g4tools @ Wollongong

diff –u « since last workshop »

G.Barrand, CNRS/IN2P3/LAL

Main ideas

- g4tools is an automatic extraction of some code found in the softinex/inlib and namespaced “g4tools” for an embedding in Geant4.
- Pure header code. Highly portable (including iOS and Android). Easily embeddable (no “config.h” or specific build tool in the way).
- Strongly OO. No implicit management.
- Thread safe (no writable statics).
- See http://softinex.lal.in2p3.fr

What's new : HDF5

- Code to write histos and ntuples by using the HDF5 library.
- (Then a relationship to an external package).
- HDF5 : a library dedicated to do binary IO for scientific data. See https://www.hdfgroup.org/HDF5 for the technicalities.
- See https://www.hdfgroup.org for the sociology and also the "A few of our users" for existing applications.

HDF5 : directories

- API and functionalities are here to do what we want. No blocking point for the moment.
- In particular we use the named “H5Group” to organize data in file in “directories” in a way similar to what is done with ROOT/IO.
- (In HDF5, whilst you know what is a H5File, a H5Group and a H5Dataset, you know the essential).

HDF5 : histos

- The data for one histo are put in one named group and there is one named “H5Dataset” per “field” (bin_Sw, bin_Sw2, etc..).
- This permits to keep a “data schema” comprehensible without having to read the writing (g4tools/hdf5/h2file) code. In particular the “h5ls” program, coming with the library, permits to scan easily a file.
- (For generic data as an histo, it looks more interesting to store in this way than having a “H5Compound” storing an histo in one blind “BLOB” in the file).

HDF5 : ntuples

- Similar API than for the ROOT/IO driver.
- An ntuple handles columns which are paged.
- An ntuple is attached to a named H5Group which contains one group per column. Each column group maintains a H5Dataset, and a page of data (a buffer in memory) is appended/written to the dataset when full.
- The logic here is similar to the basket logic of ROOT/IO.

HDF5 : ntuples (2)

- We can handle columns of basic types and string and vector of basic types and string.
- There is also compression (done with zlib).
- Performances are similar to CERN-ROOT (if using same page/basket size). (Sure, the logic is the same and we are dominated, in both cases, by the speed of the file system).
- We provide an example (cern_root_hdf5_ntuple.cpp) to read an ntuple from a g4tools/hdf5 file, project on an TH1D and plot in a TCanvas.

HDF5 : //

- It works in multi-threads, with one file per thread, with an HDF5 lib built for multi-threads.
- We have to look if we can do the same logic that we have now for the ROOT/IO format to have only one file for multiple threads and/or MPI workers.
- Passing histos should be ok. For an ntuple we have to look if we can pass “pages” from one slave to a master handling a single file.
- But HDF5 seems to have some “parallel” logic, then perhaps all is already here. To be looked.

HDF5 : H5Fclose

- For the moment I am very happy with HDF5.
- (And beside my Mac and Linux, I can read a file on my Windows-10, Android and iOS devices).
- The "data schemas" in file is keep simple so that it should be easy to read histos and ntuples from other contexts without the g4tools/hdf5 readers (from python ? from some web tool ?).
- It gives ideas : storing G4 geometries with HDF5 ?