Andrei Gheata, ALICE offline week,

ANALYSIS FRAMEWORK - STATUS

Added since the last meeting

- Creation of delta AOD in the same session producing the main AOD
  - AliAnalysisTaskSE::AddAODBranch(cname, addobj, filename) – will add this branch in a file
- Added support for copying the outputs of the analysis in an AliEn folder from a PROOF job.
  - AliAnalysisManager::SetSpecialOutputLocation(“ alien:///path_to_my_directory”)

Enhancements of the AliEn plugin

- Use as output files the outputs of the tasks connected to the analysis manager
  - SetDefaultOutputs(Bool_t flag)
- Exclude some outputs from merging (e.g. AliAOD.root)
  - SetMergeExcludes(const char* list)
- Adjust the number of files used in the "test" case
  - SetNtestFiles(Int_t nfiles)

Other features

- Possibility to use par packages (not encouraged) and to set the include path on the remote site
  - EnablePackage(const char *package)
  - AddIncludePath(const char *path)
- Possibility to select SE for output
  - SetPreferedSE(const char *se)
- Upgrade of the generated validation script (A.Maire)

Yet other features

- Support for processing delta AOD files via the plugin (A.Dainese)
  - SetFriendChainName(const char*name)
- Support processing of run ranges within a production
  - SetRunRange(Int_t min, Int_t max)
- Possibility to process a full directory (without specifying run numbers or ranges)
  - Avoid the problem of folder names like 00012345

Merging improvements

- Possibility to merge in chunks of files and one can specify the number of files in a chunk
  - SetMaxMergeFiles(Int_t nfiles)
- Possibility to resume merging from the last chunk
  - Currently merging done fully on the client
    - Advantage: full control for debugging
    - A possible improvement: merging per masterjob first
- A major leak when merging lists was fixed in the TFileMerger

Improvements for batch analysis

- Possibility to adjust number of runs per master job: SetNrunsPerMaster()
  - 1 xml collection per run; merged into a single “master” xml collection
- “Smart” automated submit procedure
  - Submit all jobs in a raw at production scale cannot be currently handled by AliEn
  - New submit policy implemented:
    - Submit first master job and wait until gets SPLIT
    - Estimate the number of sub-jobs per master and compute number of new jobs to submit next until getting at most 100 jobs WAITING
    - Wait until Nwaiting+Npermaster < 100 then submit next

Ongoing work

- Handle AOD + deltaAOD in PROOF analysis
  - Difficult unless we zip them together
  - Maybe wait for dataset functionality for associated files (CAF savannah #9148)
- Possibility to resume AF in PROOF
- Add fine granularity memory profiling per task
  - To get an idea of the allocation issues