

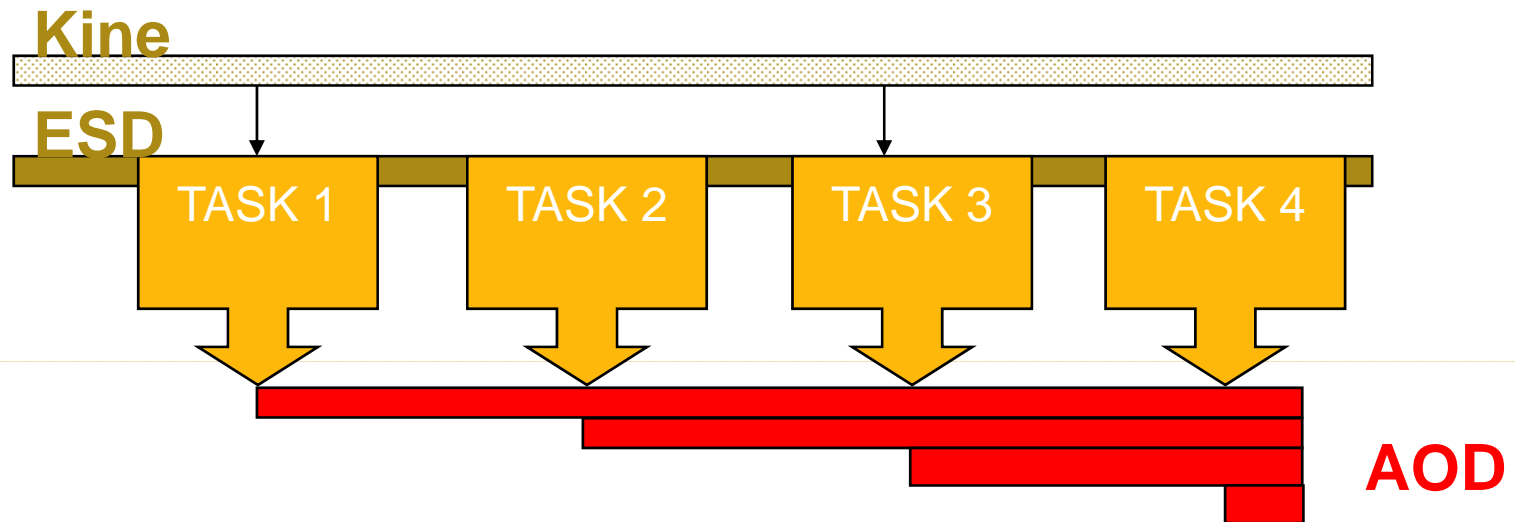


STATUS OF ANALYSIS TRAIN

Mihaela Gheata
ALICE offline week April'08



ANALYSIS TRAIN(S) MODEL



- Using the analysis framework to optimize organized analysis cycles in a production environment
 - The picture is becoming more complex (correction framework, AOD based analysis)
- The train is made of several independent tasks in the same *AliAnalysisManager* session (sharing the same CPU)
 - Using the same input data and sharing the same event loop
 - Having compatible I/O, comparable processing times and fitting existing memory resources



WHAT IS IN THE TRAIN

- Currently we are testing all available analysis tasks
 - Should run in local/proof/grid modes
 - Should fit in memory all together
- Tasks that will be tested in the train should be available in AliRoot
 - Due to the rate of framework and task development , the train is run with *.par* archives, but we will work with libraries in production mode
- The general packages used by the train are:
 - STEERbase, ESD, AOD (base classes and data structures)
 - ANALYSIS, ANALYSISalice (framework base classes)

AOD producer/updater tasks

- The output AOD tree creation/filling is managed in the framework by *AliAODHandler* class – mandatory connected to the analysis manager
- AliAnalysisTaskESDFilter – The task executed first in the train.
 - Copying to the output AOD standard information and other ESD objects based on the applied cuts
- User tasks (PWG2, 3, 4)
 - Filling existing or adding their own branch to the common AOD tree
 - Using AliAnalysisTaskSE functionality
 - To have access to input/output events
 - To add branches to AOD (AddAODBranch)
 - AOD information produced by a task can be used by a subtask in the same session by connecting the input slot of type TTree to the output container for AOD

Train code

- Some example-based info, train-related code status and code working in all analysis modes were put together on the analysis web page
 - <http://aliceinfo.cern.ch/Offline/Activities/Analysis/AnalysisFramework/index.html>
- The current train is tested in CAF and AliEn
 - <http://aliceinfo.cern.ch/system/galleries/download/OfflineDownload/AnalysisTrain/train.tgz>
 - Local/proof/grid mode macros are provided together with README files

Status table

- Maintained on analysis page

NOTE: this table is not complete and will be update as soon as the developers report the correct status

Group	class	local	CAF	GRID
PWG0	AliDndEtaTask	?	?	?
PWG0	AliDndEtaCorrectionTask	?	?	?
PWG0	AliAnalysisTaskESDfilter	OK	OK	OK
PWG1				
PWG2	AliAnalysisTaskProtons	OK	OK	OK
PWG2	AliAnalysisTaskFemto	?	?	?
PWG2	AliAnalysisTaskESDCheckV0	?	?	?
PWG2	AliAnalysisTaskESDDedx	?	?	?
PWG2	AliAnalysisTaskESDStrange	?	?	?
PWG2	AliAnalysisTaskESDStrangeMC	?	?	?
PWG3	AliAnalysisTaskMounAODfromGeneral	?	?	?
PWG3	AliAnalysisTaskSingleMu	?	?	?
PWG3	AliAnalysisTaskVertexingHF	OK	OK	OK
PWG4	AliAnalysisTaskGamma	OK	OK	OK
PWG4	AliAnalysisTaskJets	OK	OK	OK



Adding/testing new tasks

- Several examples existing already in AliRoot (see tasks in the train)
 - Each analysis module should be able to generate a par file
 - Making a PROOF-INF.MODULE + Makefile for running in proof mode
 - For proof mode the task and all dependent classes must be correctly initialized (follow rules described on analysis page)
- We debugged/included several tasks in the train – a list with most common problems will be prepared
 - We will continue helping in making new tasks compliant with the framework and testing them in the train – contact us if needed



Summary

- New documentation and examples related to the analysis train available
 - List for including new tasks will be prepared
- Working train example
 - More and more ALICE analysis becoming compliant with this run mode
- Train tested regularly in all run modes with code from HEAD
 - Status table updated on any change