LHCb software and MoEDAL simulation

Matthew King

IFIC, Valencia

5/21/2014

Attached scripts and packages

- Attached to these slides is a .tar file with the scripts necessary to set up Gauss or Panoramix on a CernVM local machine or on lxplus.

- The file also contains the most recent version of the MonopolePhysics package, which has been written to allow the LHCb simulation software to incorporate Monopoles.
  - It also contains scripts and instructions on how to install and run this package.

- Ideally, an SVN for storage and distribution of this code would be preferred, but does not currently exist.

LHCb software updates

- LHCb software is organised into:
  - Packages – Sets of classes for a particular purpose (tools, algorithms etc.).
  - Groups – Sets of packages that perform similar operations or work in a particular processing step (Generation, Simulation).
  - Projects – Complete Gaudi software package consisting of several groups. (i.e. Gauss, Panoramix)

- Complete projects can be:
  - setup and run on lxplus via afs
  - setup locally or installed locally via the CernVM File System (cvmfs)

- Checking out packages for modification inside a project had proven a little more complicated
  - (And was necessary for making the MoEDAL simulation)

LHCb software updates

- Checking out packages cannot be done using the "getpack" command, as instructed in all LHCb tutorials, without using an LHCb account.

- However, the direct SVN command does still work. Procedure:
  1. **Source _loginscript.sh (sets necessary environment variables)**
  2. **SetupProject [Project] [version] (gets session ready to run project)**
  3. **setenv[Project] [version] (goto a working directory for the project)**
  4. **svn co "svn+ssh://${user}@svn.cern.ch/reps/lhcb/[Project]/tags/[Group]/[Package]/[version]" [Package]**

- The path can be most easily found on the svn tracker, i.e. to check out the controlling package for **Gauss-v46r7** the svn tracker page:

  https://svnweb.cern.ch/trac/lhcb/browser/Gauss/tags/Sim/Gauss/v46r7

- corresponds to a command (for me) of:

  svn co "svn+ssh://mking@svn.cern.ch/reps/lhcb/Gauss/tags/Sim/Gauss/v46r7" Gauss

MoEDAL lxplus accounts?

- lxplus accounts with native access to the LHCb svn remain preferable.
  - Some bugs may later appear.
  - Unclear yet whether grid jobs requiring LHCb software will work from a non-LHCb account.

- However, for now it appears that Gauss simulation jobs and Panoramix can be run on any lxplus account
  - MoEDAL accounts do not currently appear to be essential.

MoEDAL Simulation

- Gauss is the LHCb simulation project.
  - The simulation engine used is Geant4 (a separate project in LHCb)
  - Gauss communicates with the Geant4 session using the classes in the GiGa (Geant4 in Gauss) package.

- GiGa provides a set of base classes for: Physics lists, Field setups etc.
  - New physics is implemented in an inheriting class and added to the Gauss algorithm.

- It currently appears that physics can only be added by altering the python/Gauss/Configuration.py file in the Sim/Gauss package
  - => Need to check out, modify and compile Sim/Gauss package to run MoEDAL simulation

MoEDAL simulation

- Monopole physics can be added to Gauss by adding GiGaPhysContructorMonopole (MonopolePhysics) to the algorithm's Physics List. This class:
  - initialises the monopole particle instance (G4Monopole).
  - uses a modified transportation class (G4MonopoleTransportation) to replace the default (electric charge only) Lorentz equation with a completed version (G4MonopoleEquation).
  - adds a monopole specific ionisation process to the monopole (G4mplIonisation).

- The attached script compiles and runs. It has been confirmed to use the above classes in describing monopole movement.
- However, adding an algorithm to get results from the simulation into an nTuple or histogram file has proved trickier than expected.
  - Meeting with Gloria Corti next week to address this problem and confirm validity of progress so far.

Further work

- The MoEDAL detector elements will probably need to be made into sensitive detectors in the simulation if we want to know the hit positions in the plastic layers.
  - Not sure how to implement this yet. Will have to ask Gloria.
    - (Work so far is in the MonopoleAnalysis package in the tar file)

- We will need to have our own output format from the simulation
  - Output format:
    - Direct to histograms?
    - nTuple containing all information for histogram generation in ROOT?
  - Variables:
    - List of variables required would help.
    - Do we care about the monopole's path or only its end point?