CI for CORSIKA 8
Status and plans

Lukas Nellen
ICN-UNAM
luas@nucleares.unam.mx

General Call 2022-10-13

CI container update

- Created and uploaded containers with conan 1.52.0
  - Ubuntu 18.04, 22.04
  - Corresponding clang: 8, 14

- No problems with the containers based on Ubuntu 18.04

- Changes for Ubuntu 22.04
  - Use Legendre functions from boost
    C++ library has a bug before gcc 11
  - Fix in CONEX cmake for select option passing
    Was passing gfortran specific options to C++ compiler

- Ready to merge

conan and cmake

- In MR with container updates
- Separated conan install from cmake
  - Can see conan install and build actions
  - Use special `corsika8` profile for conan
    - Make compiler specific (clang vs gcc, version)
  - Use also for pre-loading cache in containers
- Need to deal with cmake warnings when using more recent cmake

```
(conan) lukas@snufkin:~/CORSIKA8/corsika-build$ ../corsika/conan-install.sh
Found gcc 10
Found clang 11.0
gcc>=5, using the major as version
.
.
.
conanfile.txt: Applying build-requirement: termcap/1.3.1
conanfile.txt: Applying build-requirement: m4/1.4.19
conanfile.txt: Generator txt created conanbuildinfo.txt
conanfile.txt: Generator cmake created conanbuildinfo.cmake
conanfile.txt: Aggregating env generators
conanfile.txt: Generated conaninfo.txt
conanfile.txt: Generated graphinfo
(conan) lukas@snufkin:~/CORSIKA8/corsika-build$ cmake ../corsika \
-DCMAKE_INSTALL_PREFIX=../corsika-install
CMake Warning (dev) at /usr/share/cmake-3.18/Modules/GNUInstallDirs.cmake:225 (messag
  Unable to determine default CMAKE_INSTALL_LIBDIR directory because no
  target architecture is known.  Please enable at least one language before
  including GNUInstallDirs.
Call Stack (most recent call first):
  CMakeLists.txt:16 (include)
This warning is for project developers.  Use -Wno-dev to suppress it.

-- The CXX compiler identification is GNU 10.2.1
-- Detecting CXX compiler ABI info
-- Detecting CXX compiler ABI info - done
.
.
.
-- Configuring done
-- Generating done
-- Build files have been written to: /home/lukas/CORSIKA8/corsika-build
(conan) lukas@snufkin:~/CORSIKA8/corsika-build$ make -j 20
Scanning dependencies of target GenParticlesHeaders
Scanning dependencies of target CorsikaData
Scanning dependencies of target EPOS_static
Scanning dependencies of target cnpy
Scanning dependencies of target Sibyll_static
Scanning dependencies of target GenMediaProperties
Scanning dependencies of target pythia8
Scanning dependencies of target Sibyll
Scanning dependencies of target UrQMD_static
[  0%] Read PYTHIA8 particle data and produce C++ source code GeneratedParticle[...].
[  1%] Read NIST properties8 data file and produce C++ source code GeneratedMediaProp
[  1%] Building CXX object modules/data/readLib/CMakeFiles/CorsikaData.dir/source/Int
[  2%] Creating directories for 'pythia8'
[  3%] Building CXX object externals/cnpy/CMakeFiles/cnpy.dir/cnpy.cpp.o
[  3%] Building Fortran object modules/sibyll/CMakeFiles/Sibyll_static.dir/sibyll2.3d
```

clang-format-14 reports

- How do we deal with changes in clang-format?
  - Do we freeze clang-8?
  - Change tool?
  - Update reference?
- Could provide apptainer container

```
Apptainer> export CLANG_FORMAT=clang-format-14
Apptainer> ./do-clang-format.py --all
Ubuntu clang-format version 14.0.0-1ubuntu1

Note: the clang-format version has an impact on the result. Make sure you are
consistent with current CI. Consider '--docker' option.
./cmake-build-debug/CMakeFiles/3.23.2/CompilerIdCXX/CMakeCXXCompilerId.cpp
./cmake-build-debug/modules/pythia8/Pythia8ConfigurationDirectory_install.hpp
./cmake-build-debug/corsika/modules/pythia8/Pythia8ConfigurationDirectory.hpp
./cmake-build-release/CMakeFiles/3.23.2/CompilerIdCXX/CMakeCXXCompilerId.cpp
./cmake-build-release/corsika/modules/pythia8/Pythia8ConfigurationDirectory.hpp
./cmake-build-release/modules/pythia8/Pythia8ConfigurationDirectory_install.hpp
./cmake-build-relwithdebinfo/corsika/modules/pythia8/Pythia8ConfigurationDirectory.hpp
./cmake-build-relwithdebinfo/CMakeFiles/3.23.2/CompilerIdCXX/CMakeCXXCompilerId.cpp
./cmake-build-relwithdebinfo/modules/pythia8/Pythia8ConfigurationDirectory_install.hpp
./tests/framework/testProcessSequence.cpp
./corsika/media/UniformMagneticField.hpp
./corsika/detail/media/WeightProvider.inl
./corsika/detail/modules/proposal/ProposalProcessBase.inl
./corsika/detail/framework/geometry/QuantityVector.inl
./corsika/detail/framework/geometry/FourVector.inl
```

User container: apptainer (singularity)

- Tested using converted docker container from CI
  - Will create special user containers
- Tests suggest that this is the way to distribute binary
  - Should work on CentOS 7 (more tests pending)

```
lukas@snufkin:~/CORSIKA8/corsika-build-c14$ apptainer run \
  --bind $HOME/.conan:/conan/.conan ../buildc14.sif
INFO:    underlay of /etc/localtime required more than 50 (83) bind mounts
Apptainer> cat /etc/os-release
PRETTY_NAME="Ubuntu 22.04.1 LTS"
NAME="Ubuntu"
VERSION_ID="22.04"
VERSION="22.04.1 LTS (Jammy Jellyfish)"
VERSION_CODENAME=jammy
ID=ubuntu
ID_LIKE=debian
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
UBUNTU_CODENAME=jammy
Apptainer> echo $CXX
clang++-14
Apptainer> $CXX --version
Ubuntu clang version 14.0.0-1ubuntu1
Target: x86_64-pc-linux-gnu
Thread model: posix
InstalledDir: /usr/bin
Apptainer> ctest
Test project /home/lukas/CORSIKA8/corsika-build-c14
    Start 1: copyright_notices
1/7 Test #1: copyright_notices ................   Passed    0.32 sec
    Start 2: testData
2/7 Test #2: testData .........................   Passed    0.04 sec
    Start 3: testFramework
3/7 Test #3: testFramework ....................   Passed    2.00 sec
    Start 4: testMedia
4/7 Test #4: testMedia ........................   Passed    0.04 sec
    Start 5: testStack
5/7 Test #5: testStack ........................   Passed    0.01 sec
    Start 6: testModules
6/7 Test #6: testModules ......................   Passed  114.28 sec
    Start 7: testOutput
7/7 Test #7: testOutput .......................   Passed    0.10 sec

100% tests passed, 0 tests failed out of 7

Total Test time (real) = 116.85 sec
Apptainer>
```

Pending: CI server and runner

- Add caching
  - compiled code
  - proposal (and other?) tables
  - will clone from auger for single server use
  - eventually: use S3 @ ICN-UNAM
- Runners with different number of cores
- In November …

Colateral: review Geomagnetic Field

- Why this: Legendre Polynomials
- Relevant: potential intense use in tracking

- Better design: factory for field at given date
  - interpolate/extrapolate coefficients once

- Optimize evaluation of Legendre Polynomials
  - Need all polynomials up to degree of expansion
  - Explicit use of recursion relations
    - Recursion relations are used internally in library functions
  - Avoids repeated evaluation

- Alternative: could be pre-calculate in region of interest and interpolate?
  - ROI: cylinder or prism around core

Conclusion

- Conan update done in containers
  - Reduce CI runtime: don’t compiler externals every time
- Separation of conan and cmake done
- CI server/runner updates pending
- Success using apptainer [1] containers
- Pending: re-implement container building tools in buildah
  - Simpler scrips
  - More flexible
  - CI (docker, podman) and user (apptainer, podman)

[1]: https://apptainer.org/