Data Quality Monitoring in the ATLAS Inner Detector

Martin White
University of Cambridge

On behalf of ATLAS, with special thanks to Florian Hirsch, Beate Hienemann, James Degenhardt, Nathan Triplett, Sergei Smirnov, Ole Rohne, Heidi Sandaker

Overview

- The ATLAS Inner Detector is a complicated system with three different components

  ⇒ pixel detector, silicon strip tracker, transition radiation tracker

- Monitoring of the detector is essential, for several reasons

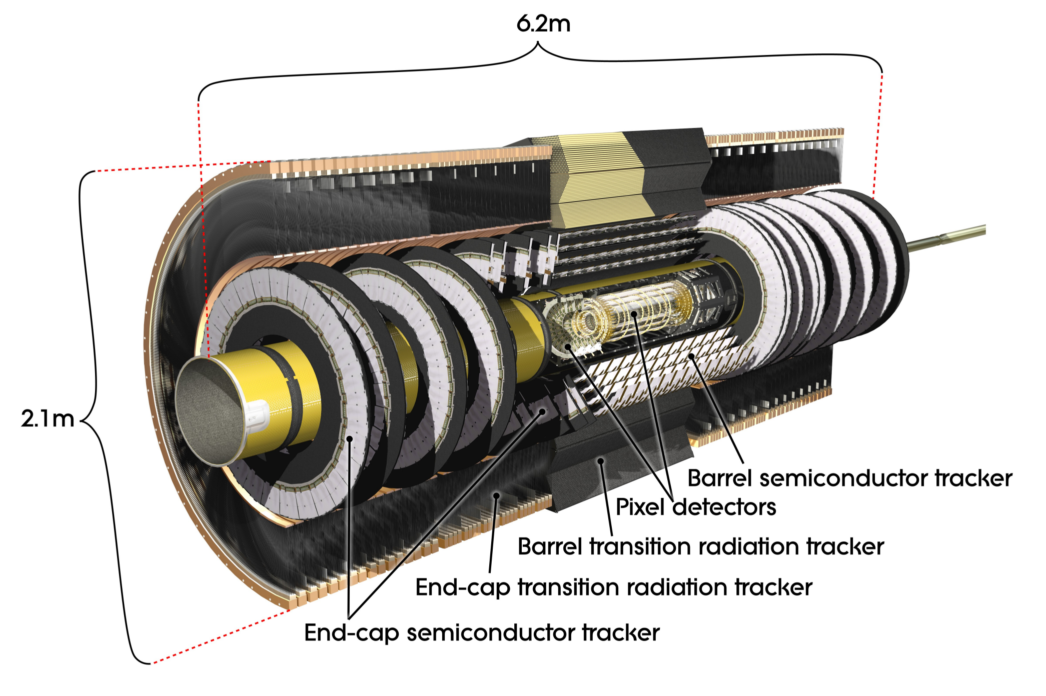
  ⇒ need to give visual feedback to shift workers during a run
  ⇒ need to monitor the quality of data sets (it's pointless reconstructing bad data!)
  ⇒ need to monitor long term degradation of the detector
  ⇒ need to store detector parameters for later use

- There are several software frameworks in ATLAS that provide these functions

- Will proceeds as follows:

1) A brief overview of the ATLAS dataflow and monitoring infrastructure

2) Examples of monitoring systems from each subsystem of the inner detector


6.2m
2.1m
Barrel semiconductor tracker
Pixel detectors
Barrel transition radiation tracker
End-cap transition radiation tracker
End-cap semiconductor tracker

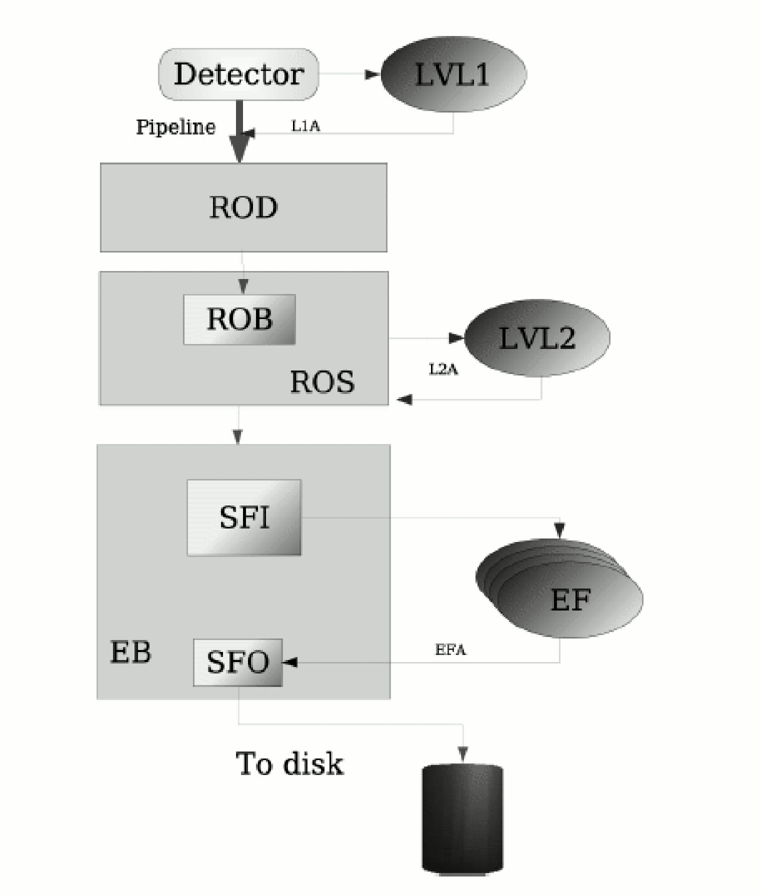
The basics of monitoring

- In the natural world monitoring works like this:

- Large, complicated systems (alpacas) keep an eye on the data stream (chickens)
- Typically they can't follow all the chickens- they sample them
- They can either watch the chickens in real time or think about the chickens again later
- One alpaca might fall asleep or get bored- use two (or more!) to add safety and redundancy

A brief guide to ATLAS dataflow

- ATLAS works on the same principle
- There are several ways of monitoring the data stream
- These are distinguished by the point at which they sample the data stream



IEEE Trans.Nucl.Sci.53:1317-1322,2006

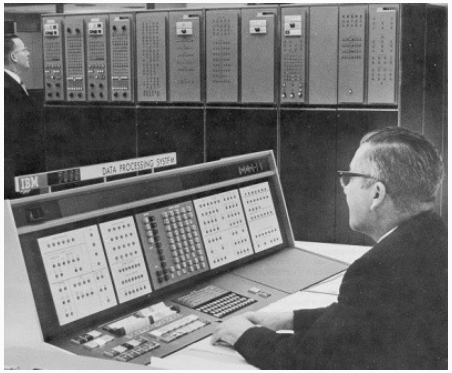
- Data from the detector pass to a Read Out Driver when the first level trigger has accepted an event
- ROD formats data, sends a basic event fragment to a Read Out Buffer (stored until level 2 trigger decision)
- If the event is good, a Read Out Subsystem moves the data block to the Event Builder
- The Event Builder stores events in a Sub Farm Input from which the Event Filter fully reconstructs them (using the ATLAS reconstruction software)
- Accepted events are now fine to be written to disk- they pass through the Sub Farm Output to a file
- Can monitor data at *any* of these stages

ATLAS Monitoring

- There are several schemes for data monitoring in ATLAS
- Online we have:

1) GNAM
⇒ can work at all DataFlow levels (ROD - ROS - SFI – SFO)
⇒ used mostly for fast feedback on detector
⇒ monitors all information available on a 'raw data' level
⇒ e.g. hits, errors, bunch crossing ID, simple derived quantities (simple clusters, averages)

2) Athena
⇒ Athena is the official ATLAS reconstruction software framework
⇒ typically runs at the last stage of the trigger process (Event Filter)
⇒ has access to fully reconstructed events!
⇒ anything using *tracks* must be Athena based
⇒ is much slower than GNAM- monitoring rates are typically ~ 1 Hz



3) Custom tools
⇒ some subsystems have histogramming tools built into the RODs (e.g SCT)
⇒ have extra systems for monitoring environmental parameters

- Offline we have:

1) Athena (can use same code as online, typically use extra tools)
2) Custom tools

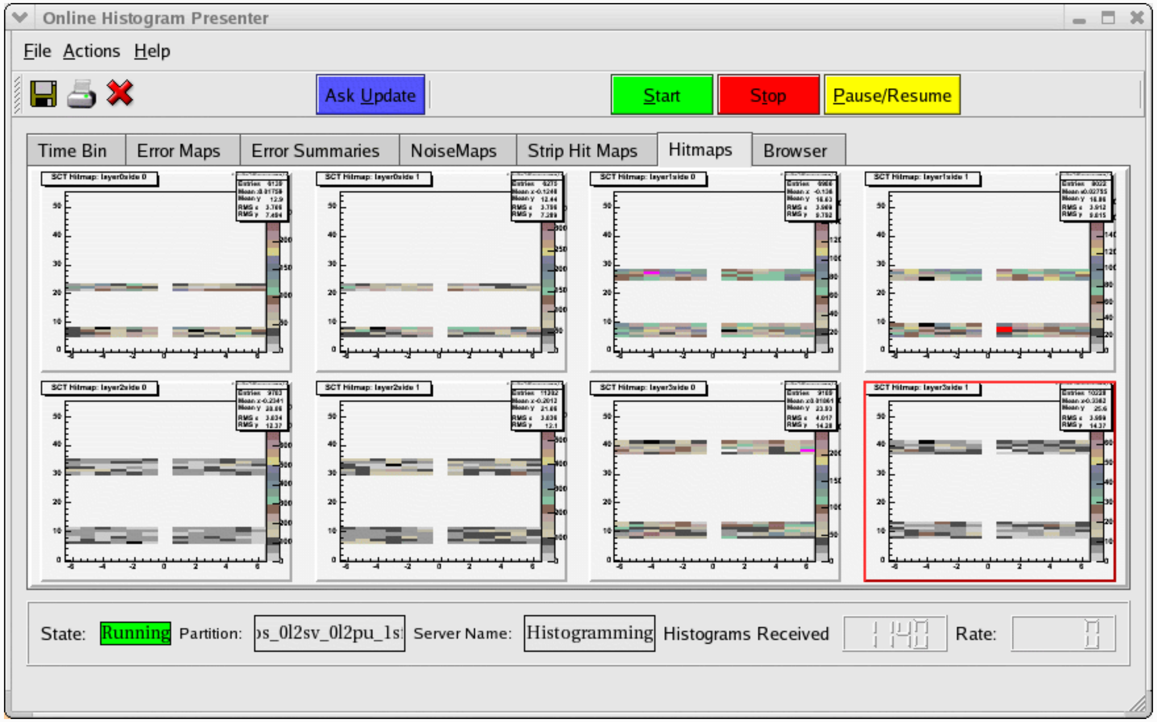
ATLAS Monitoring Output

- The offline tools write output to a ROOT file

  ⇒ contains histograms and, in the SCT case, an ntuple
  ⇒ can be checked automatically by an offline Data Quality package
  ⇒ output is also stored in a COOL database for permanent reference

- The online tools publish histograms in an online histogram server

- These are retrieved and displayed using a program called the Online Histogram Presenter



State: Running Partition: ›s_0l2sv_0l2pu_1s‹ Server Name: Histogramming Histograms Received 1140 Rate: 0

- Also send a subset of the online histograms to an automatic Data Quality checker (DQM)

  ⇒ provides simple checks and produces a 'traffic light' status

- Have separate shift workers for

  ⇒ detector subsystems
  ⇒ data quality checks

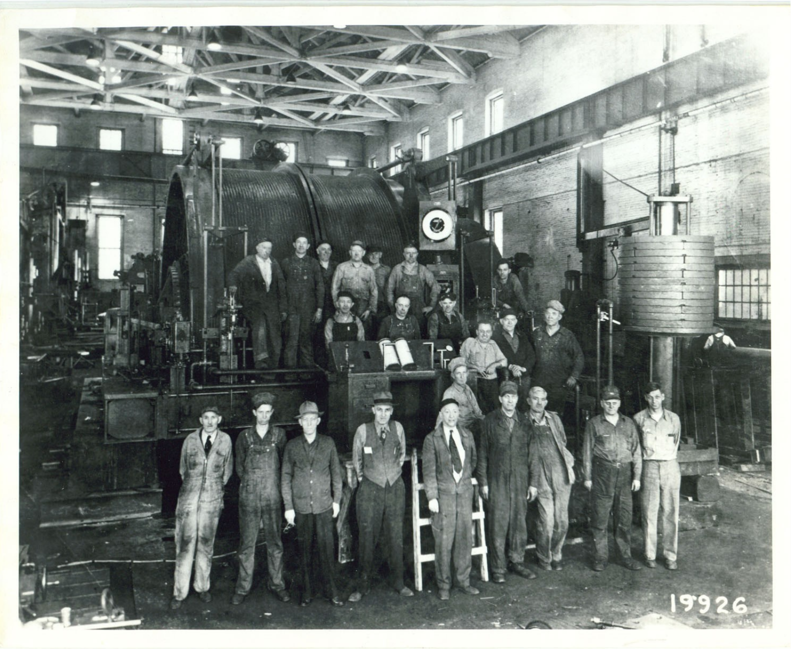
ATLAS Monitoring Infrastructure

- The core monitoring infrastructure is part of the ATLAS Trigger & Data AcQuisition (TDAQ) Software

  ⇒ provides the methods for sampling the data stream
  ⇒ keeps monitoring infrastructure relatively homogeneous across subsystems
  ⇒ synchronises monitoring with other DAQ activities

- Many algorithms also use the the Athena software framework and tools

  ⇒ thus monitoring development is truly a collaboration wide effort!
  ⇒ provides a bridge between offline and online software developers



Recent ATLAS tests

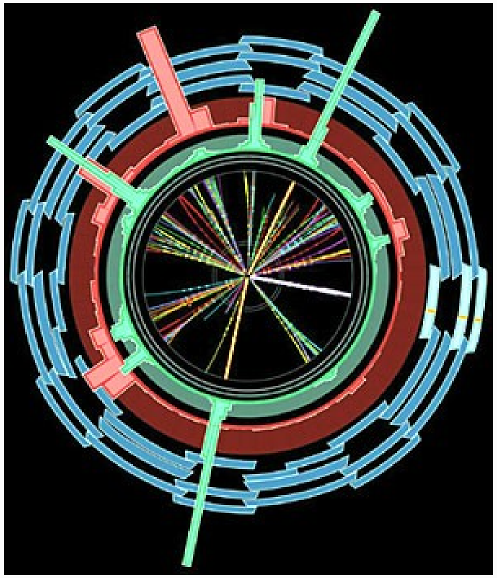
- There have been several opportunities for testing the monitoring software:

  1) M weeks. These are DAQ integration exercises during which we run as much of ATLAS as possible and test the DAQ system, including monitoring. Most recent was M8.

  2) FDR weeks. These are Full Dress Rehearsal exercises in which simulated data is piped through the offline infrastructure. Tests include monitoring and data quality checks. The most recent was FDR2b, and a FDR2c is being planned.

- In reality, we're now at the FLMC stage

  ⇒ Frantic Last Minute Commissioning!
  ⇒ every week provides a test of almost all parts of the system
  ⇒ things are looking good, we'll be ready for first data



Monitoring Examples

ATLAS ID Monitoring by Subsystem

- Each detector subsystem is responsible for its own monitoring systems
- The pixels have:

  ⇒ Athena monitoring
  ⇒ GNAM monitoring
  ⇒ some hardware level ROS monitoring

- The SCT have:

  ⇒ Athena monitoring
  ⇒ GNAM monitoring
  ⇒ ROS level monitoring

- The TRT have:

  ⇒ Athena monitoring
  ⇒ A custom monitoring tool called TRTViewer

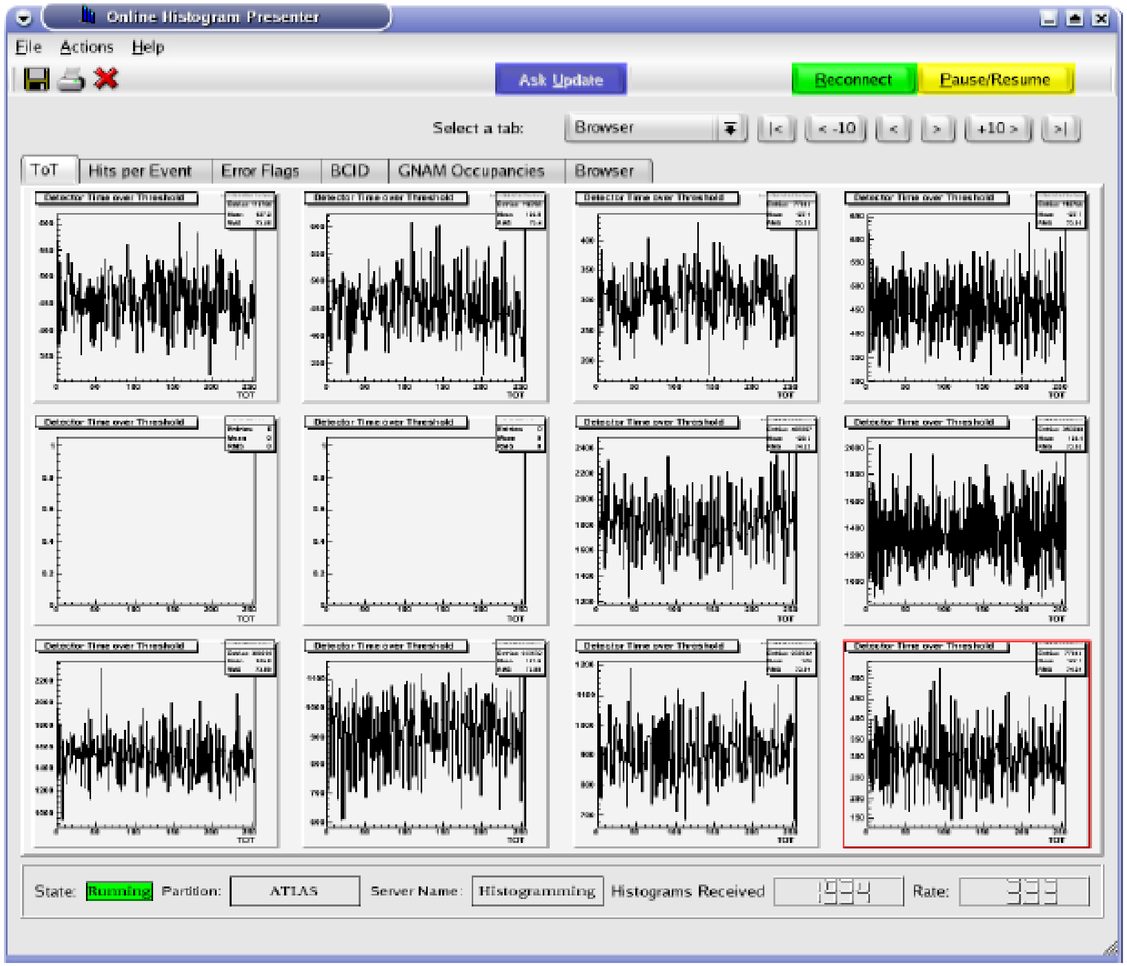
- In addition there is a global inner detector monitoring tool

  ⇒ Athena based
  ⇒ looks at synchronisation between detectors, etc

Pixel Monitoring - GNAM

- Samples events at ROS PC's (high rate ~1 kHz)

  ⇒ have 1 GNAM process per ROS on a dedicated machine (12 ROS's)
  ⇒ has survived tests up to 100 Hz, further tests are forthcoming



State: Running Partition: ATLAS Server Name: Histogramming Histograms Received 1934 Rate: 333

- Monitors:

  ⇒ total hits in each ROS
  ⇒ # hits per event
  ⇒ error flags for each ROS
  ⇒ occupancy for pixel layers

- In progress:

  ⇒ merging with DQMF (automatic data checking)

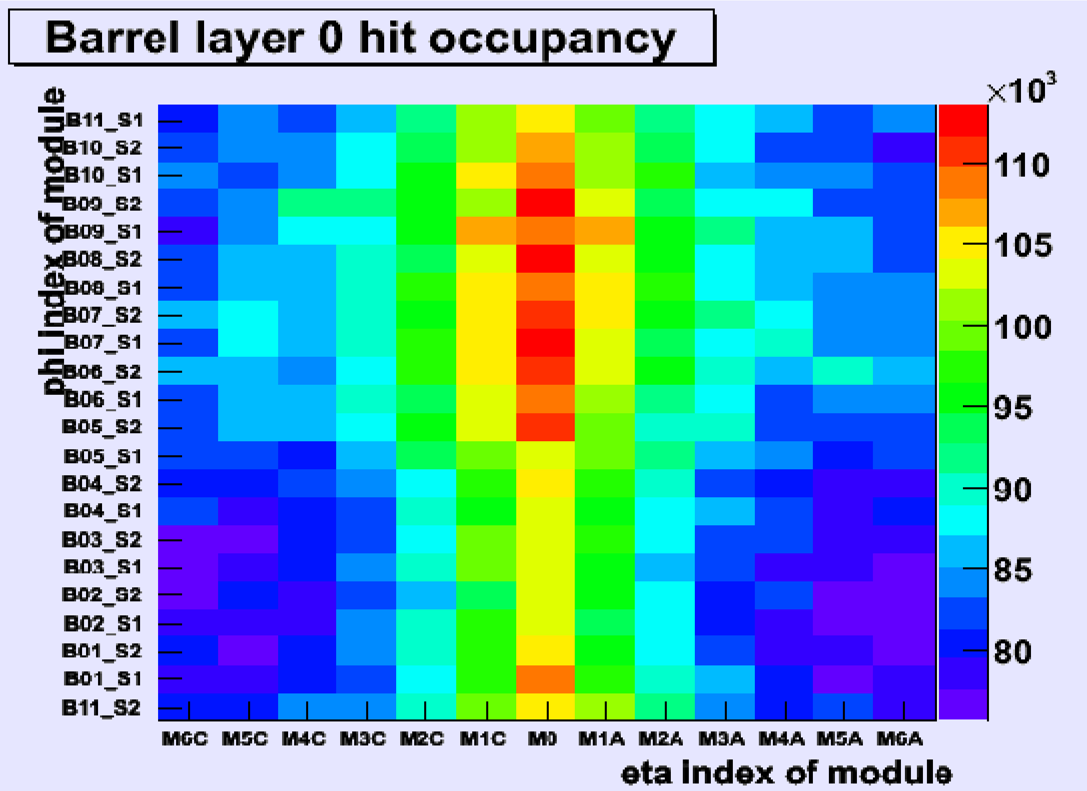
- Results are shown here for a simulator run

  ⇒ a ROD simulator is piping data through the DAQ chain

Pixel Monitoring - Athena

- Samples events from the SFI, and reconstructs Pixel clusters

  ⇒ for extra speed, reconstructs only Pixel quantities (hits and clusters only)
  ⇒ maximum rate is ~ 20 Hz
  ⇒ updated histograms are sent to shifters every 1 minute
  ⇒ an expanded version of the same code is run offline (provides more thorough DQ checks)



- Monitors:
  ⇒ occupancies at module level
  ⇒ distribution of module occupancies
  ⇒ Time over Threshold (ToT)
  ⇒ number of pixels per cluster
  ⇒ time dependent quantities (e.g. # hits per module vs time)

- This is the most recent Athena package in the ID, but it is in good shape

  ⇒ both online and offline have been tested
  ⇒ currently improving usability, training shifters

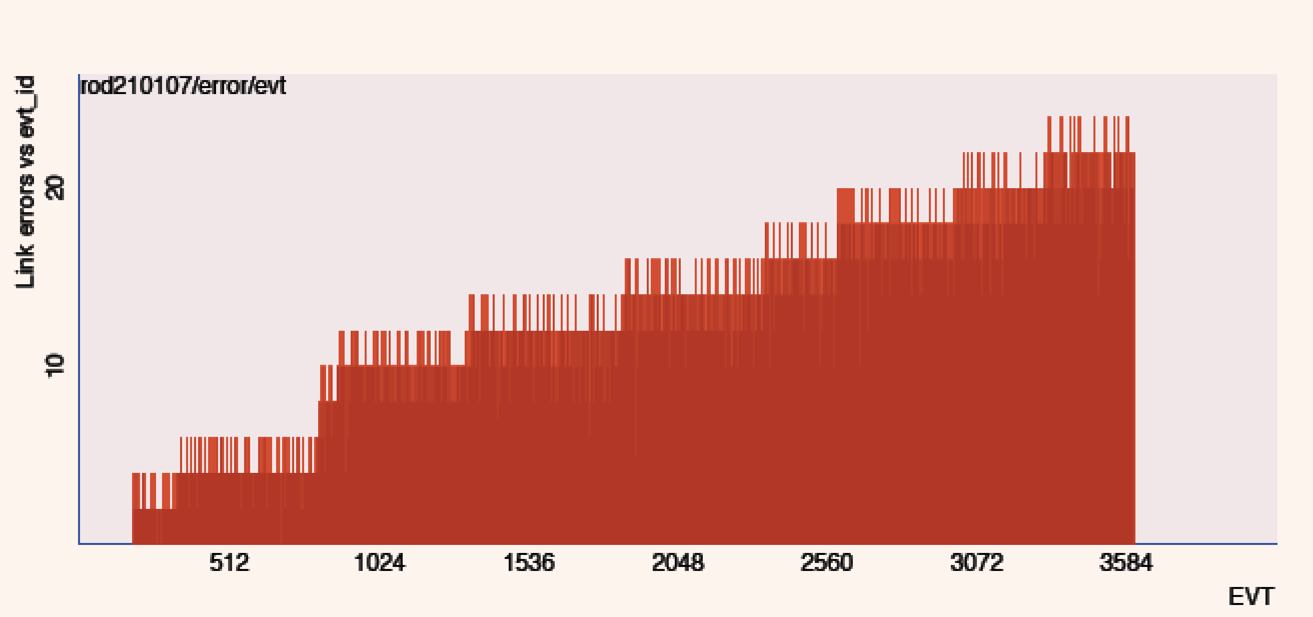
- Certain to be ready for first LHC data!

SCT Monitoring - GNAM

- Samples data fragments at ROS's

  ⇒ looks at errors, hits, hit coincidences, plots maps and trends
  ⇒ puts histograms in Online Histogram Server
  ⇒ these can be displayed using the OHP, or the custom SctRodDaq GUI

- Provides fast, basic feedback- very quick response to detector problems



- This recent example shows ROD error rates being plotted per event (there is a problem here!)

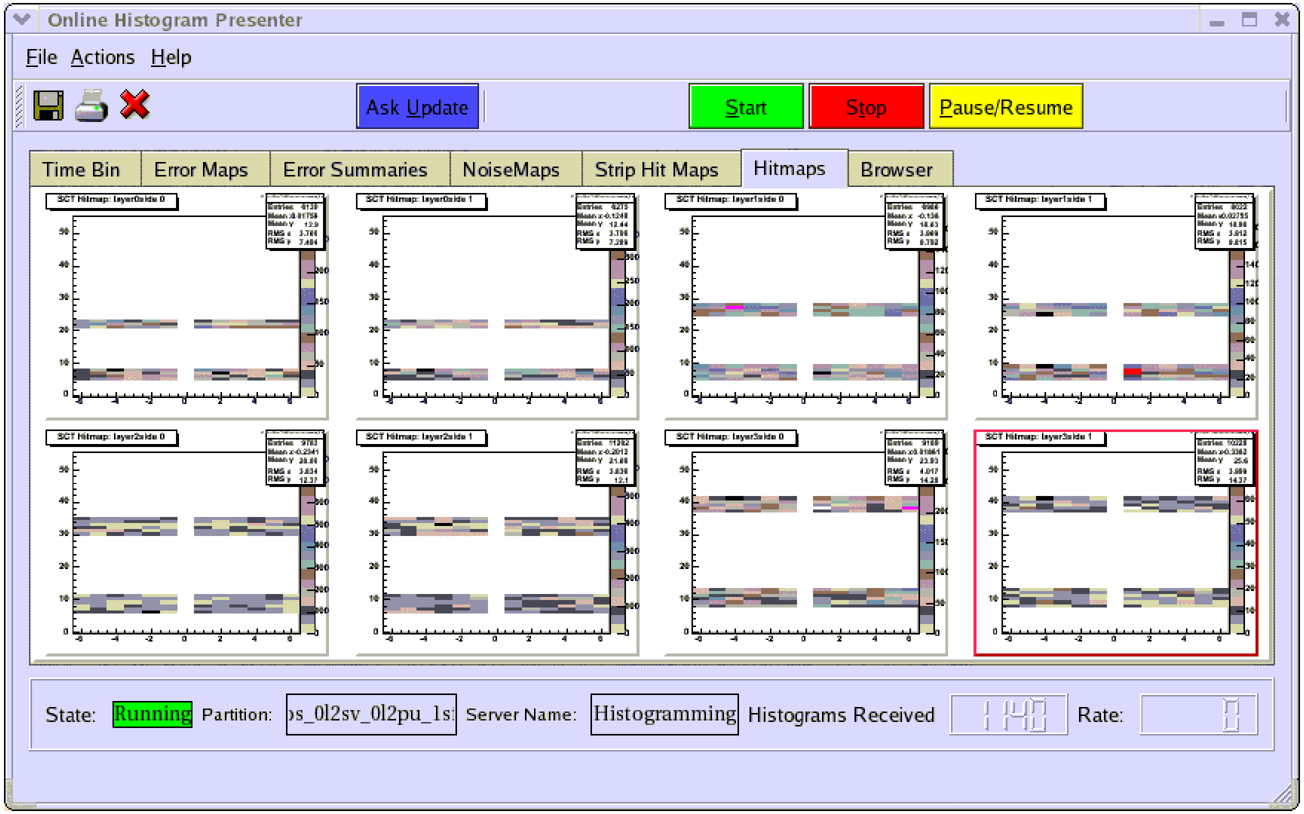
- Have also used the tool to time in the detector

- In progress:

  ⇒ rolling trends (update intuitively over a run)
  ⇒ interface with DQM software for automatic checking

SCT Monitoring – Athena

- Samples events at SFIs
  - ⇒ uses full event data (hits, clusters, tracks, etc)
  - ⇒ puts histograms in Online Histogram Server, retrieved by OHP
  - ⇒ rate is ~1 Hz due to tracking software (though can run in stripped down mode)
  - ⇒ run offline for more detailed checks, and to log problems in an offline database
- Have extended functionality of online Athena monitoring (see next slide)



State: Running Partition: ›s_0l2sv_0l2pu_1s Server Name: Histogramming Histograms Received 1140 Rate: 0

- Monitors:
  - ⇒ hit occupancy
  - ⇒ noise occupancy
  - ⇒ errors
  - ⇒ efficiencies
  - ⇒ track parameters (residuals, pulls, track χ^2)
- Is already interfaced with DQM framework for automatic checking
- Has performed well in tests
- In progress:
  - ⇒ time dependent quantities (hits vs time for each wafer)
  - ⇒ more intuitive online displays for shifters

SCT Monitoring – Athena (extra functionality)

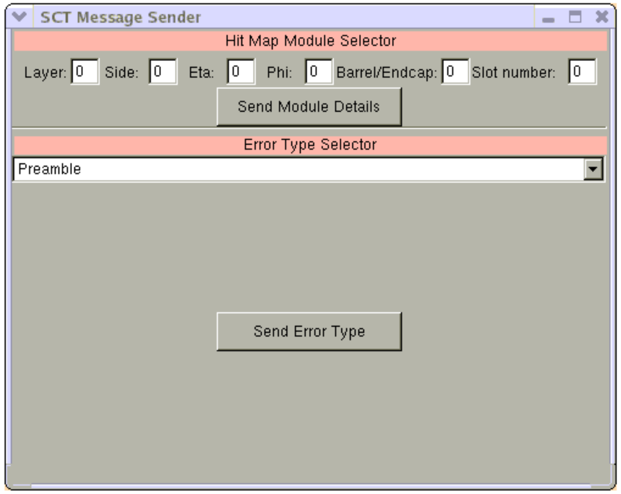
- The problem with the Athena monitoring system is that histograms need to be booked at the start of the run

  ⇒ can choose which get sent to the server, but can't add histograms in real time
  ⇒ if too many histograms get sent to the server it hangs under the strain

- For the SCT this means we can't look at each wafer online (there are over 6000 wafers)

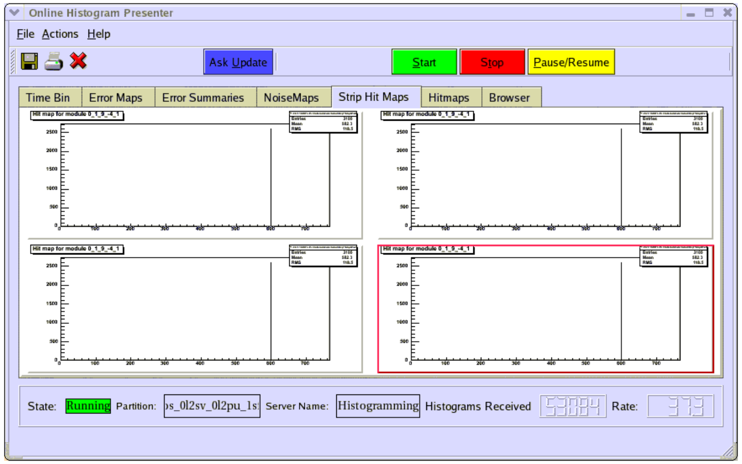
- Have got round this by sending 'template' histograms to the server and filling them dynamically

  ⇒ means we have online monitoring of wafers at the strip level
  ⇒ can also choose which type of errors to display
  ⇒ can choose to look at modules in detail if a problem is suspected
  ⇒ this is being made automatic (i.e. dodgy wafers will automatically be displayed to shifters)



Layer: 0 Side: 0 Eta: 0 Phi: 0 Barrel/Endcap: 0 Slot number: 0

Preamble

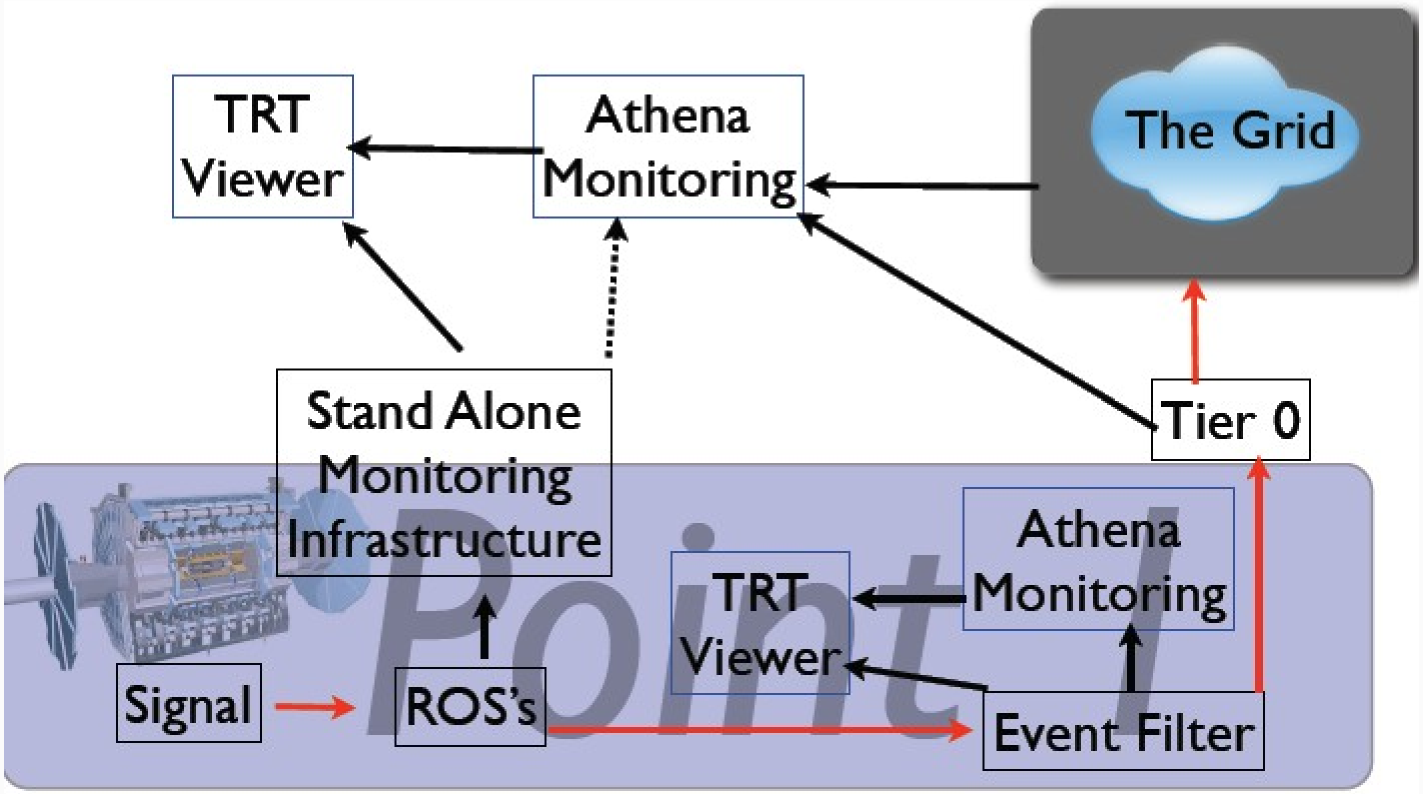


State: Running Partition: ɔs_0l2sv_0l2pu_1s Server Name: Histogramming Histograms Received 53084 Rate: 37.3

TRT Monitoring – Overview

- TRT uses a combination of Athena and a custom tool (TRTViewer)
- TRTViewer can either:

  ⇒ display output of Athena monitoring (more sophisticated than OHP)
  ⇒ display histograms from the ROS level

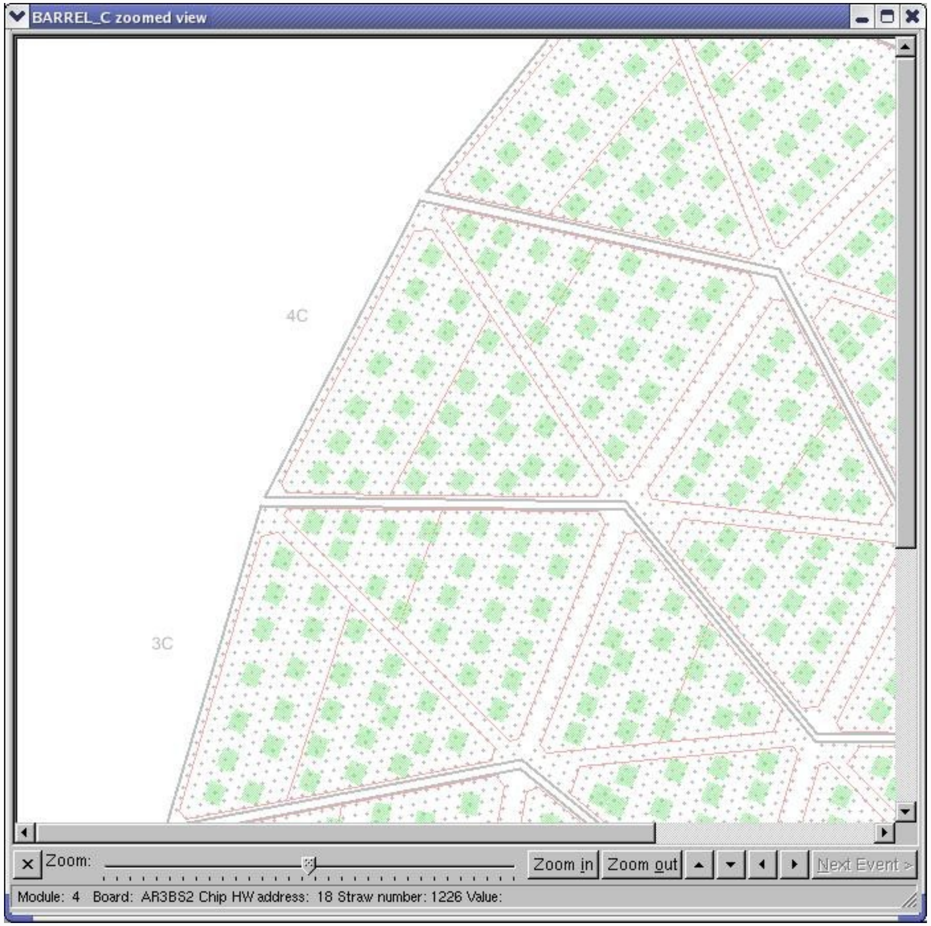


- Histograms:

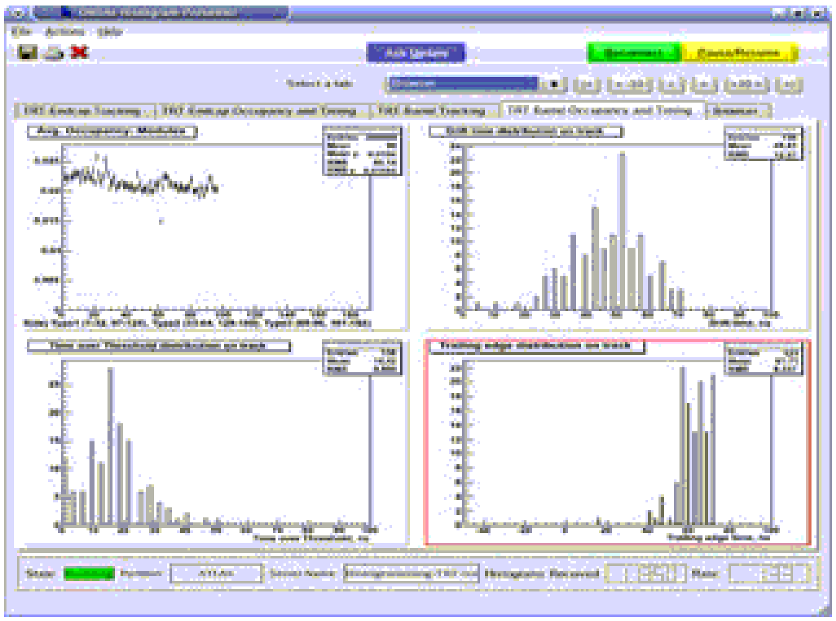
  ⇒ LL, HL occupancy
  ⇒ average trailing edge
  ⇒ average drift time
  ⇒ trends
  ⇒ number of tracks
  ⇒ track residuals
  ⇒ plots vs phi angle

- Interfaced with DQM for automatic checking
- Everything is essentially in place and tested
- In final stages of debugging for a smooth running system

TRT Monitoring – Screenshots



Module: 4 Board: AR3BS2 Chip HW address: 18 Straw number: 1226 Value:


File
Actions
Help
Ask Update
TRT Endcap Tracking
TRT Endcap Occupancy and Timing
TRT Barrel Tracking
TRT Barrel Occupancy and Timing
Avg. Occupancy: Modules
Drift time distribution on track
Time over Threshold distribution on track
Trailing edge distribution on track
State
ATLAS
Histograms Received
Rate

ID Global Monitoring

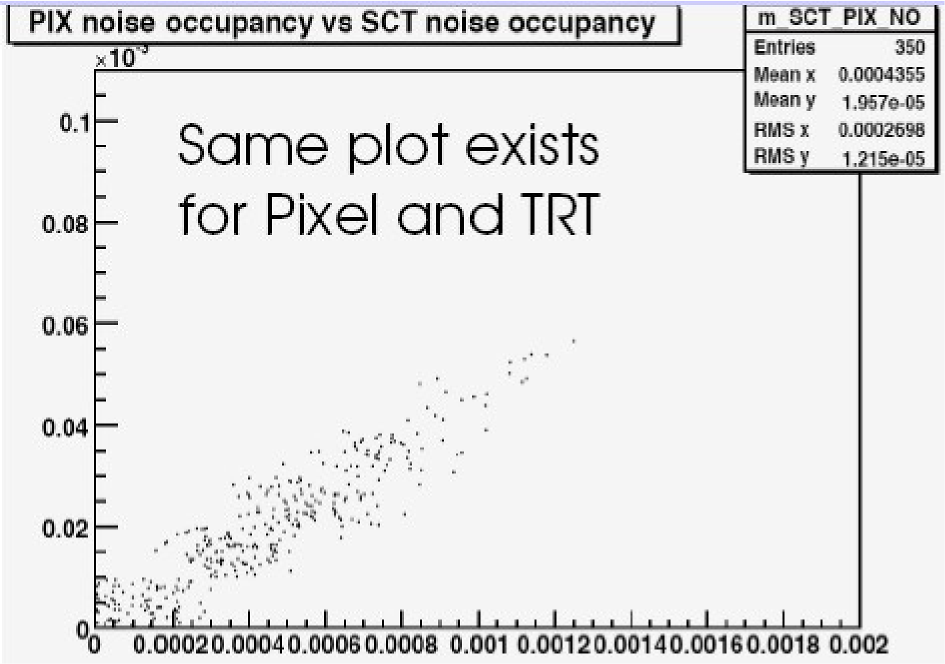
- Written in Athena
  - ⇒ samples events from SFI
  - ⇒ can also run offline
  - ⇒ anything that requires more than 2 subdetectors is monitored by the ID global tool

- Looks at:
  - ⇒ synchronization between detectors (bunch crossing id, level 1 trigger info)
  - ⇒ correlation of occupancy between detectors
  - ⇒ hits, track parameters and residuals
  - ⇒ angles between track segments for the different detectors

- Has been tested in both the recent DAQ integration runs and the Full Dress Rehearsals

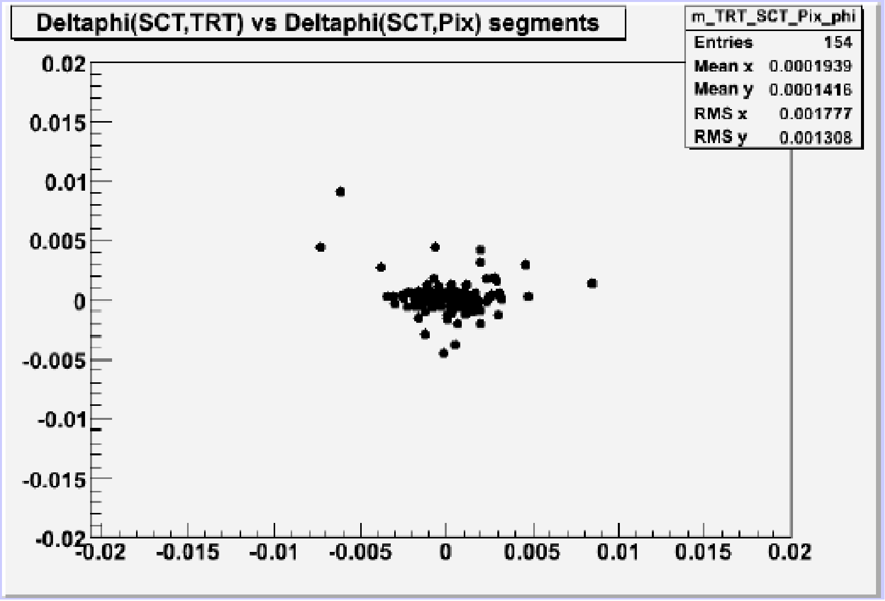
- Have several different tools for different purposes:
  - ⇒ alignment monitoring focussing on hit efficiencies, residuals and track properties
  - ⇒ beam spot monitoring – compares derived beam spot with reconstructed vertex positions
  - ⇒ performance monitoring – looks at some classic resonances (mass, width and yield)

ID Global Monitoring

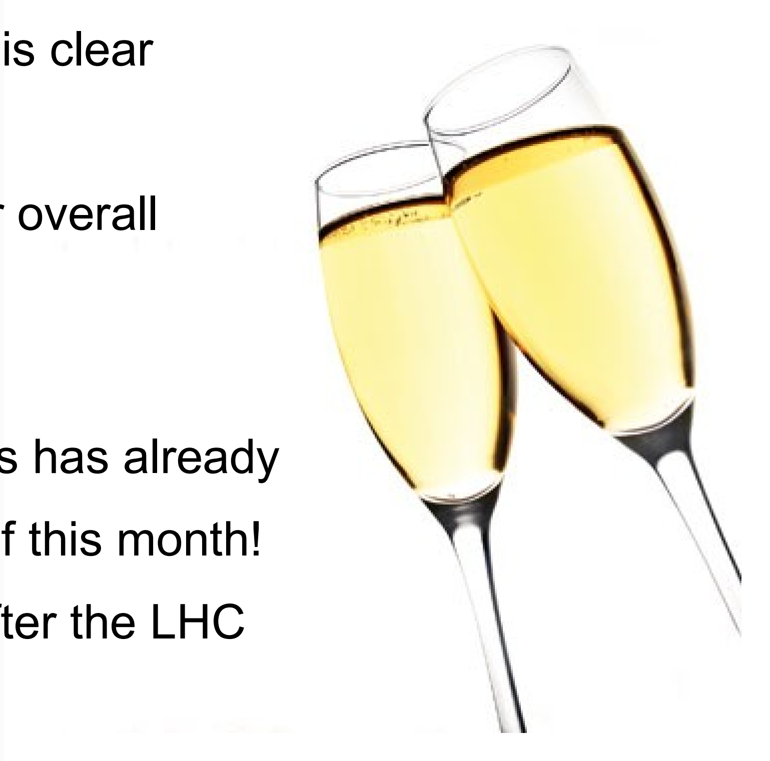


Same plot exists for Pixel and TRT

$\Delta\phi$ between TRT, SCT and Pixel segments



Summary



- The need for high quality monitoring of the ATLAS detector is clear

  ⇒ each subdetector has a dedicated monitoring team
  ⇒ each subdetector has more than one system
  ⇒ each has a slightly different implementation but similar overall strategy

- The packages are moving towards their final form:

  ⇒ need to make online systems as intuitive as possible
  ⇒ need to structure a shift program and train shifters (this has already started)
  ⇒ need to be ready to run 24 hours a day from the end of this month!

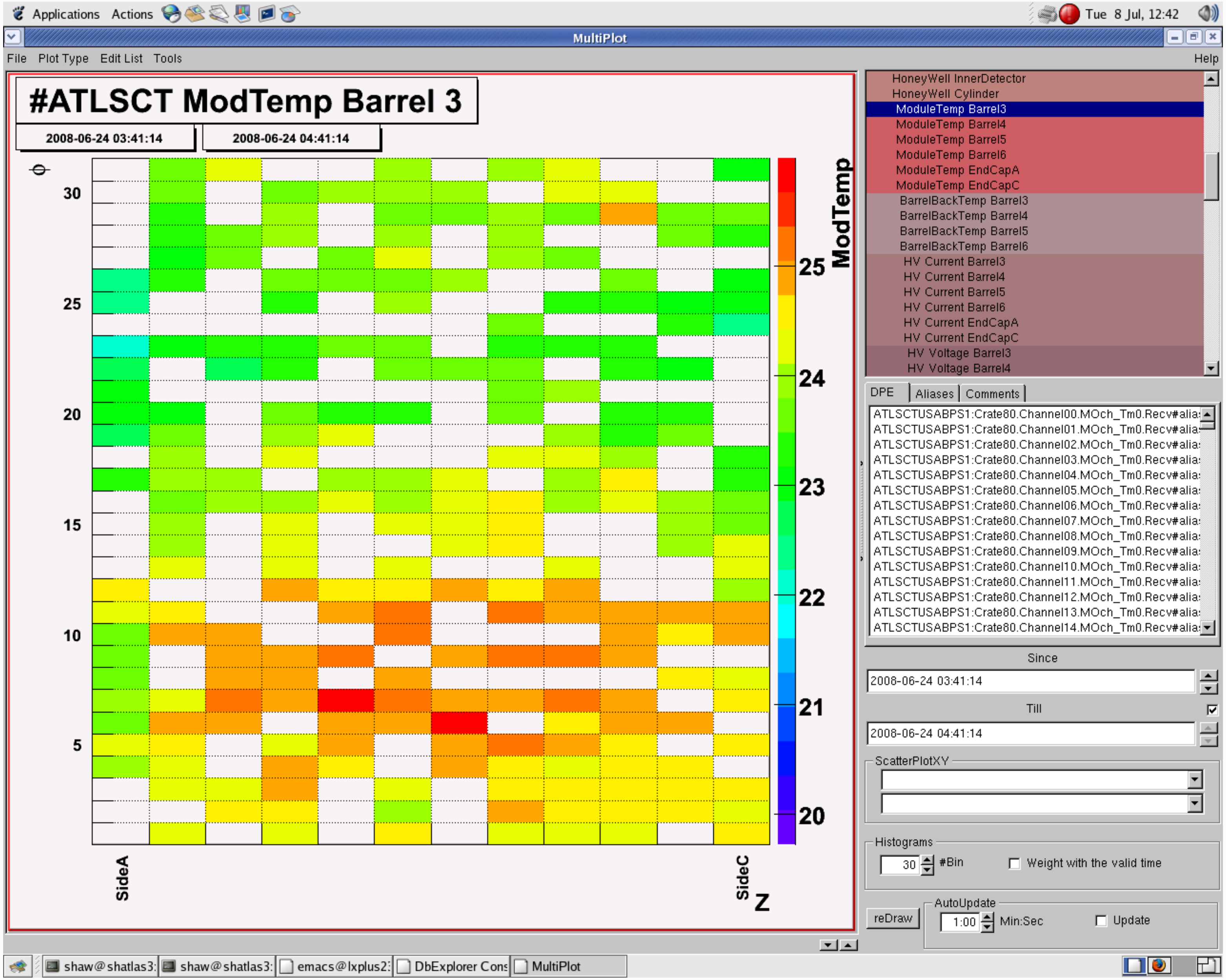
- Development/improvements can be expected to continue after the LHC turns on

  ⇒ all running so far has used only parts of each subsystem and/or a simulator
  ⇒ beam data may well contain some surprises
  ⇒ preparation with MC data is important and is being done

- Exciting times lie ahead!

Backup

DCS monitoring



HoneyWell InnerDetector
HoneyWell Cylinder
ModuleTemp Barrel3
ModuleTemp Barrel4
ModuleTemp Barrel5
ModuleTemp Barrel6
ModuleTemp EndCapA
ModuleTemp EndCapC
BarrelBackTemp Barrel3
BarrelBackTemp Barrel4
BarrelBackTemp Barrel5
BarrelBackTemp Barrel6
HV Current Barrel3
HV Current Barrel4
HV Current Barrel5
HV Current Barrel6
HV Current EndCapA
HV Current EndCapC
HV Voltage Barrel3
HV Voltage Barrel4

```
ATLSCTUSABPS1:Crate80.Channel00.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel01.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel02.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel03.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel04.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel05.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel06.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel07.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel08.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel09.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel10.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel11.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel12.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel13.MOch_Tm0.Recv#alia:
ATLSCTUSABPS1:Crate80.Channel14.MOch_Tm0.Recv#alia:
```

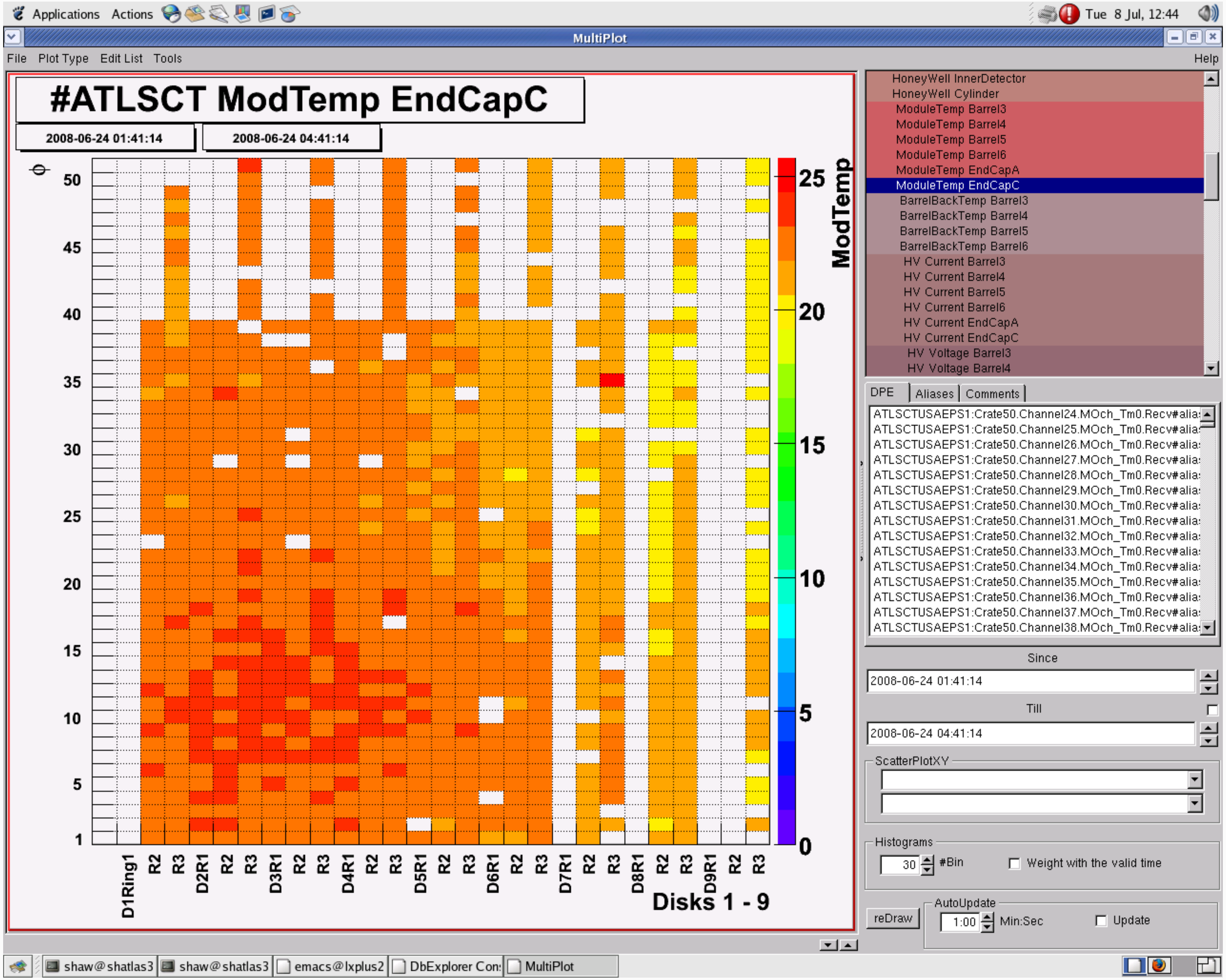
Since: 2008-06-24 03:41:14

Till ☑: 2008-06-24 04:41:14

30 #Bin ☐ Weight with the valid time

AutoUpdate: 1:00 Min:Sec ☐ Update

DCS monitoring



#ATLSCT ModTemp EndCapC

2008-06-24 01:41:14

2008-06-24 04:41:14

HoneyWell InnerDetector
HoneyWell Cylinder
ModuleTemp Barrel3
ModuleTemp Barrel4
ModuleTemp Barrel5
ModuleTemp Barrel6
ModuleTemp EndCapA
ModuleTemp EndCapC
BarrelBackTemp Barrel3
BarrelBackTemp Barrel4
BarrelBackTemp Barrel5
BarrelBackTemp Barrel6
HV Current Barrel3
HV Current Barrel4
HV Current Barrel5
HV Current Barrel6
HV Current EndCapA
HV Current EndCapC
HV Voltage Barrel3
HV Voltage Barrel4

```
ATLSCTUSAEPS1:Crate50.Channel24.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel25.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel26.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel27.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel28.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel29.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel30.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel31.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel32.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel33.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel34.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel35.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel36.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel37.MOch_Tm0.Recv#alia:
ATLSCTUSAEPS1:Crate50.Channel38.MOch_Tm0.Recv#alia:
```

Since: 2008-06-24 01:41:14

Till □: 2008-06-24 04:41:14

ScatterPlotXY

Histograms: 30 #Bin □ Weight with the valid time

AutoUpdate: 1:00 Min:Sec □ Update

Martin White – Cambridge ATLAS