layer_1
    - ☑ *Map_layer_2*
    - □ Map_layer_3
    - □ Map_layer_4
    - □ Map_layer_5
    - □ Map_layer_6
    - ☑ *Map_layer_7*
    - □ Map_layer_8
    - □ Map_layer_9

Clear | Browse

Test interface | Performance

A test histogram (1)

histo1	
Entries	10000
Mean	4.977
Std Dev	1.009

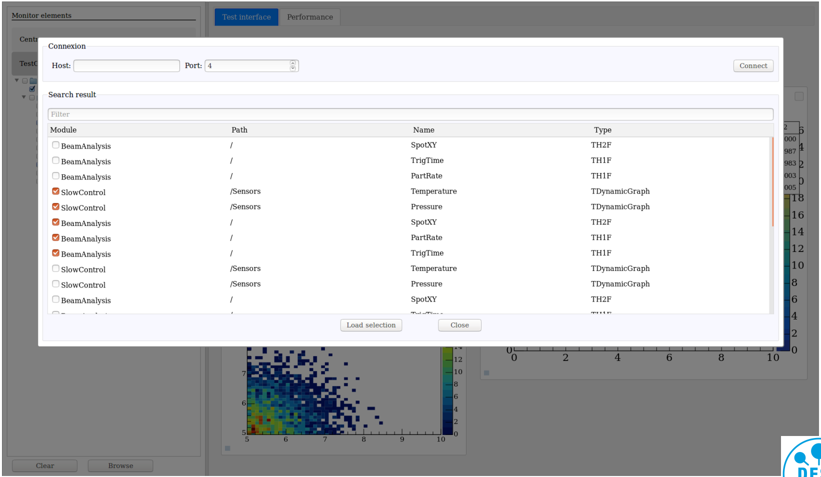
Hit map of layer 7

Map_layer_7	
Entries	10000
Mean x	5.734
Mean y	5.748
Std Dev x	0.6178
Std Dev y	0.6222

Hit map of layer 2

Map_layer_2	
Entries	10000
Mean x	4.987
Mean y	4.983
Std Dev x	1.003
Std Dev y	1.005

Web monitoring interface browser. Ongoing ...



Host:

Port: 4

Module	Path	Name	Type
□ BeamAnalysis	/	SpotXY	TH2F
□ BeamAnalysis	/	TrigTime	TH1F
□ BeamAnalysis	/	PartRate	TH1F
☑ SlowControl	/Sensors	Temperature	TDynamicGraph
☑ SlowControl	/Sensors	Pressure	TDynamicGraph
☑ BeamAnalysis	/	SpotXY	TH2F
☑ BeamAnalysis	/	PartRate	TH1F
☑ BeamAnalysis	/	TrigTime	TH1F
□ SlowControl	/Sensors	Temperature	TDynamicGraph
□ SlowControl	/Sensors	Pressure	TDynamicGraph
□ BeamAnalysis	/	SpotXY	TH2F
□ BeamAnalysis	/	TrigTime	TH1F

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)
- ILCSoft 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
- Allways looking for new collaborations !

DQM4hep

URLs and contact

GitHub collaboration

https://github.com/dqm4hep

Documentation

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

Read the Docs http://dqm4hep.readthedocs.io/en/latest/

Slack channel (Announcements, help, management)

https://dqm4hep.slack.com

Citation

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

IEEE 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

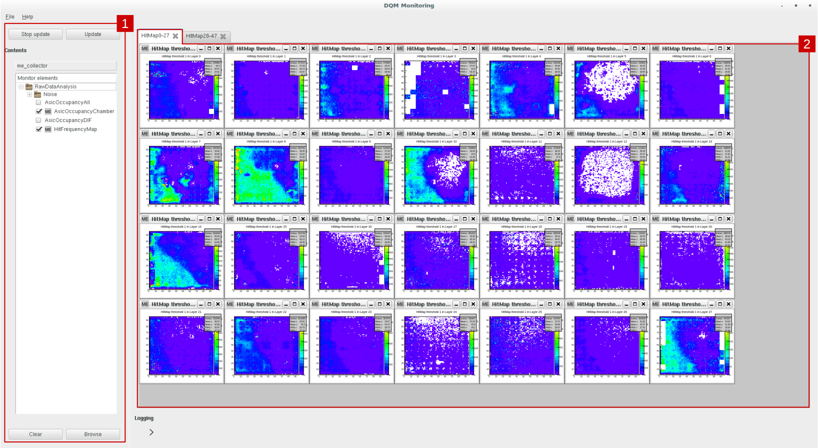
The DQM4hep online system components

Provide monitoring of data recorded by the DAQ system.

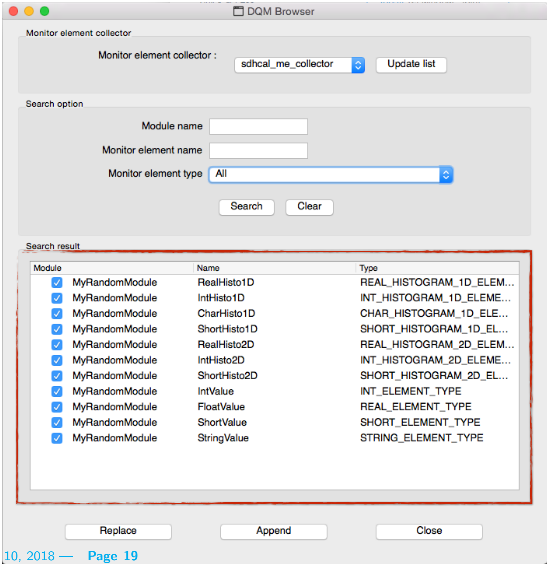
Basic sub-components:

- **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


DQM Monitoring
File
Help
1
Stop update
Update
Contents
me_collector
Monitor elements
RawDataAnalysis
Noise
AsicOccupancyAll
☑ ME AsicOccupancyChamber
AsicOccupancyDIF
☑ ME HitFrequencyMap
Clear
Browse
HitMap0-27
HitMap28-47
2
ME HitMap thresho...
HitMap threshold 1 in Layer 0
HitMap threshold 1 in Layer 1
HitMap threshold 1 in Layer 2
HitMap threshold 1 in Layer 3
HitMap threshold 1 in Layer 4
HitMap threshold 1 in Layer 5
HitMap threshold 1 in Layer 6
HitMap threshold 1 in Layer 7
HitMap threshold 1 in Layer 8
HitMap threshold 1 in Layer 9
HitMap threshold 1 in Layer 10
HitMap threshold 1 in Layer 11
HitMap threshold 1 in Layer 12
HitMap threshold 1 in Layer 13
HitMap threshold 1 in Layer 14
HitMap threshold 1 in Layer 15
HitMap threshold 1 in Layer 16
HitMap threshold 1 in Layer 17
HitMap threshold 1 in Layer 18
HitMap threshold 1 in Layer 19
HitMap threshold 1 in Layer 20
HitMap threshold 1 in Layer 21
HitMap threshold 1 in Layer 22
HitMap threshold 1 in Layer 23
HitMap threshold 1 in Layer 24
HitMap threshold 1 in Layer 25
HitMap threshold 1 in Layer 26
HitMap threshold 1 in Layer 27
Logging

Old Qt4 GUI monitoring browser



Monitor element collector

Monitor element collector : sdhcal_me_collector

Search option

Module name

Monitor element name

Monitor element type: All

Search result

Module	Name	Type
☑ MyRandomModule	RealHisto1D	REAL_HISTOGRAM_1D_ELEM...
☑ MyRandomModule	IntHisto1D	INT_HISTOGRAM_1D_ELEME...
☑ MyRandomModule	CharHisto1D	CHAR_HISTOGRAM_1D_ELE...
☑ MyRandomModule	ShortHisto1D	SHORT_HISTOGRAM_1D_EL...
☑ MyRandomModule	RealHisto2D	REAL_HISTOGRAM_2D_ELEM...
☑ MyRandomModule	IntHisto2D	INT_HISTOGRAM_2D_ELEME...
☑ MyRandomModule	ShortHisto2D	SHORT_HISTOGRAM_2D_EL...
☑ MyRandomModule	IntValue	INT_ELEMENT_TYPE
☑ MyRandomModule	FloatValue	REAL_ELEMENT_TYPE
☑ MyRandomModule	ShortValue	SHORT_ELEMENT_TYPE
☑ MyRandomModule	StringValue	STRING_ELEMENT_TYPE