Review of the Trigger Offline Monitoring

Trigger Offline Monitoring
Review aims and procedure
Review Questions and answers
Last Comments

Szymon Gadomski and Ricardo Gonçalo
Trigger Open Meeting, 21/1/2009

Trigger Offline Monitoring

The offline monitoring area covers two parts of the Trigger Operation:

- Processing or reprocessing of jobs for:
  1. Testing new software and menus
  2. Producing HLT data when this was not in the run (*)
  3. Produce ESD's/monitoring output from bytestream with or without HLT info
  4. Special monitoring jobs that cannot run at Tier0
  5. Processing and classifying data from the debug stream

(*) i.e. running the HLT on data selected by Level 1 only (and with a minimal HLT menu for streaming)

- Monitoring and DQ of the Trigger
  1. Analyze the histograms from Tier0
  2. Correlate them with the online histograms written after the run
  3. Produce an assessment of the DQ from the Trigger

- People involved:
  1. Trigger offline expert
  2. Trigger offline shifter
  3. Expert(s) providing support for monitoring infrastructure
  4. Trigger slice on-call experts

Review Aims

- Find any necessary improvements in the organisation of the offline monitoring shifts, including documentation and flow of information
- Determine the need for new functionality in the monitoring infrastructure
- Help to formalize roles and responsabilities of the trigger offline monitoring shifter and expert
- Establish a tentative operational schedule for each role
- Estimate the use of computing resources and find any additional needs

Starting questions:

- How to submit jobs? It is automatic enough?
- What tools exist and what are still needed?
- Where and how the log files and other data is stored?
- How the DB of what was run is stored?
- How results are published and documented?
- Is the infrastructure for testing fast/prepared enough?
- How should the histograms checking work?
- What should be the interaction with the slice experts? (I believe this will improve when we are in beam)
- How the report of the shifter per run should be given?
- What other tools/system do we need?

Review Procedure

A few complementary procedures were followed:

1. Analyze the software infrastructure and see what is still missing
2. Analyze the hardware available and explore what may be needed
3. Build a view on how things will work when we are in running
   - Can the shifter be remote?
   - What is still needed to make sure the communication expert/shifter is efficient even in this case?
   - Can the expert be remote?
   - How tasks are assigned and resolved?
   - Could we describe a "working day" for a shifter and an expert?
4. Consult with the people who have been experts and shifters
5. Consult with people responsible for the software infrastructure
6. Take into account that the system was still in building phase and some comment might not be relevant
7. Extrapolate to the future to see what the week points will be

- Input from shifters/experts: Olya, Iwona, Aart, Anna, Hegoi, Attila, so far (also Szymon, Ricardo)

- Some partial conclusions already possible, but another week or two would help refine things

- How to submit jobs? It is automatic enough?
- What tools exist and what are still needed?
- Where and how the log files and other data is stored?
- Is the infrastructure for testing fast/prepared enough?

Trigger Tasks in CAF

CAF was used in 2008 run for 3 main purposes:

1. Run High Level Trigger on Level 1-selected bytestream data
   - Test new Super Master Keys before online deployment
   - Classify High Level Trigger errors, crashes, etc
2. Run trigger offline monitoring on bytestream data from step 1
3. Produce ESDs with trigger information from step 1 bytestream data for analysis
4. Estimate trigger rates for new menus (occasional and lower priority)

Plans for the CAF in 2009:

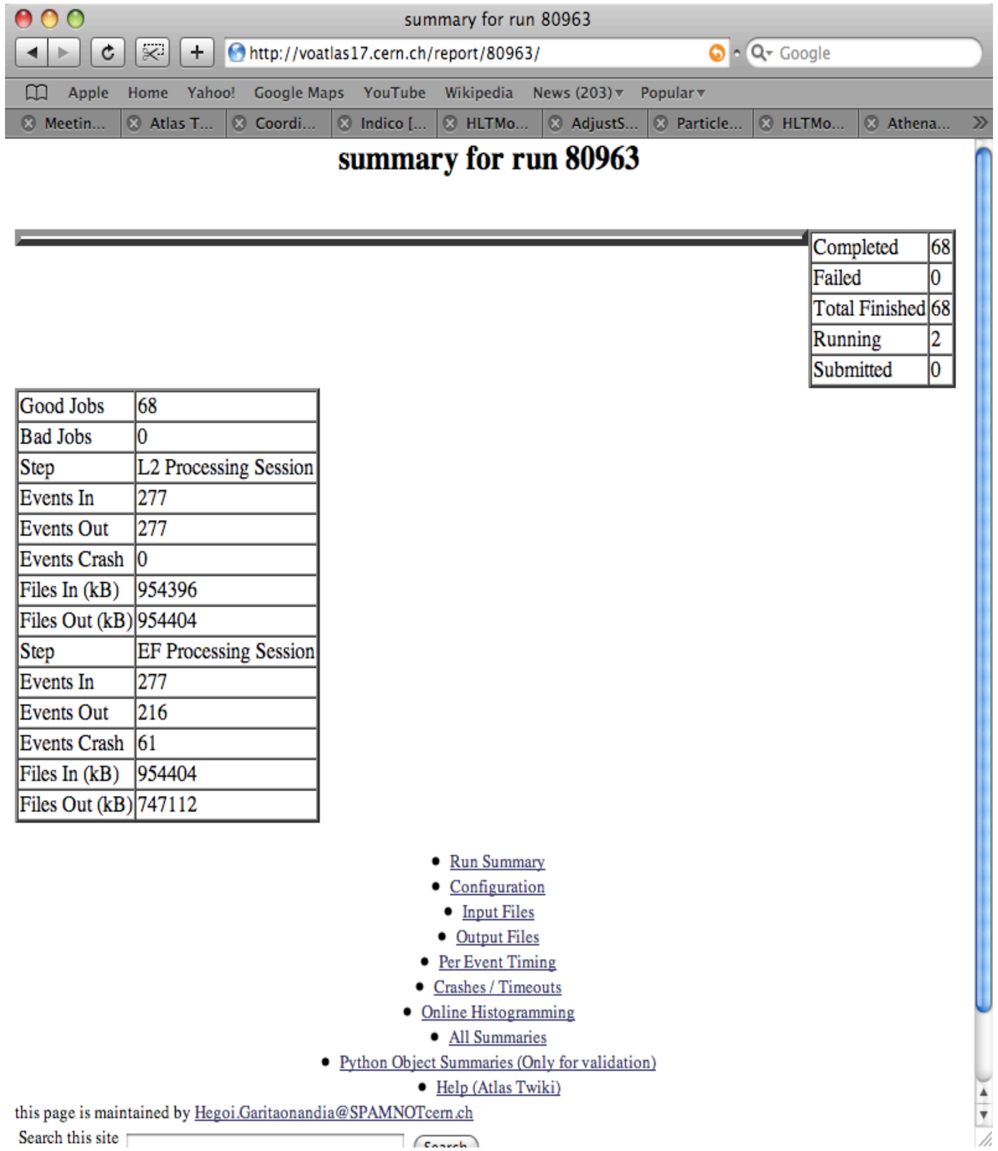
- Initial running will be pretty much the same as 2008 (running HLT on L1-only data, etc)
- Plans for steady-state data taking :
  1. Run error analysis/classification/recovery on all debug stream events
  2. Run Data Quality monitoring jobs on some/all express stream data
  3. Run online/offline trigger result comparison on some/all express stream data
  4. Continue to test new menus and code offline in the CAF before deploying them

Task Management

- Initial system written and developed for 2008 run:
  - HDEBUG framework - Hegoi Garitaonandia
  - Error analysis scripts - Anna Sfyrla
  - Offline monitoring, BS->ESD - Aart Heijboer
- Job submission for step 1 used HDEBUG, based on GANGA, and publishes results to web server
- Monitoring jobs run trigger monitoring tools in CAF/Tier0
- Monitoring and ESD (steps 2. and 3.) used simple queue submission scripts (bsub)
- Small library of useful scripts for error classification, etc

Recommendations for 2009:

- Automate job submission in HDEBUG framework – eliminate manual submission of jobs on debug and express stream
- Complete merger of error classification scripts into HDEBUG
- Ongoing development of analysis algorithms for online/offline comparison – to be managed by HDEBUG
- Continue to use CAF for testing new SMKs before online deployment of menu
  - Simplify submission of test jobs and make it more robust



summary for run 80963
http://voatlas17.cern.ch/report/80963/
Google
Apple Home Yahoo! Google Maps YouTube Wikipedia News (203) Popular
Meetin... Atlas T... Coordi... Indico [... HLTMo... AdjustS... Particle... HLTMo... Athena...

summary for run 80963

Completed	68
Failed	0
Total Finished	68
Running	2
Submitted	0

Good Jobs	68
Bad Jobs	0
Step	L2 Processing Session
Events In	277
Events Out	277
Events Crash	0
Files In (kB)	954396
Files Out (kB)	954404
Step	EF Processing Session
Events In	277
Events Out	216
Events Crash	61
Files In (kB)	954404
Files Out (kB)	747112

- Run Summary
- Configuration
- Input Files
- Output Files
- Per Event Timing
- Crashes / Timeouts
- Online Histogramming
- All Summaries
- Python Object Summaries (Only for validation)
- Help (Atlas Twiki)

this page is maintained by Hegoi.Garitaonandia@SPAMNOTcern.ch

Search this site

Proposal for centralizing task management?

- Submission of jobs on DEBUG and EXPRESS streams:
  - Initial data: run by hand, possibly different menu and release than was used online
  - Steady state: run same release and menu as used online
  - Hegoi and Attila working on system to automatically send jobs to all new data from these streams – perhaps also useful elsewhere

- Testing new menus: need to specify data set, menu, release (possibly nightly)
  - Need tool to un-stream data before running – avoid mixing streams after new HLT version runs on data
  - Open question: do we need to be able to test nightly+extra tag? – initial answer is **no**

- Log files for debug stream/test jobs and reference files should be stored in web server and periodically archived to castor (in automatic way if possible)
  - Must be accessible to shifter

- Other constraints:
  - DQ, debug stream and test jobs need to publish results in web-accessible way for remote DQ
  - Need to run this asynchronously from (before) offline reconstruction
  - Farm/queue load varies mostly depending on demand for testing new menus (time critical)
    - Total farm load should be quantified and monitored

- How the DB of what was run is stored?
- How results are published and documented?

Data Quality Flags

- Conditions DB is the place to store all DQ flags for the whole collaboration
- To complement this, add eLog category for use by offline monitoring expert/shifter

- Old flags: red, green, yellow, grey, AND undefined (before any value is set)
- New proposal: **red** (bad), **green** (good), **yellow** (partially bad, some channels missing, hole in calo, etc), **black** (disabled, not in partition), **white** (in partition), **grey** (or blue, undefined, tried to check quality but couldn't)

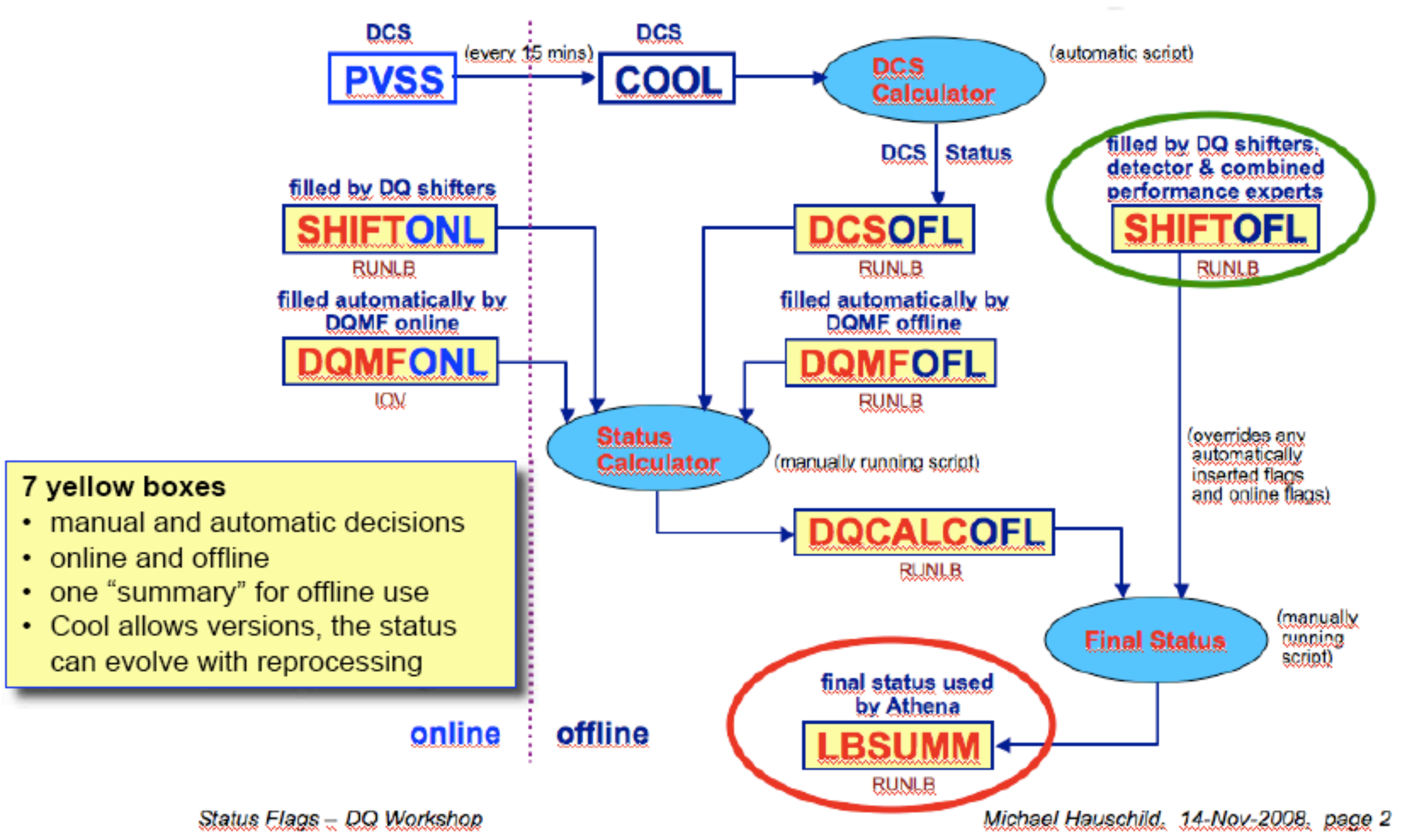
- Comments: can be added for each flag per lumi block,
  - But not in DB folder which is seen by athena
  - Flags and comments are stored only when something changes

- Evolution with time: flags are redone for each reprocessing (yearly...)

- There will be several good-run lists: DQ provides tools to query DB

- Should be possible to check if a detector is in the partition and include this in the logic for deciding on data quality flags
  - e.g. don't use some histo for checks if some detector is not in the partition

Offline DQ Monitoring


DCS
PVSS
(every 15 mins)
DCS
COOL
DCS Calculator
(automatic script)
DCS Status
filled by DQ shifters
SHIFTONL
RUNLB
filled automatically by DQMF online
DQMFONL
IOV
DCSOFL
RUNLB
filled automatically by DQMF offline
DQMFOFL
RUNLB
filled by DQ shifters, detector & combined performance experts
SHIFTOFL
RUNLB
Status Calculator
(manually running script)
(overrides any automatically inserted flags and online flags)
DQCALCOFL
RUNLB
Final Status
(manually running script)
final status used by Athena
LBSUMM
RUNLB
online
offline

7 yellow boxes
- manual and automatic decisions
- online and offline
- one "summary" for offline use
- Cool allows versions, the status can evolve with reprocessing

Status Flags – DQ Workshop *Michael Hauschild, 14-Nov-2008, page 2*

- How should the histograms checking work?
- What should be the interaction with the slice experts?
- How the report of the shifter per run should be given?
- What other tools/system do we need?

Trigger Offline Monitoring Expert Role

- The expert should at any moment be aware of the ongoing operational issues
- He/she must provide the link between the shifter and the trigger operations
  - The shifter needs clear information on the day's priorities and instructions on what to do outside the routine tasks
- Must be at CERN, at least during first year or so
- The expert has several tasks assigned:
  1. Attend the 09:00 trigger meeting, before the daily run meeting
  2. If needed, attend the run meeting
  3. Attend the meeting with trigger slice experts in the afternoon to be prepared for the DQ meeting
  4. Report on the Data Quality meeting
  5. Reassign Savannah bugs coming from Tier0
  6. Assist the trigger offline monitoring shifter

Trigger Offline Monitoring Shifter Role

- It should be possible to do the offline monitoring shift remotely

- The shifter should be expected to:
  1. Verify the daily monitoring histograms for each run
  2. Provide the expert with an accurate view of the data quality:
     - Agreement between monitoring histograms and references
     - Fraction of failed jobs, error classification
     - General quantities such as which triggers are running, stream overlaps, trigger rates, nr events processed, etc
  3. Be able to launch monitoring jobs on recent data and analyze output
     - Essential, and time critical, for testing new menus

- Good communication is essential between the shifter and the expert
  – The shifter needs clear information on the day's priorities and instructions on what to do outside the routine tasks

- Slice experts must be accessible to shifter to help interpret monitoring histograms which fall outside the norm
  – Reference histograms should be made available as soon as possible, and maintained by slice experts
  – Even before that, essential that experts supply clear description of each monitoring histogram to be checked – should be obvious when data is bad

- Shifter report might benefit from a shift checklist which writes to eLog

Other comments/conclusions on 2008 run

Other comments/conclusions on 2008 run

- Expert and shifter: clearly cannot be same person, work load too much
  - Expert:
    - 40% of time talking to people to understand problems and how to process data
    - rest looking at data quality and in meetings:
      - 1/2 hour Data Quality
      - 1/2 hour slice expert
      - 1/2 hour monitoring meeting
    - Expert sometimes needed to know how to run jobs
  - Shifter:
    - 70-80% debugging and coping with problems in the machinery or interacting with tool maintainer
    - Running jobs took a lot of babysitting
  - Whiteboard a very useful feature – needs to be kept lean and clean and up to date (expert responsibility?)

- Monitoring Tools:
  - Changes to standard procedure required a lot of manual intervention and cleanups
    - E.g. keeping log files required change of framework
  - Robustness needs to be improved: e.g. number of failed jobs in web site not always corresponded to trigger errors (monitoring thread fragile)
  - BS conversion, memory leaks, etc not done

- Documentation:
  - Reasonable documentation available, but still lots of interaction with tool experts essential
  - No description of monitoring histograms
  - Sometimes “too much to do to [thoroughly] read documentation” (cannot happen)

- Tier0:
  - Should have the same histograms in Tier0 and online
  - Should have a comparison between online and offline trigger results
  - Needed reference histograms and up to date histo boundaries - most offline histos are empty
  - Differences between online and offline went unnoticed

- CAF usage:
  - Perception was that farm load was excessive, but in practice lots of idle time
  - This should be investigated and optimised
  - Should improve with automatic job submission
  - Seems very difficult to avoid using a single (trigcomm) account

Conclusions

- Initial system worked acceptably but needed much manual intervention
- There is now opportunity to be better prepared
- Short term plans are:
  - Improve error analysis and integrate with HDEBUG
  - Automatise HDEBUG job submission
  - Produce online/offline comparison
- Should also think of shifter training
- Proposal from offline DQ to have common infrastructure for CAF – will know more soon
- I feel 1-2 more weeks needed to conclude this review