ATLAS DQM: brief overview

Peter Onyisi

ALICE QA Workshop, 3 Dec 2018

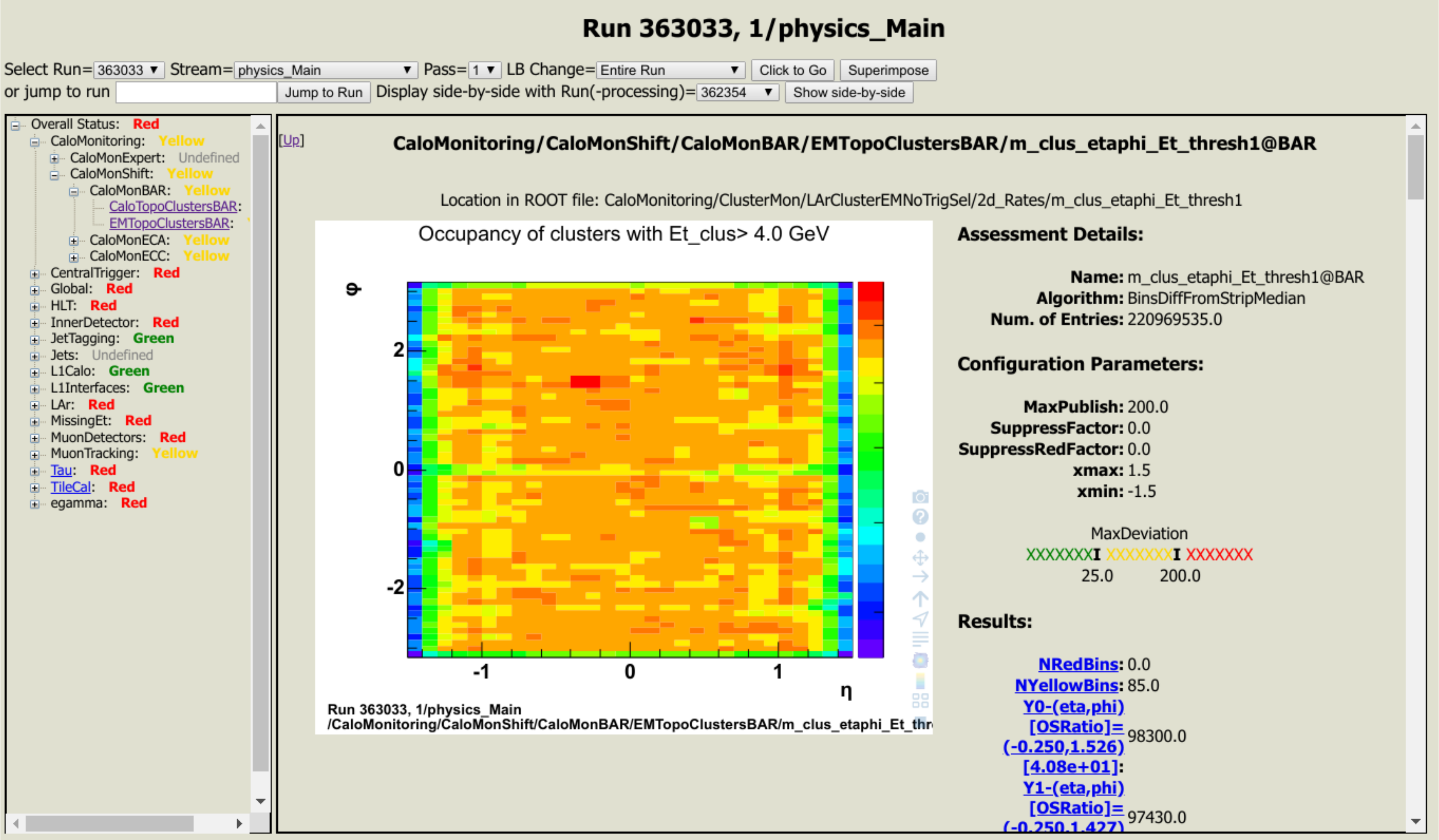
TEXAS
The University of Texas at Austin

General DQM overview

Automated framework paper:
Data Quality Monitoring Framework for the ATLAS Experiment at the LHC

- Online:
  - histograms generated by several frameworks, including full Athena reconstruction of some events; shared over DAQ interprocess communication
  - tested with Data Quality Monitoring Framework (DQMF)
  - displayed to users through C++ GUIs
  - histograms snapshotted at regular intervals & archived; DQMF results stored
- Offline:
  - histograms generated in Athena reconstruction of all events, stored in HIST files & saved to grid
  - tested with DQMF; results stored in special ROOT files, used for visualization, archived on disk & EOS
  - DQMF results (only) displayed to users through web GUI
  - other information sources (e.g. detector slow control status) used to automatically mark data quality (“defects”)

Offline Display + DQMF



Offline "Defects"

ATLAS DQ Defect Entry System

You are logged in as *ponyisi*. Log out of CERN applications

Database: Production ▾   Tag: HEAD ▾

Show defects in a run | Upload | Sign off a run | Multirun upload — alternate format

Filter:

Show defects marked absent: ☐

Show defects in run 350479 ▾ Load

Hover mouse pointer over LB ranges to see comments, over defect names to see descriptions.
Bold *defects are considered intolerable by someone.*

Defect	Present in LBs
EGAMMA_ETAPHI_SPIKES	1-719
GLOBAL_BUSY	1-7, 12-16, 49, 456, 501, 547, 608, 613, 658
GLOBAL_NOTREADY	1-64, 718-719
ID_BS_RUNAVERAGE	1-64, 718-719
ID_IBL_TRACKCOVERAGE	1-719
JET_LOWOCCUPANCY	1-719
LAR_DATACORRUPT	117, 118-132, 133, 199, 717
LAR_EMBA_DATACORRUPT	118-132
LAR_EMECA_NOISEBURST	69, 75, 78, 118, 123, 142, 150, 154, 161, 165-166, 169-171, 176, 206, 224, 227, 250, 263, 265, 280, 283, 298, 300, 328, 352, 357, 366, 371, 375, 376, 380-381, 383, 390, 394, 400-401, 404, 426, 440-442, 445, 460-461, 468, 469, 481, 516, 534, 549, 592, 599, 611, 634, 639, 643, 646, 651, 661, 670, 712-713
LAR_EMECC_NOISEBURST	125, 140, 165, 167, 173, 180, 190, 203, 212, 215, 218, 221, 228, 235, 257, 261, 277, 296-297, 299, 324, 344, 352, 379, 403, 405, 408, 416, 431, 441, 445, 468, 488, 518, 551, 599, 607, 633-634, 659, 710

Q&A

How similar are your online and offline DQM systems?

- They share a common framework for testing histograms (DQMF), and common algorithms for doing so. Otherwise not similar –
  - different configuration languages (XML vs domain-specific language);
  - different visualizations (C++ with DAQ software dependencies vs web app);
  - different persistency backends
- (Aside: some of the DQMF algorithms are very clever, e.g. hot/cold spot finding in 2D histograms taking into account η dependence.)

Q&A

How are checks & GUIs configured? Are users easily able to change them (e.g. few min latency) or is there an approval process required?

- DQMF uses more-or-less static configurations (XML online/domain-specific language offline). In production configurations, online can be changed relatively quickly, offline requires git pull request & manual deployment at Tier-0.

Are users able to run their own DQM instances?

- Yes, although it's a bit tricky online (DQM not self-contained), and offline visualization uses a central server which picks up the results of user runs.

Q&A

How important is archiving results to you? What backends do you use?

- Very important. Both online and offline are archived.
  - Online status flag changes are archived; histograms are snapshotted.
  - All offline results (computed over full run) are archived in ROOT files and stored on a server & on EOS. Complete info – plots, references, DQMF test results. An attempt was made to put results into a relational database, but was limited by large number of rows and write once/read rarely nature of many plots.

How automated are the final good/bad decisions for data?

- Detector Control System checks are automatic. Some (very limited) automation for detector configuration
  - Flagging of potential issues for shifter consideration is quite advanced for some detectors – e.g. automated hotspot extraction in η-φ plots based on difference-from-median in η strips

Q&A

Do you provide access to the DQM data through some API?

- Yes. Online data require somewhat specialized APIs; all offline data available via XML-RPC.

Do you run code as part of data reconstruction? If so, how constraining is this for you?

- Yes. It is run as part of a standard Athena reconstruction job.
  - creates significant memory constraints, which affect our choice of workflow (e.g. RAW data → HIST + all other outputs in one step can only run on Tier-0, where we have lots of memory per core).
  - improvement here partially motivates multithreaded Athena.

ATLAS DQ & Machine Learning

- No active deployment of ML in DQM during Run 2. Developments under consideration for Run 3.
- Investigated so far:
  - prediction of L1 trigger rates from luminosity, learning from time series in a given run
  - anomaly detection: try to flag luminosity blocks which look "different" from others (using autoencoders, BDTs, ...)
- Conceptual ideas:
  - automated prediction of reference histograms given e.g. luminosity, run length ...
  - discover correlations of detector "defects" and characteristics of monitoring histograms (and learn to predict)
- General problems:
  - very easy to have false positives (keep discovering that luminosity / prescales change during run)
  - need value-added over human checks