FOCAL simulation

Ionut Arsene (Oslo)

Code/libraries structure

- Main software repository: **https://gitlab.cern.ch/mvl/FOCAL**
- For the integration in **ali-sw/** we created a dedicated directory (FOCAL/aliroot/)
- Structure of the code and libraries under aliroot/:
  - FOCALbase/
    - Contains most classes
  - FOCALsim/
    - AliFOCALv0, AliFOCALv1, AliFOCALv2, AliHALLv4, AliPIPEFOCAL
  - FOCALgen/
    - AliGen*
- Corresponding libraries: libFOCAL(**base**,**sim**,**gen**).so

Installation using aliBuild

- Follow the general instructions from the aliBuild documentation
- FOCAL depends on AliRoot and ROOT
  - aliBuild installs all dependencies automatically
- How to install FOCAL with aliBuild:
  1) Go to your alice software directory
  2) Update alidist:
     1) `shell> cd alidist`
     2) `shell> git pull`
  3) Initialize the FOCAL package
     1) `aliBuild init FOCAL`
  4) Build FOCAL
     1) with ROOT-5: `aliBuild build FOCAL`
     2) with ROOT-6: `aliBuild build FOCAL –defaults next-root6`

Installation using aliBuild

- Load the FOCAL environment
  - `alienv enter VO_ALICE@FOCAL::latest`
  - FOCAL system variables are created: $FOCAL, $FOCAL_RELEASE, $FOCAL_VERSION
    - $FOCAL points to the installation directory and can be used to conveniently point to include/ and lib/ directories
  - No need to load the individual *_cxx.so in macros, just load the libraries

DPG tools

- Use DPG existing tools for running simulations both on a local machine or on the grid
  - Install AliDPG:
    - `cd <your ALICE software dir>`
    - `aliBuild init AliDPG`
    - `aliBuild build AliDPG`
  - Initialize the environment using **`alienv`** (load AliDPG, FOCAL and AliROOT):
    - `alienv enter VO_ALICE@FOCAL::latest VO_ALICE@AliDPG::latest`
    - Optionally one can load also AliRoot-OCDB if you want to use a local OCDB:
    - `alienv enter VO_ALICE@FOCAL::latest VO_ALICE@AliDPG::latest VO_ALICE@AliRoot-OCDB::latest`

Run simulations locally

- `cd <working directory>`
- It is convenient to have two steps:
  - ➔ Create an OCDB snapshot (*this step is common to both local and grid productions*):

    Get the OCDB entries from grid (default in AliDPG):

    - `$ALIDPG_ROOT/bin/aliroot_dpgsim.sh --mode ocdb --run 294925`

    or, pick the OCDB from a custom location (need to provide an OCDBCustom.C macro in your working directory):

    - `$ALIDPG_ROOT/bin/aliroot_dpgsim.sh --mode ocdb --run 294925 –ocdbCustom`

    Note: AliDPG needs a run number to be provided in order to generate the OCDB snapshot. Here “294925” is the last good run from Run-2 pp at 13 TeV

Run simulations locally

- `$ALIDPG_ROOT/bin/aliroot_dpgsim.sh --mode sim --run 294925 --detector FOCAL --simulation NoDigitization --focalGeometryFile <geometryFile.txt> --generator <generatorName> --uid 1 --nevents <#events>`
  - `<generatorName>` can be any generator implemented in AliDPG (see https://twiki.cern.ch/twiki/bin/viewauth/ALICE/AliDPGMonteCarloTutorial).
  - The generators already configured in *$FOCAL/simulation/Config.C* are commited also to AliDPG (*$ALIDPG_ROOT/MC/CustomGenerators/Upgrade/FOCAL_Generators.C*) and one can use them like this:
  - `--generator Upgrade:FOCAL_Generators:<gentype>`

    `<gentype>`: `gun, box, pythia, pythia_MBtrig, pythia_dirgamma_trig, ntuple, hijing, hijingAP, PA_cocktail_dirgam, PA_cocktail_MBtrig, PA_cocktail_EPOS_MBtrig, PA_cocktail_EPOS_dirgam`
  - Additional options are available via the DPG script (e.g. use GEANT4 or FLUKA instead of GEANT3)
- Known issues:
  - STARLIGHT generator is not compatible with ROOT6
  - EPOS in FOCAL_Generators still needs to be adapted to running with AliDPG

Run simulations on grid

- Make sure you have an active alien token

1) Create a working directory on alien: `alien_mkdir </alice/cern.ch/user/.../<workDirectory>>`
2) Copy to the grid working directory the OCDB snapshots (*OCDBsim.root* and *OCDBrec.root*), the FOCAL geometry file and *focalJDL.jdl* file
   - The FOCAL geometry file can be any of the geometry files
   - The *focalJDL.jdl* is a special file containing the simulation command to be run and a few other parameters needed to steer the grid simulations. I prepared it and I will provid it either in the repository or via twiki.
3) Go to alien: `alien.py`
4) Launch the jobs: `submit <jdl Filename> <working directory> <geometry file> <generator> <output directory relative to workdir> <nevents> <njobs>`

   Example: `submit focalJDL.jdl /alice/cern.ch/user/i/iarsene/work/focalDPG/ geometry.txt Upgrade:FOCAL_Generators:PA_cocktail_dirgam run10 5 15`

   Note: The number of arguments and/or meaning of the provided JDL arguments may still change, as I am trying to make the submission tools more configurable for the user.

   focalJDL.jdl file available on the FOCAL twiki

Run simulations on grid

- All the steps on previous slide will be automatized using a script (bash or python), to be commited shortly to $FOCAL
- When running as a simple user, job submission to grid is limited by user quotas
  - If we need larger scale simulations we can either request to increase user quotas or ask for simulations via the usual channels (DPG and PB).
- To do:
  - Currently with AliDPG only Hits are generated, ongoing work to implement also the digitization part
  - Higher level operations, like clusterization or analyses of produced Kinematics and Hits can also be run on the grid in principle; *to be discussed*

FOCAL packages in cvmfs

- Check available packages in cvmfs: http://alimonitor.cern.ch/packages/
  - VO_ALICE@FOCAL::db2df60f9f6735b5c258d8b23a921e16d136d082-1 (based on ROOT6)
    - First FOCAL version deployed on grid
  - VO_ALICE@FOCAL::v1.0-1 (based on ROOT5), VO_ALICE@FOCAL::v1.0-2 (based on ROOT6)
    - Contains latest updates from Norbert
    - Should appear on cvmfs shortly (https://alice.its.cern.ch/jira/browse/ALIROOT-8767)

Analysis software

- MCInfo (FOCAL/MCInfo)
  - Creates a tree with up to 4 tracks per event and their matched clusters
  - Useful for simulations with one generated primary particle
- Clustering (FOCAL/clustering)
  - Runs the clusterizer
- Analysis (FOCAL/analysis, FOCAL/analysis_pythia)
  - Match reconstructed clusters to simulated particles
- FOCAL twiki: https://twiki.cern.ch/twiki/bin/view/ALICE/FOCALSoftware
- Analysis of simulations can be done currently on a local machine/farm (see twiki)
  - Working on a solution to run simulation analysis on grid

Summary and outlook

- FOCAL software integrated with the ALICE aliBuild framework
  - Currently there are two copies of the FOCAL classes
    - The one used by aliBuild (FOCAL/aliroot) and the old directory structure
    - We should converge to a single version
- Tools to run simulations with a variety of options (MC generator, G3/G4/Fluka,etc) possible using the DPG central tools (aliroot_dpgsim.sh)
  - Few issues are still there, but work is ongoing
- Example macros for single particle and full event simulation analysis exist
- Twiki page: https://twiki.cern.ch/twiki/bin/view/ALICE/FOCALSoftware
- Regular meetings (weekly or bi-weekly ?)
  - Main timeslot: Mondays at 13:00
  - Reserve slot (on request): Thursday at 16:00