AliRoot

Implementation of native CMake build system

A. Grigoras, P. Hristov

CMake

- open-source system that manages the build process independent of the operating system and compiler
- designed to be used in conjunction with the native build environment (Makefiles, Visual Studio, XCode)
- in-place and out-of-place builds, support for multiple builds from a single source tree
- support for complex directory hierarchies and dependencies
  - Focuses on organizing a project in a hierarchy of folders
- generates project files for major IDEs: Visual Studio, Xcode, Eclipse, Kdevelop
- works with parallel make and it is fast
- Very active community providing extensions and documentation

AliRoot – current CMake

- Put in production in July 2008
- Rough translation of the previous Makefiles to Cmake
- Installation done inside the source tree
- Using plenty of environment variables to control the build behavior
- Mixed build environment using source tree, build folder and install folder
- Following a one level folder hierarchy
  - Folder per detector/feature
  - More than one library in the same CMake file sharing the same environment
- Build controlled by centralized macros that leave little flexibility to the way the libraries are built.
- Missing Find macros or not fully implemented ones
- Dependencies not defined correctly
- Very difficult to debug

AliRoot – native CMake

- Trying to follow best practices
- No environment variables, build is controlled only by CMake variables
  - No more ALICE_ROOT, ROOTSYS, DATE_ROOT etc
  - Always use -DCMAKE_INSTALL_PREFIX, -DROOTSYS, -DFASTJET, -DDATE_CONFIG etc
- Build always outside the source tree
  - Always use –DCMAKE_INSTALL_PREFIX
- Source tree restructured to map the library tree
  - TPC/TPCbase will contain a CMakeLists describing libTPCbase
  - TPC will contain a CMakeLists that will include all the subfolders mapped to the TPC libraries
  - Ability to custom control the build per library
    - Setting compilation flags, link flags etc

AliRoot – native CMake

- New Find macros, plus completely rewritten the existing ones
- New macro to generate the rootmaps for the dynamic library loading
- Completely new DA generation
  - DA sources moved to a DA folder with its own CMakeLists file
  - Build twice faster by reusing the existing objects
  - Rpm generation

AliRoot – native CMake

- Put in production at the same time with the AliRoot/AliPhysics split
- Full build/install documentation inside the source tree and webpage
- Development branch: splitdev
  - Give it a try and help us test it!
  - Questions, suggestions: alina.grigoras@cern.ch