Superbuilds for ROOT

Project midterm status

PAVLO SVIRIN, 20-08-2024

ROOT: modular builds

- Goal:
  - to allow to select the components to be built
  - to allow to identify the components which are already built and installed to target directory and offer to skip their compilation
  - to easily add new components to existing installation
  - in case of admin-only rights to write into ROOT’s installation directory: to install new components together with their modulemap files to different directory and then on ROOT’s start combine all of the necessary modulemaps into one

Modular builds: list of works

- distributed modulemap
- definition of ROOT's components as external projects
- compilation of external projects
- installation and packaging

Distributed modulemap files

- Modulemap in ROOT is a file which defines available components in the installation directory, their headers and shared libraries
- Currently include/module.modulemap a file of several hundreds lines
- We managed to split it into multiple files:
  - each file defines one component
  - main modulemap file just includes all of these files
- Benefits:
  - easy to add new components
  - easy to identify which components are already installed
- PR to ROOT #16211

```
hellcat@debian1:/mnt/sdb1/opt/root-modules/bin$ cat ../include/module.modulemap
extern module C "module.modulemap.d/module.modulemap.unix"
extern module C "module.modulemap.d/module.modulemap.Rint"
extern module C "module.modulemap.d/module.modulemap.Thread"
extern module C "module.modulemap.d/module.modulemap.Core"
extern module C "module.modulemap.d/module.modulemap.RIO"
extern module C "module.modulemap.d/module.modulemap.SQLIO"
extern module C "module.modulemap.d/module.modulemap.Net"
extern module C "module.modulemap.d/module.modulemap.RootAuth"
extern module C "module.modulemap.d/module.modulemap.NetxNG"
hellcat@debian1:/mnt/sdb1/opt/root-modules/bin$
```

ROOT: external projects

- Components “interpreter+core+io+math” are considered as an essential part of ROOT
- Naming: for example, “ROOT::net” for the ROOT internal components
  - this allows to decrease the number of modifications to the ROOT macros for CMake
- Inside of a such project there are build targets with unchanged names
  - this is will prevent changes to the DEPENDENCIES section of the targets

CMake/make example calls

```
cmake
    -DENABLE_ROOT_PROJECTS=“net;graf2d” \
    -DROOT_BASE=“/usr;/opt/root2” \
    -DCMAKE_INSTALL_PREFIX=/opt/root2 \
    -DROOT_SRC_DIR=$HOME/devel/root \
    .....
make all install
make package
```

Compilation: components detection

- Lookup is performed under each ROOT_BASE directory
  - if “lib${DEPENDENCY_COMPONENT_NAME}” is present under $ROOT_BASE/lib directory, then a pseudo-target is created for the $ROOT_BASE/lib/lib$DEPENDENCY_COMPONENT_NAME and $ROOT_BASE/include is added to include paths
  - otherwise, the search will be performed under global ROOT sources diirectory ($ROOT_SRC_DIR), necessary external projects and their targets will be enabled

Installation and packaging

- option to install external project files into a directory outside of ROOT_BASE directory works: successful
- Packaging: RPM creation successful for ROOT's external projects
- Internal steps for installation via RPM:
  - copy all of the files from binary dir of external project
  - add a line which points to project's module into ROOT.modulemap

Closest tasks

- introduce selection of components (-DENABLE_ROOT_PROJECTS)
- dependendency tracking for targets and projects which have to be turned on, options:
  - create some kind of manifest (see DIANA-presentation: https://docs.google.com/presentation/d/1s2QbycFOv-en8jiZzWvzMg0Py8XDXJwK9vgB0xIq_Ns/edit#slide=id.g47e0133c06_1_6 , slide 8)
  - build dependency map on configuration step and analyze dependencies
- introduce script which looks for installed components into main configuration process into the whole bundle
- understand and fix the issues with compiler's C++17 support in external projects
- try to minimize mandatory lines in an external project's declaration