

Analysis Framework - status

Andrei Gheata, Mihaela Gheata,
Andreas Morsch

ALICE Offline Week – 09 March 2011

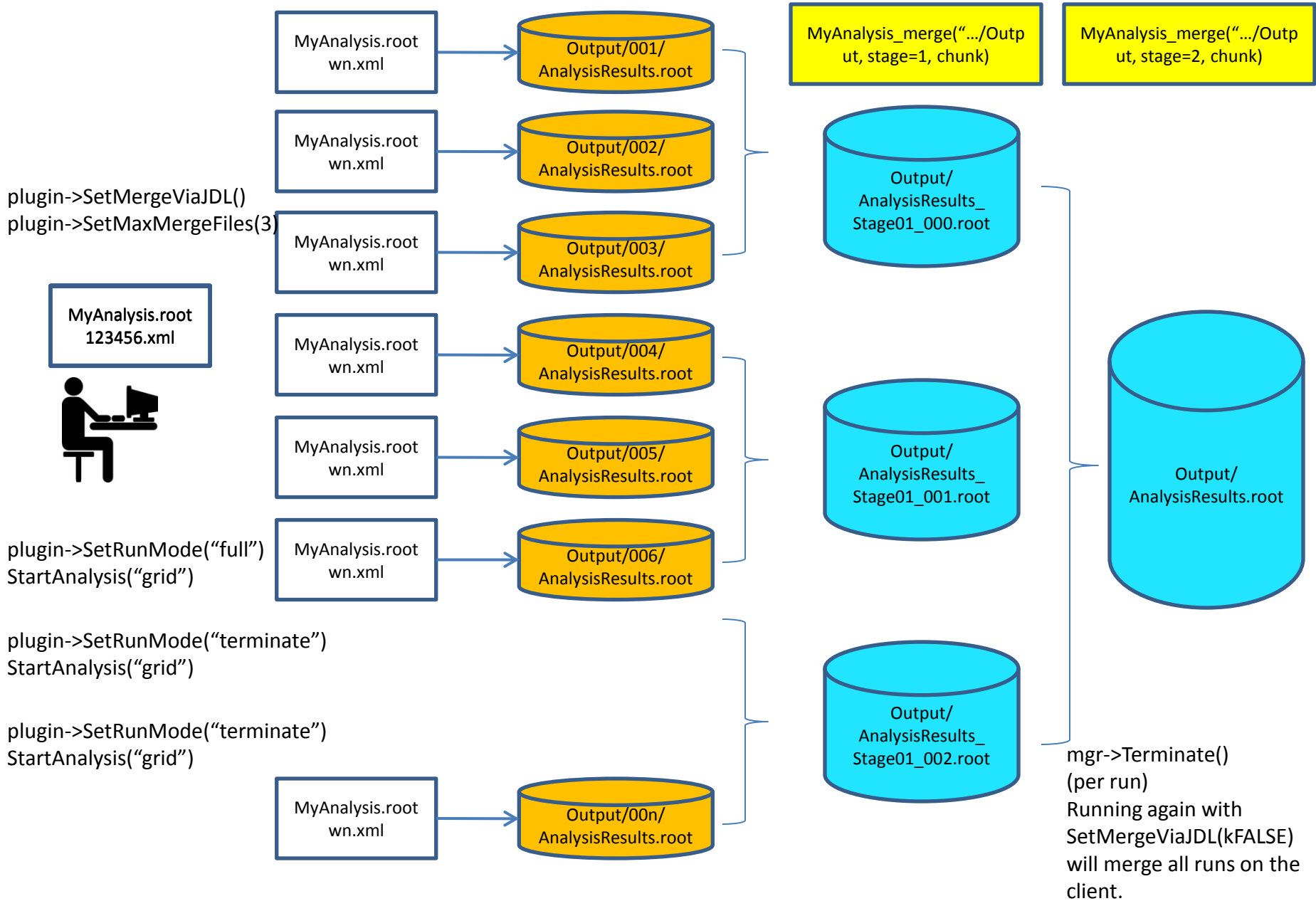
Analysis and analyzers

- The framework enters a more mature phase
 - Focusing more on stability and error detection
 - Providing more and more general purpose utilities (event topology, merging, OADB)
 - Increasing control on EUB (**E**ratic **U**ser **B**ehavior) both in the core framework and back-ends
- User community developing a solid common knowledge base
 - More and more experienced users contributing directly to the framework development (AOD, I/O optimizations, mixing, ...)
 - Many thanks to all !
 - A large and proactive user community exchanging experience on the very popular analysis mailing list
- While the more and more demanding collaboration needs put more and more pressure on the production systems, things are still far from being fully under control
 - Central coordination of massive analysis efforts must prevail EUB...
 - Resources cannot be stretched, only better used

AliEn handler – many small fixes

- Fixes for staged merging
 - Lots of questions on the mailing list while migrating to the new merging
 - Still not perfect but scaling much better with number of files
 - Typical setup:

```
plugin->SetMergeViaJDL(kTRUE);  
plugin->SetMaxMergeFiles(nfiles); // customize to your output  
plugin->SetMaxMergeStages(nstages); // correlate with number of subjobs
```
- Extra checks for loading all needed libraries, overwriting files in output directories
- Generating on demand a file with statistics information
 - mgr->AddStatisticsTask(AliVEvent::kMB)
 - Publishing in Monalisa event processing statistics



New warning messages

- « FIX YOUR CODE – this will produce a Fatal error in future »
 - This looks probably familiar lately, make sure you follow the advice
- Call `PostData()` for all output slots connected to containers of type `kOutputContainer`
 - Will prevent un-balanced output files when having chunks without selected events
- Call `fOutputList->SetOwner()` when creating the containers
 - Will prevent unnecessary leaks during merging
- Other detectable errors will be treated in the same manner.

Central services

- Several improvements in the physics and centrality selection
- Statistics histograms produced by physics selection made available on demand on AOD analysis
 - `aodInputHandler->SetCheckStatistics(kTRUE);`
 - Available via: `inputHandler->GetStatistics()` in `FinishTaskOutput()` for normalization purposes
- Cleaned-up filtering (Laurent) + possibility to select useful AOD/MC branches only
- OADB setup – see presentation by Andreas
- Multi event handler used for optimizing mixing (M.Vala)

Stability issues

- Skipping « bad » events
 - I/O errors are now caught and the events skipped, is this enough ?
 - Correlated with: corrupted files, xrootd issues, temporary SE problems
 - Affecting efficiency of long jobs
- Some fixes or patches already proposed and implemented in root (preventing the nasty effects of unzip errors)
 - Investigating dynamic replica switching

Scalability issues and central trains

- Constrains related to the proliferation of analysis tasks/jobs of all quality trying to process all data as many times as possible
 - User quotas, queue policy, watchdogs
 - So far users have higher priority than production
 - Probably not the best approach anymore
- All PWG need to provide results at due dates
 - PWG central trains were set up so that UEB gets streamed in a well monitored, controlled and scheduled effort
 - This should be the main-stream effort for PWG groups to join
 - Tools: Savannah for communicating and fixing problems within PWG, new mailing list for train operators & central team (to be set up)