AliReconstruction: Future Plans

C. Cheshkov & J. Belikov

9/10/2007

ALICE Offline Week

Recent developments

- □ Extraction of the event loop from the detectors code to AliReconstruction
- □ Removal of run-loaders from the detector reconstruction code
- □ Cleaning of various ‘AliReconstructor’ methods:
  - 4 Reconstruct methods -> 2 Reconstruct method (with TTree interface to digits and clusters and AliRawReader for raw-data access)
  - 4 FillESD methods -> 2 FillESD methods (we can go to only 1 method if we drop the access to digits/raw data, currently it is being used by PHOS)

Open issues and plans

- Memory consumption (see Marian's talk):
  - We need to improve drastically the memory usage!
  - Correct the remaining memory leaks
  - Re-organization/optimization of the run-loaders (P. Hristov) and data containers?

Open issues and plans

- CPU time:
  - p-p
    - HMPID cluster unfolding (takes about ¼ of the CPU time)
    - MUON local reconstruction
  - Pb-Pb
    - Stand-alone ITS tracker
    - Vertexers (both ITS and ESD tracks)

Open issues and plans

- □ Technical: removal of run-loaders from the vertexers (re-organization of the interfaces)
- □ Introduction of additional vertex finders: (TPC-only, T0, ...)
- □ Array of vertices in ESD?

Open issues and plans

- Event selection (possibility to reconstruct event subset according to):
  - Trigger mask
  - Event type (DAQ)
  - List of active/validated by DAQ QA detectors
  - “Timestamp” (bc#, orbit id, period)
  - …

Open issues and plans

- Removal of dependencies between sub-detector local reconstructions

Open issues and plans

- HLT reconstructor:
  - HLT reconstruction chain has been moved to simulation
  - We need now reader/reconstructor of the HLT “raw-data” (clusters, tracks etc.)

Conclusions

- □ Collection of additional requirements
- □ Identification of detector-specific and core-offline tasks -> offline planning tool (description, dead-lines, responsibles)