FCC Software: Step by step

Valentin Volkl, for the Key4hep Software Group
Feb 10th, 2022
FCC Physics Workshop 2022

This work benefited from support by the CERN Strategic R&D Programme on Technologies for Future Experiments (https://cds.cern.ch/record/2649646/, CERN-OPEN-2018-006).

0.1 This talk:

- is based on documentation available online:
  - https://cern.ch/key4hep
  - https://hep-fcc.github.io/fcc-tutorials/
- which can be freely edited and improved
- documents a rapidly evolving project

0.2 Overview

- Setup and getting started
  - CVMFS
  - Virtual Machine + CVMFS
- Working with Data Files
- Event Generation and Fast Simulation
  - Generators
  - Delphes
  - RDataFrame-based Analyses
- Full Simulation Workflows
  - FCCee LAr
  - CLD

0.3 Setup with CVMFS

- Requirements: Centos7, Bash shell

```
~$  /cvmfs/fcc.cern.ch/sw/latest/setup.sh
# is an alias for:
```

```
~$ source /cvmfs/sw.hsf.org/key4hep/setup.sh
```

```
Setting up the latest Key4HEP software stack from CVMFS ...
 ...  Key4HEP release: key4hep-stack/2021-11-26
 ... Use the following command to reproduce the current environment:
 ...
         source /cvmfs/sw.hsf.org/spackages4/key4hep-stack/2021-11-26/
         /x86_64-centos7-gcc8.3.0-opt/mynqr2z/setup.sh
 ...
 ... done.
```

0.4 Setup with a Virtual Machine

Make sure VirtualBox is installed (detailed installing instructions from the product web page). * Download the `cernvm-launch` binary for your platform either from the dedicated download page or from the following links: * Linux * Mac * Win

- Get the fcc-tutorial.context (use wget or curl)

Once you have all this you can create the VM with this command:

```
$ cernvm-launch create --name fcc-tutorial --cpus 4 --memory 8000 \
  --disk 20000 fcc-tutorial.context
```

You an choose how many CPU cores to use, the memory and the disk space. Good rules of thumb are to use half the cores of your machine, at least 2 GB memory per core, and enough disk for your job.

The above command should open a window with VirtualBox and produce on the screen an output like this

```
Using user data file: fcc-tutorial.context
Parameters used for the machine creation:
        name: fcc-tutorial
        cpus: 4
        memory: 8000
        disk: 20000
        cernvmVersion: 2020.07-1
        sharedFolder: /Users/ganis
```

From now on you can either work in the VirtualBox window or ssh to the machine with

```
cernvm-launch ssh [username@]fcc-tutorial
```

0.5 Finding data on EOS

```
/eos/experiment/fcc/
    ee
        accelerator
        analyses
        datasets
   ...
    eh
        datasets
    hh
        analyses
        CDR_script
      ...
    prod # (for DIRAC)
```

Access preferably via xrootd:

```
root root://eospublic.cern.ch//eos/experiment/fcc/hh/tests/test.root
```

0.6 The `events` tree

Uses EDM4hep format. Some branch name conventions

- `GenParticles`: Particles from either a generator like Pythia8 or a particle gun
- `GenParticlesFiltered`: The same collection, but filtered according to some criteria (mostly used to select stable particles that are seen in the detector)
- `SimTrackerHits`: Geant information on energy deposits in the tracker
- `SimTrackerHitsPositions`: The same, but including the MC truth information on the exact coordinates of the deposits
- `SimCaloHits`: Geant information on energy deposits in the calorimeter
- `SimCaloHitsPositions`: The same, but including the MC truth information on the exact coordinates of the deposits
- `SimParticles`: MC truth information about the particles in Geant simulations
- `RecTrackStates`: Helix parameters as reconstructed by the software
- `RecParticles` Full particle information after Reconstruction

0.7 The `metadata` tree

```
%%bash
k4-print-joboptions https://fccsw.web.cern.ch/fccsw/testsamples/tutorial/
↪fccee_idea_pgun.root
```

```
GeoSvc.OutputLevel = 3
GeoSvc.detectors = ['/cvmfs/fcc.cern.ch/sw/releases/fccsw/0.12/
↪x86_64-centos7-gcc8-opt/share/FCCSW/Detector/DetFCCeeIDEA/compact/
↪FCCee_DectMaster.xml']

SimG4Alg.outputs = ['SimG4SaveParticleHistory/saveHistory',␣
↪'SimG4SaveTrackerHits/saveTrackerHits_Barrel', 'SimG4SaveTrackerHits/
↪saveTrackerHits_Endcap', 'SimG4SaveTrackerHits/saveTrackerHits_DCH']

SimG4Alg.GeantinoGun.energyMax = "2400.0"
SimG4Alg.GeantinoGun.energyMin = "2400.0"
SimG4Alg.GeantinoGun.etaMax = "3.5"
...
```

1 Event Generation and Fast Simulation

1.1 Generators

- Whizard

```
whizard <process_config>.sin
```

- Madgraph

```
mg5_aMC
```

- Herwig

```
Herwig -h
```

- KKMCee

```
KKMCee -f Mu -e 91.2 -n 1000 -o LHE_OUT_1.LHE
```

- Pythia

Pythia is integrated in the framework: hep-fcc/k4gen

```
k4run $FCCSW/Examples/options/pythia.py
```

1.2 Delphes

key4hep/k4SimDelphes

Direct EDM4hep output with `Delphes{*}_EDM4HEP` commands: (Files available from here)

```
DelphesPythia8_EDM4HEP card_IDEA.tcl edm4hep_IDEA.tcl
            p8_noBES_ee_ZH_ecm240.cmd  p8_ee_ZH_ecm240_edm4hep.root
```

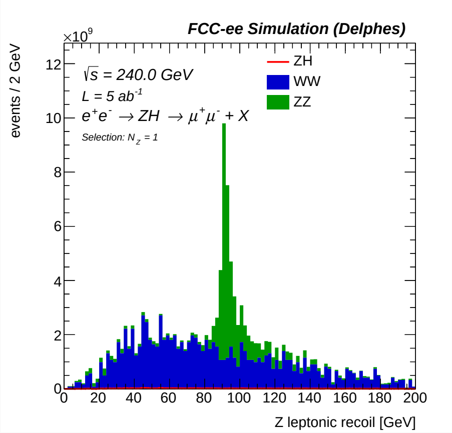
```
...


  33.782     52.054    -67.455     91.657      0.000
     9           1  d                      23      6      0      0
    10          -1  dbar                   23      6      0      0
                                    Charge sum:  0.000

 --------  End PYTHIA Event Listing  -------------------------
** 99 events processed
 Pythia::next(): 100 events have been generated
** 199 events processed
 Pythia::next(): 200 events have been generated
 ...
```

1.3 RDataFrame-based Analyses

hep-fcc/fccanalyses

With files from the documentation webpage:

```
python analysis.py p8_ee_ZH_ecm240.root p8_ee_ZH_ecm240_edm4hep.root
python finalSel.py
python $FCCANALYSES/config/doPlots.py plots.py
```



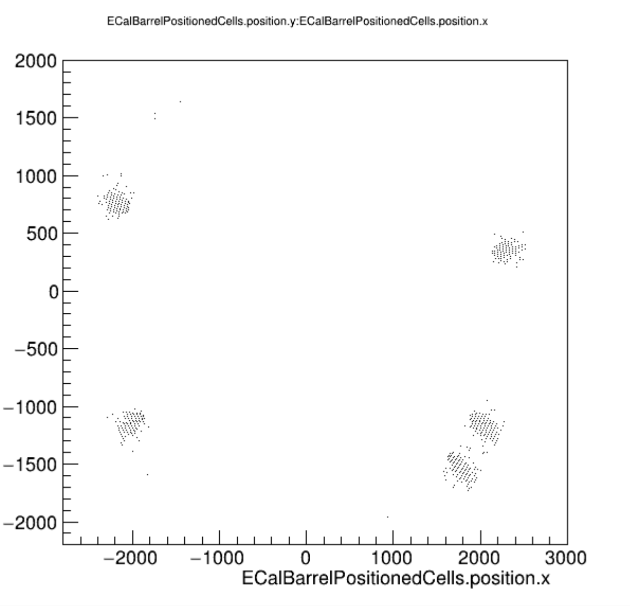
2 Full Detector Simulations

Repositories used in this section:

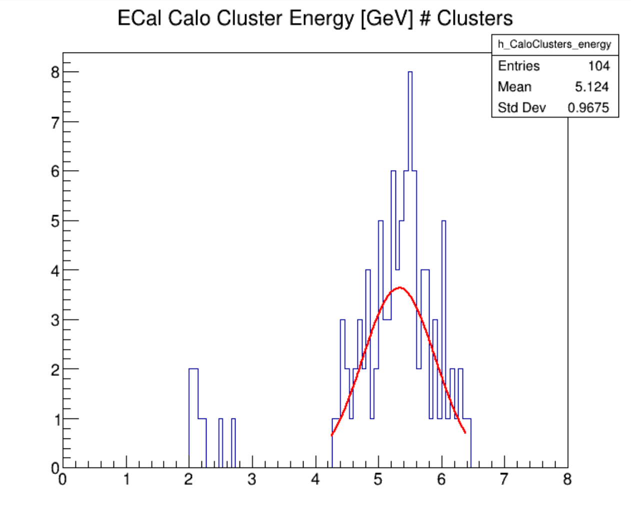
- hep-fcc/k4Gen
- key4hep/k4FWCore
- hep-fcc/k4SimGeant4
- hep-fcc/k4RecCalorimeter
- hep-fcc/FCCDetectors
- ilcsoft/lcgeo

2.1 FCCee LAr Calorimetry: Simple performance workflow

```
# Momenta in MeV
 k4run $K4RECCALORIMETER  /RecFCCeeCalorimeter/tests/options/runCaloSim.py
 --filename fccee_idea_LAr_pgun.root
 --GenAlg.MomentumRangeParticleGun.MomentumMax 5000
 --GenAlg.MomentumRangeParticleGun.MomentumMin 5000  -n 1000
```

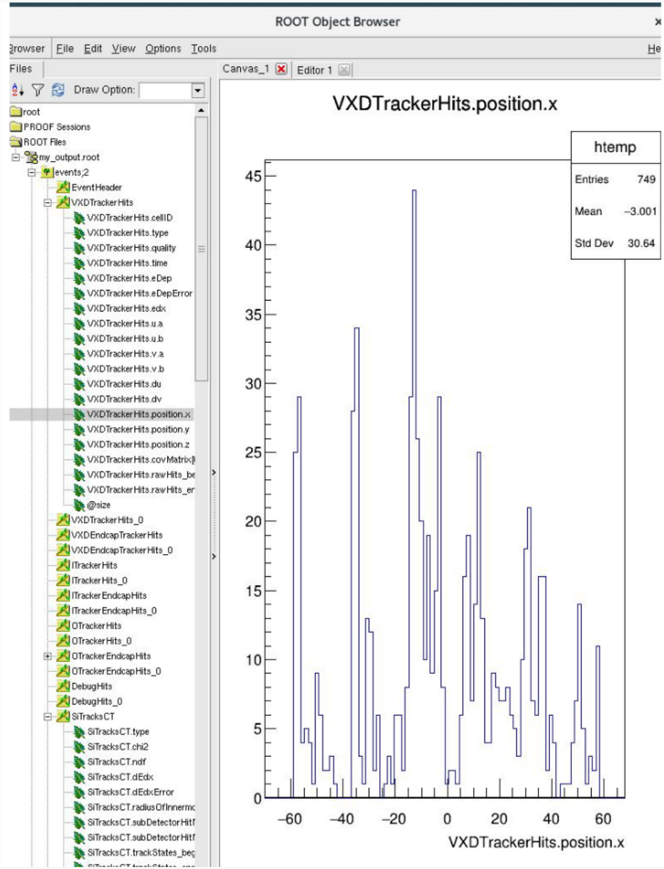


```
k4run $K4RECCALORIMETER/RecFCCeeCalorimeter/tests/options/
      /runFullCaloSystem_ReconstructionSW_noiseFromFile.py
  --input.EventDataSvc fccee_idea_LAr_pgun.root
  --noiseFileName http://fccsw.web.cern.ch/fccsw/testsamples/
      /elecNoise_ecalBarrelFCCee_50Ohm_traces1_4shieldWidth.root
  --filename output_allCalo_reco_noise.root
  -n 500
```



2.2 Full Simulation of CLD using k4MarlinWrapper

```
git clone https://github.com/vvolkl/clicperformance --branch fccee-cld
cd fcceeConfig
ddsim \
  --compactFile ${LCGEO}/FCCee/compact/FCCee_o2_v01/FCCee_o2_v01.xml \
  --outputFile ttbar.slcio \
  --steeringFile fcc_steer.py \
  --inputFiles ../Tests/yyxyev_000.stdhep \
  --numberOfEvents 3

k4run fccReconstruction.py
```



ROOT Object Browser
Browser File Edit View Options Tools
Files
Draw Option:
Canvas_1
Editor 1
root
PROOF Sessions
ROOT Files
my_output.root
events;2
EventHeader
VXDTrackerHits
VXDTrackerHits.cellID
VXDTrackerHits.type
VXDTrackerHits.quality
VXDTrackerHits.time
VXDTrackerHits.eDep
VXDTrackerHits.eDepError
VXDTrackerHits.edx
VXDTrackerHits.u.a
VXDTrackerHits.u.b
VXDTrackerHits.v.a
VXDTrackerHits.v.b
VXDTrackerHits.du
VXDTrackerHits.dv
VXDTrackerHits.position.x
VXDTrackerHits.position.y
VXDTrackerHits.position.z
@size
VXDTrackerHits_0
VXDEndcapTrackerHits
VXDEndcapTrackerHits_0
ITrackerHits
ITrackerHits_0
ITrackerEndcapHits
ITrackerEndcapHits_0
OTrackerHits
OTrackerHits_0
OTrackerEndcapHits
OTrackerEndcapHits_0
DebugHits
DebugHits_0
SiTracksCT
SiTracksCT.type
SiTracksCT.chi2
SiTracksCT.ndf
SiTracksCT.dEdx
SiTracksCT.dEdxError
VXDTrackerHits.position.x
htemp
Entries 749
Mean −3.001
Std Dev 30.64
45
40
35
30
25
20
15
10
5
0
−60
−40
−20
0
20
40
60
VXDTrackerHits.position.x

3 Conclusions

- more examples at hep-fcc/fccsw
- check back for updates:
  - https://cern.ch/key4hep
  - https://hep-fcc.github.io/fcc-tutorials/
- bring up issues and propose changes on the linked github repositories
- regular key4hep and fccsw meetings!