Running Fun4All on Cori (updated)

Reynier Cruz Torres

(borrowing some text from D. Dixit and F. Torales-Acosta)

May 12, 2021

Contents

1 Logging into Cori and getting Fun4All working

First, ssh into cori (`ssh -Y username@cori.nersc.gov`) and go to the directory where Fun4All is to be installed. Clone the Singularity container from Github and install the package following:

```
git clone https://github.com/sPHENIX-Collaboration/Singularity.git
cd Singularity/
mkdir install
./updatebuild.sh
shifter --image=docker:ddixit/fun4all:eicresearch /bin/bash
```

The last command line above lets you enter the docker image using shifter and sets the shell to bash.

Create a file called `set_g4.sh` containing the following lines:

```
export core_dir=/global/path/to/Singularity/cvmfs/sphenix.sdcc.bnl.gov/gcc-8.3/opt/sphenix/core
export G4LEVELGAMMADATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/PhotonEvaporation5.5
export G4_MAIN=$core_dir/geant4.10.06.p02
export G4LEDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4EMLOW7.9.1
export G4NEUTRONHPDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4NDL4.6
export G4ENSDFSTATEDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4ENSDFSTATE2.2
export G4RADIOACTIVEDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/RadioactiveDecay5.4
export G4ABLADATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4ABLA3.1
export G4PIIDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4PII1.3
export G4PARTICLEXSDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4PARTICLEXS2.1
export G4SAIDXSDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4SAIDDATA2.0
export G4REALSURFACEDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/RealSurface2.1.1
export G4INCLDATA=$core_dir/geant4.10.06.p02/share/Geant4-10.6.2/data/G4INCL1.0
```

Create another file called *e.g.* `setup_fun4all.sh` and include the following lines (modify the lines depending on where your files are located):

```
singularity_dir="/global/path/to/Singularity"
source ${singularity_dir}/cvmfs/sphenix.sdcc.bnl.gov/gcc-8.3/opt/sphenix/core/bin/sphenix_setup.sh
export MYINSTALL="${singularity_dir}/install"
source ${singularity_dir}/cvmfs/sphenix.sdcc.bnl.gov/gcc-8.3/opt/sphenix/core/bin/setup_local.sh $INST
source /global/path/to/set_g4.sh
```

In principle you should only need to modify the paths in the first and last lines.

`source setup_fun4all.sh`

Check that the setup was successful by running in the terminal: `lsb_release -a`. You should see the following output:

```
LSB Version: :core-4.1-amd64:core-4.1-ia32:core-4.1-noarch
Distributor ID: Scientific
Description: Scientific Linux release 7.3 (Nitrogen)
Release: 7.3
Codename: Nitrogen
Input: root #see if root works for you.
```

At this stage you can also run `root` in the terminal to check that Root is already working. Once you have root working, try the steps from the "Try an event display" tutorial here:

https://github.com/sPHENIX-Collaboration/macros

Remember, we will be using `sphenix_setup.sh` not `.csh`. If the you can run "Try an event display successfully", congratulations, you are able to run Fun4All on Cori. Finally, create a directory called `install` inside the Singularity directory.

1.1 Using Fun4All from now on (follow these steps every time you log into Cori)

From now on, you will always have to do the following steps to access Fun4All again on Cori:

```
shifter --image=docker:ddixit/fun4all:eicresearch /bin/bash
source setup_fun4all.sh
```

Step 1 enters the Singularity image, and Step 2 sets up all environment variables.

2 Additional repositories

2.1 Building a package

The steps are explained in detail in the following page in the “Building a package” section: https://wiki.bnl.gov/sPHENIX/index.php/Example_of_using_DST_nodes

The basic steps are the following:

1. Create a build directory. For instance, if the package that will be installed is called “package”, do `mkdir build_package`.

2. Go into this directory: `cd build_package`.

3. `path/to/package/source/autogen.sh --prefix=$MYINSTALL`.

4. `make install -j 4`

5. Lastly, redo the last three steps from section 1.1:

```
source local_path_to/opt/sphenix/core/bin/sphenix_setup.sh -n
export MYINSTALL=/global/path/to/Singularity/install
source Singularity/cvmfs/sphenix.sdcc.bnl.gov/x8664_sl7/opt/sphenix/core/bin/setup_local.sh $MYINSTALL
```

2.2 Jet Analysis

From inside the Singularity directory, clone the repository here:

```
git clone https://github.com/ftoralesacosta/e_Jet_sPHENIX
```

Follow the steps from section 2.1 to build this package.

2.3 All-Silicon tracker

The All-Silicon Tracker we have been studying at LBNL can be found here. From inside the Singularity directory, clone the repository:

```
git clone https://github.com/eic/g4lblvtx
```

Follow the steps from section 2.1 to build this package.

3 Running All-Si tracker code

3.1 Code details

The main macro is: `g4lblvtx/macros/Fun4All_G4_FastMom.C`. This section describes some details from this code.

3.1.1 Detector geometry

The detector geometry was created in EICroot, saved into a TGeo file, and then translated into a gdml file. The gdml file where the detector geometry is defined is: `FAIRGeom.gdml`, and is loaded in the main macro in the lines that read:

```
AllSiliconTrackerSubsystem *allsili = new AllSiliconTrackerSubsystem();
allsili->set_string_param("GDMPath","FAIRGeom.gdml");
allsili->AddAssemblyVolume("VST");      // Barrel
allsili->AddAssemblyVolume("FST");      // Forward disks
allsili->AddAssemblyVolume("BST");      // Backward disks
allsili->AddAssemblyVolume("BEAMPIPE"); // Beampipe
```

3.1.2 Event generator

```
const int particle_gen = 1;
```

Below are the possible settings for this variable:

- `particle_gen = 1`, the generator used is a particle generator, which generates a given particle per event over an extended range in angles, momenta, etcetera.
- `particle_gen = 2`, the generator used is a particle gun, which shoots the same particle with the same properties (angles, momenta, ...) over and over.
- `particle_gen = 3`, the generator used is a "simple event generator" (using the Fun4All class `PHG4SimpleEventGenerator`) which can generate, for example, 100-pion events. This cab be used for primary-vertex studies.
- `particle_gen = 4`, the generator used is pythia8.

3.1.3 Magnetic field

```
const int magnetic_field = 4;   // 1 = uniform 1.5T, 2 = uniform 3.0T, 3 = sPHENIX 1.4T map, 4 = Beast 3.0T map
```

Below are the possible settings for this variable:

- `magnetic_field = 1`, uniform 1.5 T solenoidal field.

- `magnetic_field = 2`, uniform 3.0 solenoidal field.
- `magnetic_field = 3`, realistic sPHENIX (BaBar) 1.4 T magnetic-field map.
- `magnetic_field = 4`, realistic Beast 3.0 T magnetic-field map.

3.2 Running All-Si code on login node

```
root -l Fun4All_G4_FastMom.C
```

or by first loading root, and then doing:

```
.x Fun4All_G4_FastMom.C
```

3.3 Running batch jobs on Cori

Some tips to get started with Slurm can be found here.

Here is an example on how to submit a batch job on Cori. To submit the jobs from `run_shared_pim.sh` do the following:

1. exit the container: `exit`
2. go to the directory where `run_shared_pim.sh` is.
3. run: `sbatch run_shared_pim.sh`

To check the status of your jobs, go in your internet browser to https://my.nersc.gov/ and login. Then go to the tab “Cori Queues”. Alternatively, you can run in the terminal (from outside the Singularity container):

```
squeue -u username
```

To cancel all submitted jobs, you can run in the terminal (from outside the Singularity container):

```
scancel -u username
```