Flat ESD status

M.Krzewicki

FIAS

Flat ESDs

- Rationale:
  - online processing on the HLT (or any online system).
    - means sending/receiving objects.
  - ROOT (de-)serialisation slow (dominates processing time in most cases).
  - ➡ Keep everything in a contiguous (flat) buffer to eliminate (un-)packing overhead.

- Constraints:
  - Same code running offline and online.
    - API consistent between offline ESDs and flat ESDs….
    - …. with very few exceptions -> offline code needs to be ported a bit.
    - porting easy -> few getters different (fill vs. return pointer), new getters work fine in offline.

Flat ESDs

- Main idea: KISS.
  - but also as much backward compatible as possible.

- Ideally:
  - The base classes would be pure interface classes.
  - Interface would make no assumptions on underlying data model.

- API:
  - AliFlatESDevent: AliVEvent
  - AliFlatESDTrack: AliVTrack
  - etc...
  - in principle all offline analysis code (AliAnalysisTask-like) can be supported if internally data is accessed via base classes.

Flat ESDs

- some caveats:
  - virtual functions -> virtual function table: needs reinit after receiving (instead of deserialisation) implements AliFlat*::Reinitialize();
  - short inheritance chains, e.g. track does not inherit from anything but the API base class - IS not AliExternalTrackParam, instead contains it.
  - special "blessed" getters: fill an object instead of getting a pointer to it, e.g:

```
const AliExternalTrackParam * trackIn = track->GetInnerParam();
```

becomes:

```
AliExternalTrackParam trckIn;
track->GetTrackParamIp(trckIn);
```

Flat ESDs - use cases.

- TPC online calibration
  - Calibration code ported and shown to produce results identical to current offline (Munich, see presentations by I.Vorobyev).

- Online TPC QA - work in progress (Houston, Ant Timmins et al.).

- Beam-gas background monitoring using VZERO:tracking correlations.
  - work has started (Alex Borissov et al.).

Performance of flat ESDs (on the HLT)

Processing event with 100 tracks on AMD Opteron(TM) Processor 6238, 2.9GHz.

(all the numbers scale linearly with N tracks, for events with 0-300 tracks)

	Create time, ms	Serialize, ms	Deserialize,ms	Data Size, kB
ESD	19	7	3	43
ESD Friend	110	90	55	300
FlatESD	1	0	0	45
FlatFriend	4	0	0	400

Time Calibration takes 350 ms on both ESD and Flat structures

Status

- Code ready, partly in master, more developments in feature-flatdev branch
  - many fixes +
  - VZERO container in the flat ESD

- This week (hopefully) a large scale test at GSI (full cpass0/cpass1 benchmark)
  - same input data (filtered raw from 2010, 2011, 2015).
  - different software: master and flatdev.

- many more developments in flatdev branch related to online calib:
  - ZMQ (for external communication: Event display, internal use for the feedback loop)
  - need to merge this into master as soon as it is validated (See above).

scope of changes for the base offline

- M STEER/AOD/AliAODTrack.h
- M STEER/ESD/AliESDEvent.h
- M STEER/ESD/AliESDVZERO.h
- M STEER/ESD/AliESDfriendTrack.cxx
- M STEER/ESD/AliESDfriendTrack.h
- M STEER/ESD/AliESDtrack.cxx
- M STEER/ESD/AliESDtrack.h
- D STEER/STEER/AliHLTTestInputHandler.cxx
- D STEER/STEER/AliHLTTestInputHandler.h
- A STEER/STEER/AliHLTVEventInputHandler.cxx
- A STEER/STEER/AliHLTVEventInputHandler.h
- M STEER/STEER/CMakeLists.txt
- M STEER/STEER/STEERLinkDef.h
- M STEER/STEERBase/AliSysInfo.cxx
- M STEER/STEERBase/AliSysInfo.h
- M STEER/STEERBase/AliVEvent.h
- M STEER/STEERBase/AliVTPCseed.h
- M STEER/STEERBase/AliVTrack.h
- M STEER/STEERBase/AliVfriendEvent.h
- M STEER/STEERBase/AliVfriendTrack.h
- M STEER/STEERBase/TTreeStream.cxx
- M STEER/STEERBase/TTreeStream.h

- M ANALYSIS/ANALYSIS/AliAnalysisManager.cxx
- M ANALYSIS/ANALYSIS/AliAnalysisTask.h