New Developments In Analysis

I. Hrivnacova, IPN Orsay (CNRS/IN2P3),

24th Geant4 Collaboration Meeting,
26 September 2019, Jefferson Lab

G4analysis @ Jefferson Lab
diff -u «since last workshop»

- Score Ntuple Writer
- Merging of Ntuples with MPI
- Analysis Factory Function

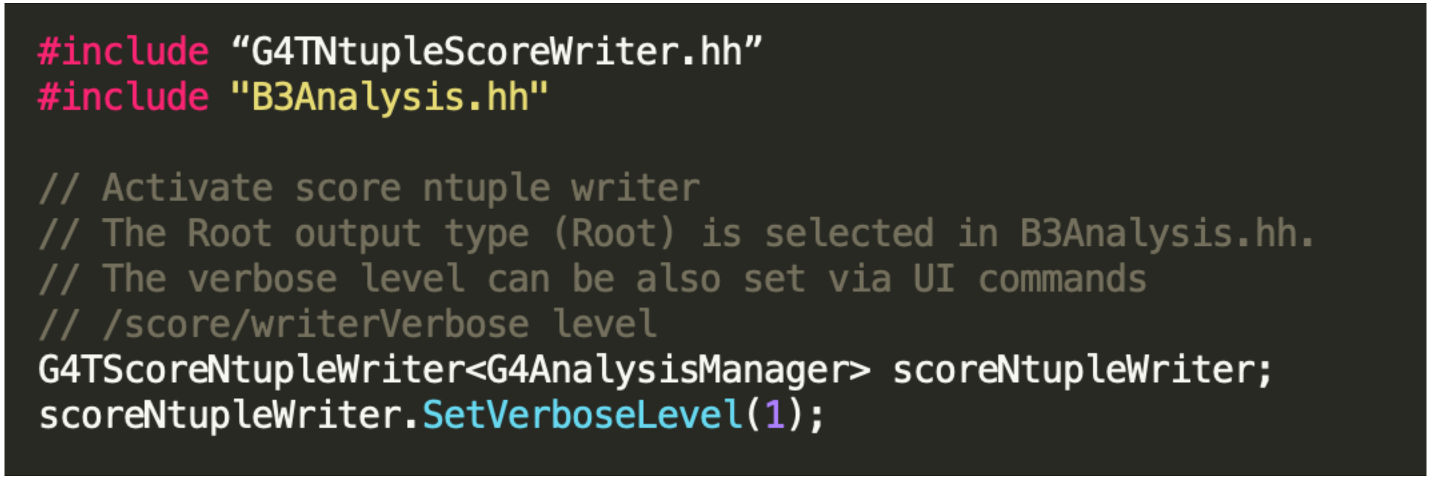
Score Ntuple Writer

Score Ntuple Writer – 10.5

- Since Geant4 10.5
- It is possible to save the scorers hits using Geant4 analysis tools. This functionality is assured by the G4VScoreNtupleWriter interface.
- In 10.5 this feature is demonstrated in the **B3aScoreWriter** and **B4dScoreWriter** extended examples in the analysis category.
  - Both examples are completely based on the basic B3a and B4d examples respectively. Storing hits is activated in the `main()` function with instantiating G4ScoreNtupleWriter.
    - Storing all hits collections of `G4THitsMap<G4double>` type
  - The G4ScoreNtupleWriter and G4ScoreNtupleWriterMessenger classes are provided in the 'scoreWriter' directory in each example.
    - The G4ScoreNtupleWriter depended on both analysis and digits_hits categories => it could not be included in kernel

Score Ntuple Writer – 10.6

- New solution in 10.6: G4ScoreNtupleWriter => G4TScoreNtupleWriter
  - The binding of analysis via a template
  - New writer class could be included in `digits_hits/utilities`
- The new B3aScoreWriter and B4dScoreWriter examples could have been removed and the feature is now demonstrated in basic examples B3a and B4d



```
#include “G4TNtupleScoreWriter.hh”
#include "B3Analysis.hh"

// Activate score ntuple writer
// The Root output type (Root) is selected in B3Analysis.hh.
// The verbose level can be also set via UI commands
// /score/writerVerbose level
G4TScoreNtupleWriter<G4AnalysisManager> scoreNtupleWriter;
scoreNtupleWriter.SetVerboseLevel(1);
```

exampleB3a.cxx

Score Ntuple Writer Messenger

- G4TScoreNtupleWriterMessenger implements the following interactive commands:

```
/score/writerFileName filename
/score/writerVerbose value
```

Merging of ntuples via MPI

Merging of ntuples via MPI

- Ntuple merging was implemented in g4tools already in 10.4, integration in the G4 analysis manager in 10.5
- In difference from merging other data (G4Run, scorers, histograms), the ntuple data are not sent to the collecting rank(s) at the end of run but *during event processing*.
- That's why **(an) extra rank(s)** have to be reserved for this purpose.
- **g4mpi** was adapted to allow to define (an) extra worker(s) for collecting data from processing workers
- Added classes:
  - G4mpi extension: G4VMPIextraWorker and G4MPIextraWorker
  - Analysis extension: G4RootMpiAnalysisManager, G4RootMpiNtupleManager, G4RootMpiPNtupleDescription, G4RootMpiPNtupleManager
    - Use of MPI via an interface, these classes are planned to be moved in the Geant4 analysis category

Merging of ntuples via MPI (2)

- New example: **exMPI04**
  - As exMPI03, but adds output in G4analysis ntuple and demonstrates ntuple merging.
- The number of extra workers requested is set in the G4MPImanager constructor in `main()`.
  - While G4MPImanager can be created with any number of extra workers (< the total number), the ntuple merging is at present supported only for one.
- The standard calls to G4AnalysisManager (creating ntuple, open, write and close file) trigger creating all necessary analysis tools object for merging ntuple data on flight.

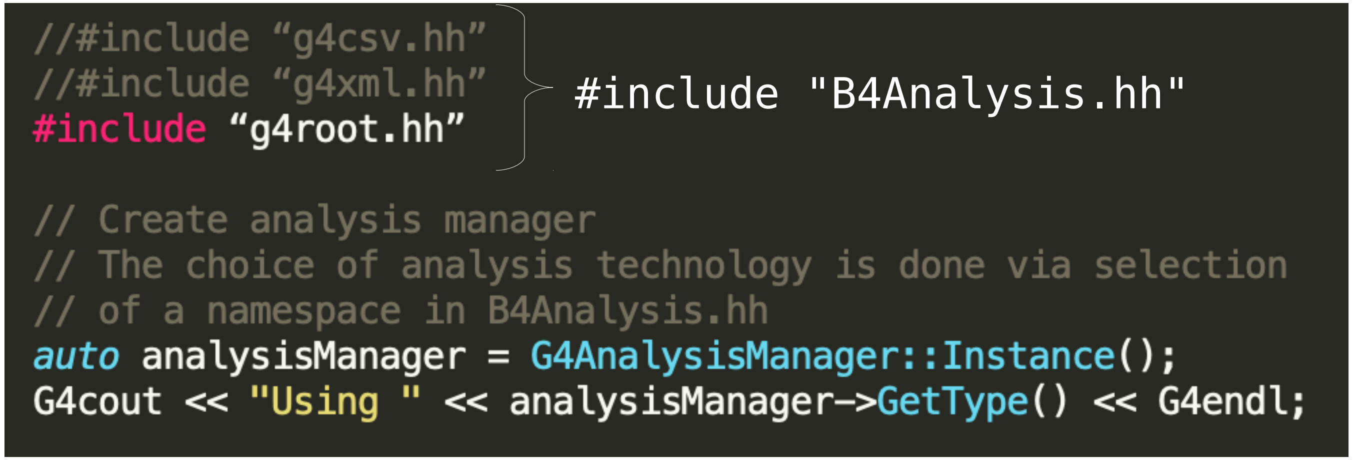
.

Merging of ntuples via MPI (3)

- MPI ntuple merging can be activated **in sequential mode only**; this activation is performed by creating the G4MPIntupleMerger object in the **RunActionMaster** constructor
- If **multithreading mode** is enabled, the ntuples are merged from threads to files per ranks.
- *Combined MT + MPI merging is not (yet) supported.*
- Merging ntuples is actually supported only with the ROOT output format.

Analysis Factory Function

Analysis Factory Function

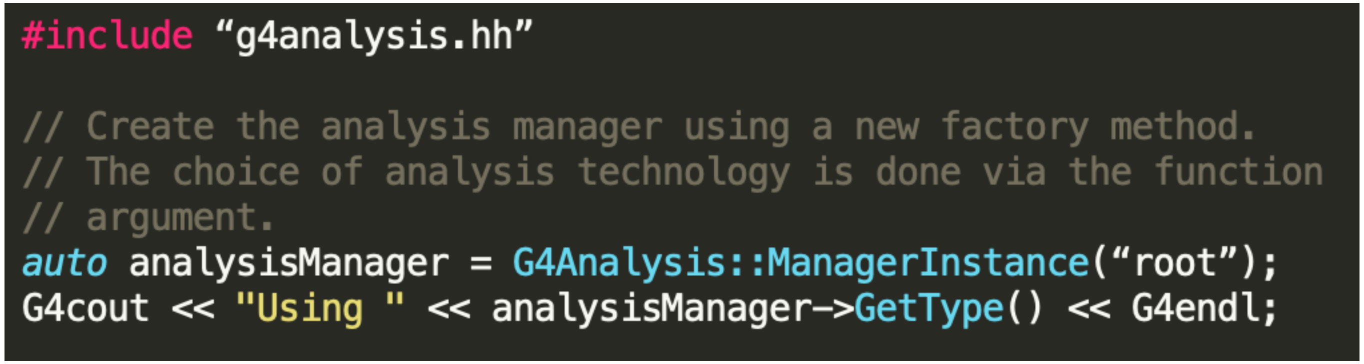


```
//#include “g4csv.hh”
//#include “g4xml.hh”
#include “g4root.hh”
```

```
#include "B4Analysis.hh"
```

```
// Create analysis manager
// The choice of analysis technology is done via selection
// of a namespace in B4Analysis.hh
auto analysisManager = G4AnalysisManager::Instance();
G4cout << "Using " << analysisManager->GetType() << G4endl;
```

OLD (10.5)



```
#include “g4analysis.hh”

// Create the analysis manager using a new factory method.
// The choice of analysis technology is done via the function
// argument.
auto analysisManager = G4Analysis::ManagerInstance(“root”);
G4cout << "Using " << analysisManager->GetType() << G4endl;
```

NEW (10.6)

Analysis Factory Function (2)

- The factory function in G4Analysis namespace:

  G4AnalysisManager*

  G4Analysis::ManagerInstance(const G4String& outputType)

  to create the analysis manager of the type selected via a string argument
- The return G4AnalysisManager type (defined in g4analysis.hh) is G4ToolsAnalysisManager
  - While in case of including the output specific typedefs, eg, g4root.hh, it is the output type specific manager, eg. G4RootAnalysisManager
- It does not make a difference in most common use cases (as in examples), an extra care will be needed when accessing the g4tools output type specific objets, ntuples
  - A static cast to the specific type would be needed in this case, eg.

```
static_cast<G4RootAnalysisManager>(analysisManager)->GetNtuple();
```

Analysis Factory Function (3)

Implementation details:

- New analysis sub-category "factory" added
- Clean-up analysis type definitions:
  - G4Hn*, G4Pn* types moved out of type specific namespace
- Methods for activating/setting ntuple merge mode were added also in G4VAnalysisManager
  - With a default implementation (issuing a warning when merging is not available)

Plan

- Improve ntuple merging in row-wise mode
  - To be included in 10.6
- Additional flexibility in resetting/deleting histograms
- Review support for writing same histogram/profile on file several times
- Handling of more files by analysis manager
  - These items may require a deeper redesign of the manager classes
  - Namely to disentangle files & ntuples handling from histograms/profiles and so to disconnect file and ntuple managers from the analysis manager
  - Brainstorming needed to minimize the impact on the user code
  - Moved for 2020