



SDD Quality Assurance (& DQM)

P. Cerello, W. Ferrarese, M. Siciliano

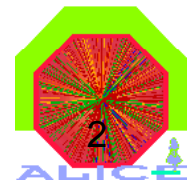
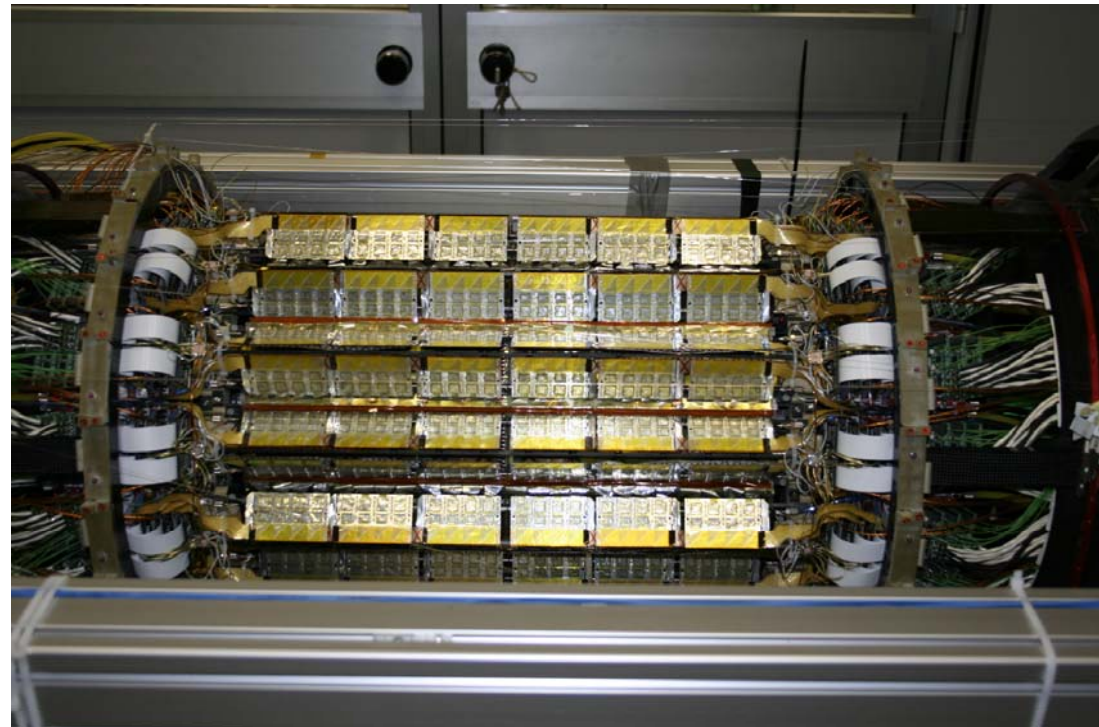
ALICE OFFLINE WEEK - October 2007





Summary

- Offline QA
 - ⇒ Goals
 - ⇒ Approach
 - ⇒ Existing code
- Online DQM
 - ⇒ Plans
- What's Next...





Offline QA

- **Goals**

- ⇒ Provide an assessment of the overall data quality by analysing all the events

- ✓ *Format/meaning to be defined in detail*

- **Approach**

- ⇒ Analyse RAW SDD Data being taken at Point2

- ✓ *Debugging of the Data Decoding (see Melinda's talk last Monday)*

- ✓ *Assessment of the Decoding Speed*

- ✓ *Feedback to the HW and DAQ SDD teams*

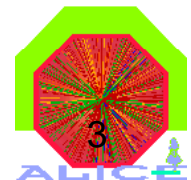
- ⇒ Build some QA-like distributions on real Data

- ✓ *8 SDD modules, No zero suppression*

- ✓ *Event size equivalent to 3% occupancy, comparable to central Pb-Pb*

- ✓ *Baseline & Noise*

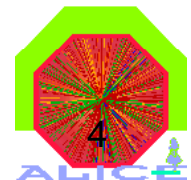
- ✓ *Injectors*





Offline QA

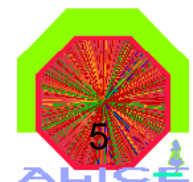
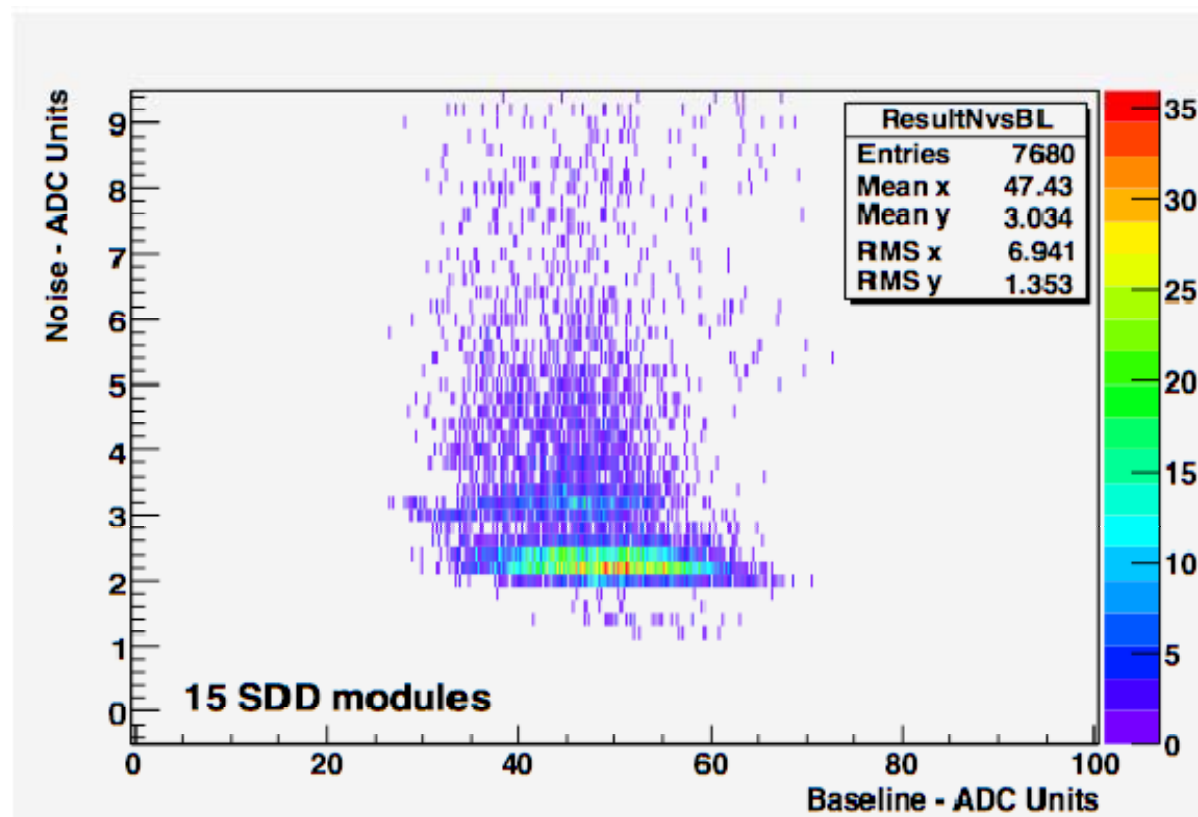
- Debugging of the Data Decoding
 - ⇒ Several bugs found in decoding SW, SDD DAQ, one in DAQ
 - ⇒ SDD buffer decoding under control now
- Assessment of the Decoding Speed
 - ⇒ Not satisfactory
 - ✓ *3% occupancy, about 2Hz*
 - ✓ *Where is the bottleneck?*
 - ✓ *It may be the buffer structure, SDD DAQ team involved*





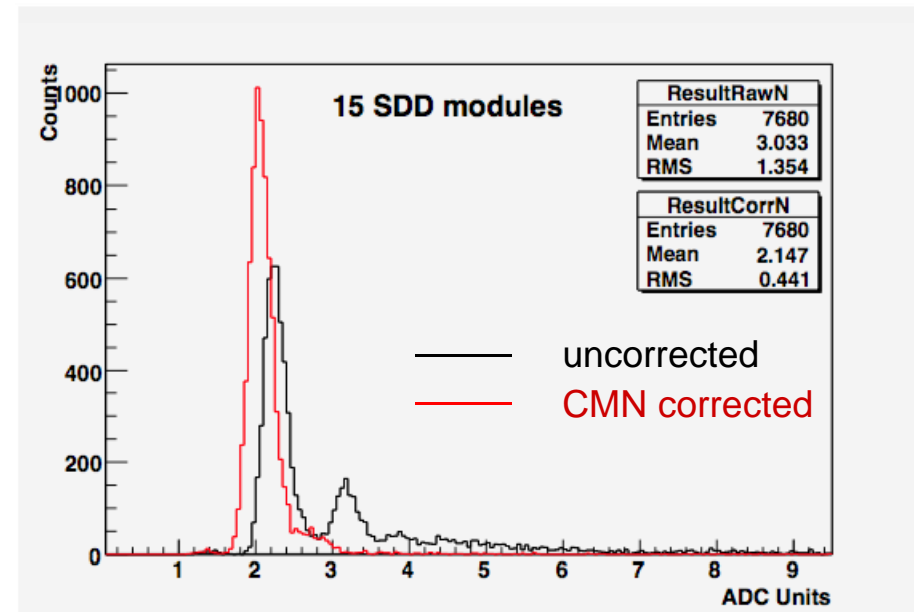
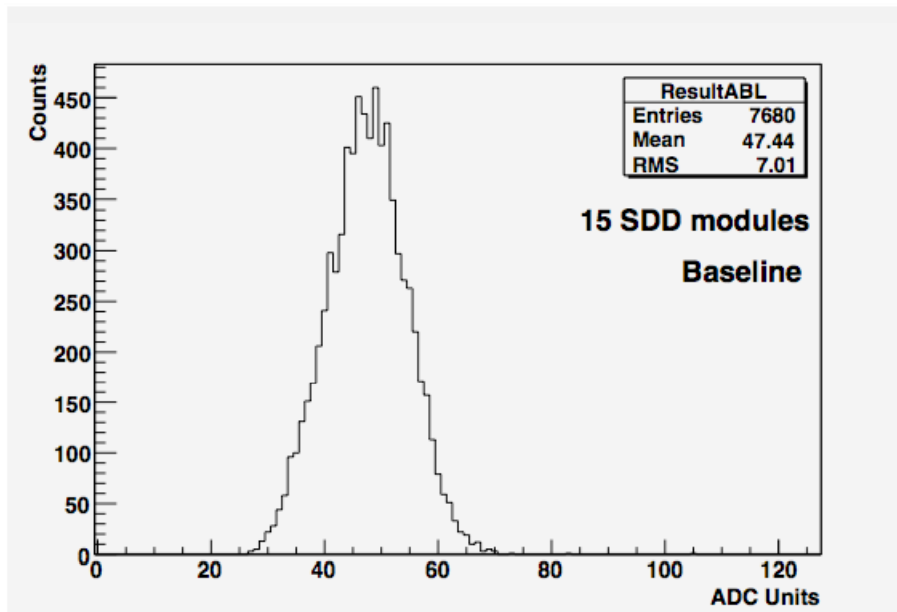
Offline QA

- Feedback to the HW and DAQ SDD teams
 - ⇒ Some SDD DAQ control words must be changed
 - ⇒ Preliminary Analysis of baseline, noise



Offline QA

- Feedback to the HW and DAQ SDD teams
 - ⇒ Preliminary Analysis of baseline, noise
 - ⇒ Pressure from HW team to analyse last week data
 - ✓ *First bunch with 20% SDDs ON*
 - ⇒ These are also DQM/QA distributions





Offline QA

- Existing code

 - ⇒ A "Maker" and a "Checker"

 - ⇒ RAW Data read & analysed from macros:

 - ✓ *ITSSDDQAMaker.C*

 - ✓ *ITSSDDQAChecker.C*

 - ⇒ Aliroot classes

 - ✓ *AliITSSDDQAMaker*

 - ✓ *AliITSSDDQAChecker*

 - Ready to be filled with histograms





Offline QA

- Preliminary Control List

⇒ *Digit (module, anode, time, charge) Level - Detector Performance*

- Baselines (-> dead channels)
- Noise
- Injectors
- Layer 1/Layer 2 entries

- ✓ *Layer Level*

- N_entries vs Ladder Number

- ✓ *Ladder Level*

- N_entries vs Detector Number

- ✓ *Detector Level*

- Patterns (N_entries vs. Anode, Time Bin)
- Q_Average vs Anode
- Q_Average vs Time
- ...





Offline QA

- Preliminary Control List

⇒ *RecPoint/Cluster (module, X, Y) Level - Reconstruction Validation (...not for ONLINE...)*

- Layer 1/Layer 2 entries

- ✓ *Layer Level*

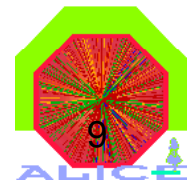
- N_entries vs Ladder Number

- ✓ *Ladder Level*

- N_entries vs Detector Number

- ✓ *Detector Level*

- Patterns (N_entries vs. X, Y)
- Q_Average vs X
- Q_Average vs Y
- ...

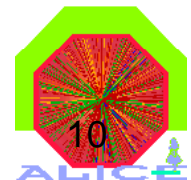




What's Next?

- OFFLINE QA

- ⇒ Create & Fill histograms in AliRoot Maker class
- ⇒ Add the calls so as to fill histograms in the reconstruction flow
 - ✓ *In order not to read RAW DATA twice, it should go inside the ClusterFinder function*
- ⇒ Write a proto-QA Checker & test it
 - ✓ *on non-zero-suppressed real data*
 - ✓ *on simulated data*
 - ✓ *asap on Cosmic Rays Data*
- ⇒ Provide Checker output in the required common format
- ⇒ Get (& use) information on Dead Channels, expected Baseline, Noise etc. from OCDB





What's Next?

- **ONLINE QA**

- ⇒ Meeting last Tuesday with Sandra, Filimon, Pierre

- ✓ *Agreement on how to proceed in case of problems: expert custom analysis will start from RAW DATA*

- ⇒ So, almost Same Goals of Offline-QA

- ✓ *The same (or very similar) set of histograms will provide feedback on the detector performance*

- ⇒ Install AMORE and learn how to use it to generate information to be published

- ✓ *Short term goal: publish one distribution*

- ✓ *... then add the others...*

- ⇒ Can we use the very same AliRoot code?

- ✓ *If yes, use it*

- ✓ *If not, we'll have to "adapt and duplicate it"*

