Andrei Gheata, ALICE offline week,

ANALYSIS FRAMEWORK - STATUS

Analysis output files

- Outputs validation improved
  - Triggered by the 'unzip error' .
  - Output files that can be just partially recovered are not good for AOD analysis.
  - The analysis manager checks if the files were can be properly opened. It writes a 0-length file that can be used for validation of jobs
- Folders in the output files
  - mgr->CreateContainer("comparison", TList::Class(), AliAnalysisManager::kOutputContainer, "myAnalysisHistos.root#comparison")
    - Will make the folder "comparison" in the output file and put results there
    - Available soon

Additions to the AliEn plugin

- Names of files generated by the alien plugin can (and should) be changed to avoid clashes
- Output directory can be an absolute alien path
- New method added to support raw run numbers (like 000087654 3)
  - AddRunNumber(const char *number)
- New method allowing to customize the command executed to run the job
  - Default: 'root –b –q'
  - 'aliroot –b –q' needed by PWG1 QA train

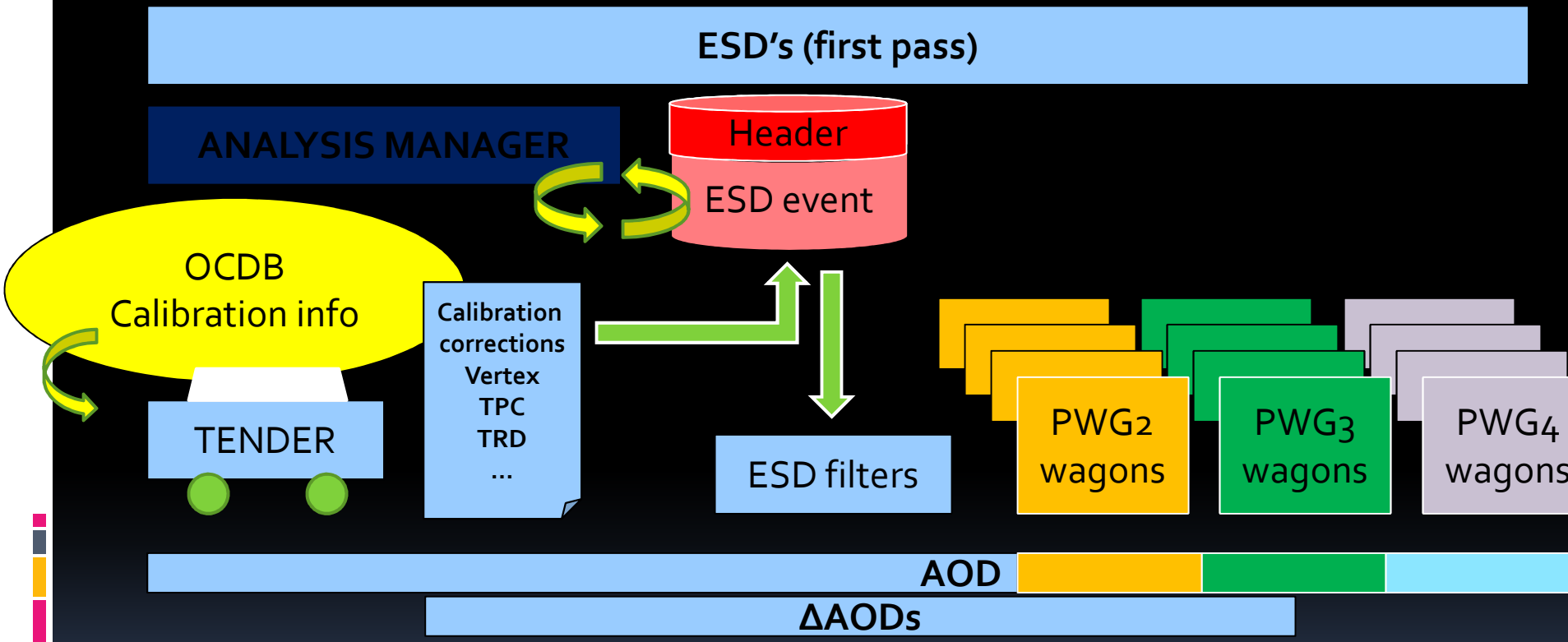
Bottlenecks in file access, memory problems

- We noticed that most failures in grid analysis for IO intensive jobs is due to timeouts
- No simple tools to make some systematic on the conditions that trigger this behavior
  - Started to make some based on xrootd environment variables for timeouts
- Memory usage plots per task will be produced on demand in "test" mode
  - So that everybody can debug leaks

Analysis tender

- Agreed proposal discussed last offline week
  - A set of detector specific algorithms attached to a tender wagon
- Read-only access to OCDB, to get the 'latest and best' set of calibration data and correct the current ESD event in memory with updated info (TPC and TRD PID, primary vertex, …)
- Do NOT create a new ESD, but just make the updated info directly available to the analysis cars after the tender
  - Probably produce a new set of AOD's to avoid running the tender too often

Workflow



- Only the tender accesses OCDB
  - Info used by detector expert algorithm updating the current ESD event in memory
  - The other wagons normally running (as without tender)
  - If writing AOD's, they will automatically contain the corrections

Tender implementation

- Basic. Two classes.
- AliAnalysisTender: public AliAnalysisTask
  - Has pointers to: current ESD event, CDB manager, CDB key (!), list of supplies (!)
  - ConnectInputData(): Unlocks CDB manager, set default storage and run number, allow supplies (!) to set the specific storages and read from CDB, lock CDB
  - Exec(): Call ProcessEvent() for all tender supplies
- AliAnalysisTenderSupply : public TNamed: the detector algorithm to run in the tender
  - 2 methodes to implement:
  - Init() to initialize CDB access and to read relevant CDB info
  - ProcessEvent() to run the algorithm and patch the ESD event in memory