

Status of the analysis framework

A.Gheata

ALICE offline week April'08

What was going on since the last report...

- Bug fixing – in the framework, but also in user code
 - Slowly approaching stability in spite of some erratic infrastructure behavior (grid & proof)
- Changes towards simplification of data access
- New developments related to AOD creation and management
- System info collected on demand
- Developments related to PROOF mode
- Restructuring of libraries related to analysis + a lot of testing of new user code



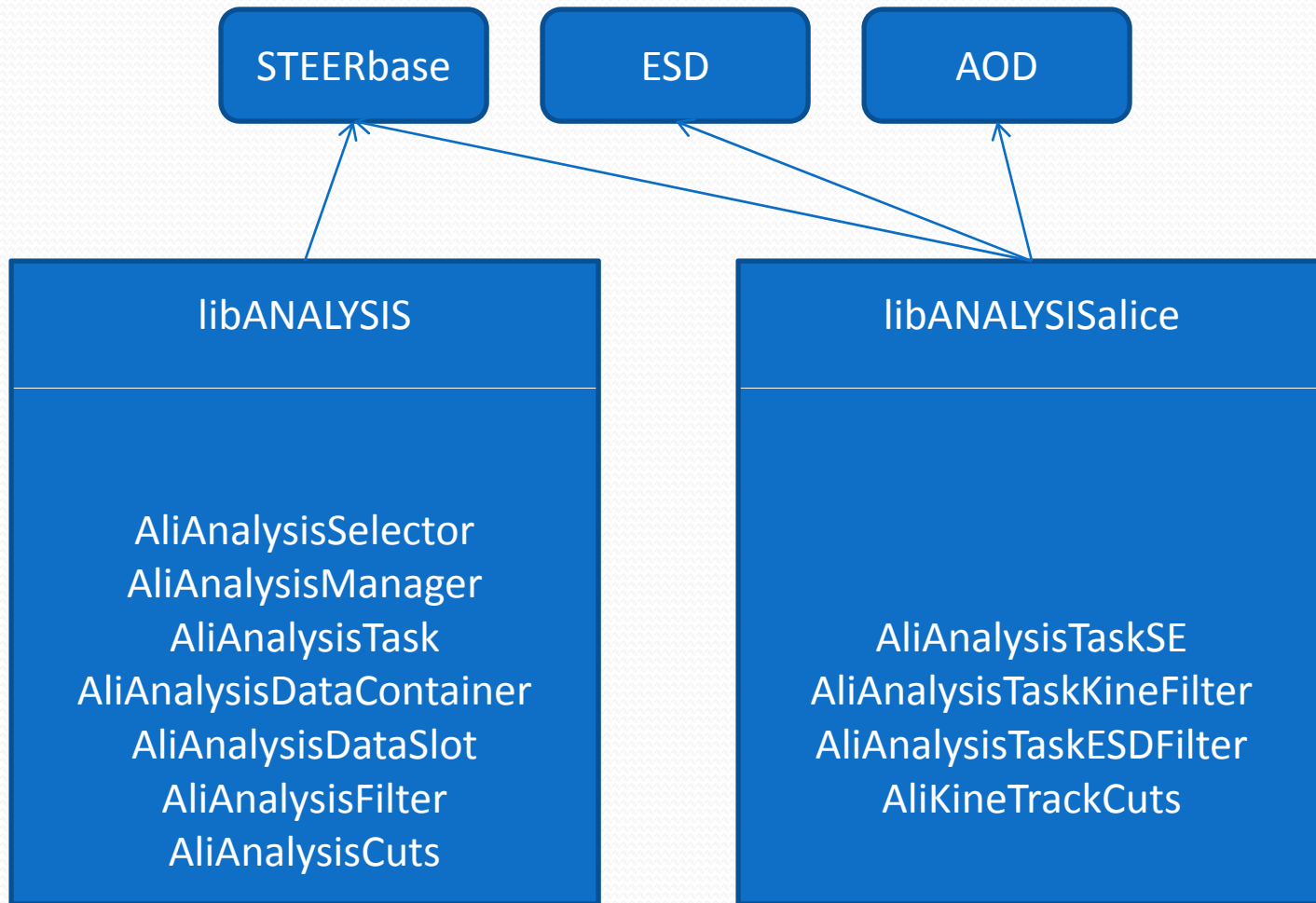
Access to data structures

- New class `AliAnalysisTaskSE` to derive from (instead of `AliAnalysisTask`)
 - To gain automatic access to input ESD or AOD event or to MC (no `ConnectInputData` needed in derived task)
 - To be able to add new user AOD branches using a simple interface
 - Make input and output data types more transparent to users
- `CreateOutputObjects()` -> `UserCreateOutputObjects()`
- `Exec()` -> `UserExec()`



Code restructuring

- Code dependent on data type moved in a separate library libANALYSISalice while the independent part stays in libANALYSIS
- Event filtering class based on a set of cuts introduced in the framework
- First implementation of kinematics and ESD filtering tasks in ANALYSISalice

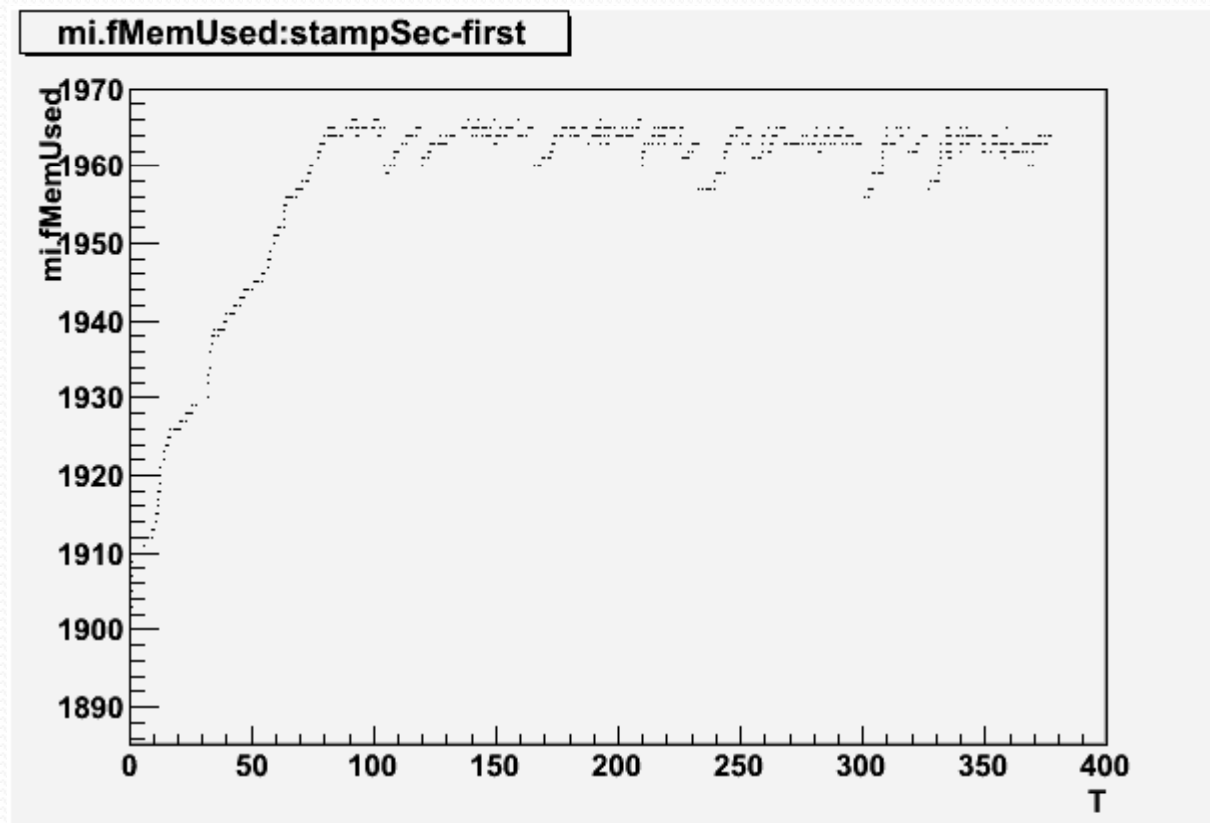




Memory checking

- **AliAnalysisManager::SetNSysInfo(Long64_t nevents)**
- Monitoring memory&timing each N events
 - Using AliSysInfo functionality to spot leaking tasks
 - Only in local mode
 - 2 output files: syswatch.log and syswatch.root
- Currently investigating how to improve monitoring per task

Memory usage profile vs time





PROOF-related add-ons

- Processing datasets instead of a chain

StartAnalysis(const char *type, const char *dataset,...)

- Saving files on proof workers

AliAnalysisDataContainer::SetSpecialOutput()

AliAnalysisManager::SetSpecialOutputLocation(const char* dir)

- Specified output files will be written to the special output location using TFile::Cp()
- If location not specified, special output are supposed to be **merged** by PROOF and registered or sent back to client – NOT working yet – investigating with PROOF team



Looked into

- Creation and management of AOD datasets (base AOD + add-ons – see talk of Andreas)
 - Work with several input trees
- Storing produced AOD's in PROOF mode
- Adding educated guess error messages when getting some pathological job behaviors in PROOF mode



Conclusions

- New developments related to making ALICE-specific input and output data types more generic and easy to handle
- AOD-based workflow figured out and started to be implemented in the framework
- Some fixes and memory checker introduced
- Steps towards handling AOD files produced in PROOF
 - To be continued with help from PROOF team
 - Still some black holes in PROOF merging
- More and more user tasks is a good sign...