

Realistic MC generation

Saverio D'Auria

- Suggestions for DIY-MC
- Suggestions for submitting requests
- MC production experience: open issues and hiccups
- Next steps and wish-list from the user point of view.

Realistic MC: the twofold-way

- runMC (mcProduction package)
 - Pro's:
 - one executable: only change tcl file.
 - Runs on farms: big production
 - All the Random seeds are taken care of
 - Run sections are generated naturally
 - Con's
 - One executable (big core dump !)
 - Machinery is complicated if you only need a few tens thousand events (signal MC)
 - Intermediate results are available for check
 - Difficult maintainance
- Alex's MC tar-file
 - Pro's
 - Uses standard packages
 - Handled easily
 - 5-step process
 - Con's
 - Need to keep intermediate results (x2 more disk)
 - 5-step process
 - (Scripts only for Bgen)

Realistic MC: the twofold-way

- runMC (mcProduction package)
 - Pro's:
 - one executable: only change tcl file.
 - Runs on farms: big production, DFC, DH
 - All the Random seeds are taken care of
 - Run sections are generated naturally
 - Con's
 - One executable (big core dump !)
 - Machinery is complicated if you only need a few tens thousand events (signal MC)
 - Intermediate results are available for check
 - Additional package to maintain
- Alex's MC tar-file
 - Pro's
 - Uses standard packages
 - Handled easily
 - 5-step process
 - Con's
 - Need to keep intermediate results (x2 more disk)
 - 5-step process
 - Scripts only for Bgen

Talk about this 

Use Alex's tar file if you like

The same modules are called in both

DIY MC

Pick up your favourite package:

Few (<100 k) events, no data handling, Bgen → use Alex's prescriptions

Large production, farm, DH, Pythia, Herwig,... → use runMC

Do and don't:

- Don't generate > 1000 evts with same run number and run section:

default length is 1000 evts, don't overwrite it

```
Talk GenInputManager
```

```
run_number set $run
```

```
RunSectionOffset set $runSection
```

```
exit
```

- Don't generate HEPG file with > 1000 events. Make many files, make good use of parallel computing !!
- Don't use tcl files for MC to analyze these sample ! They behave just like data DB-wise.

For Data handling purposes:

- Make output files of 1 GB: avoid concatenation.
- Ask for a dataset if you want to make your sample available to CDF and stored on tape.
 - **mbot## is the Montecarlo dataset for Bottom group**
- Use standard file names that are generated automatically by your FileOutput module if you specify dataset , book and database.
 - Provide a dummy filename ! The name will be overwritten, the directory stays the same.
- You can store files using SAM and DFC, but at present
 - **SAM files go to Fermilab enstore, available worldwide, not from FileInput**
 - **DFC files go to CDF enstore, but are available locally from FileInput**

MC production experience: open issues and hiccups

- runMC does not contain the full trigger simulation. It would be very welcome because one could do all in one job.
- Even with realistic MC need to run TRIGSim++ for electron and muon triggers.
- **PHYSICS charge deposition model**: reported to crash in 4.9.1, but I produced 4E6 events with one crash only.
- Sets of tcl files being prepared for subdividing runMC so that one can easily use the flexibility of multi-process job.
- Trend: to include a multi-process job (CdfSim, Production, TRGSim++, “filter”) in runMC.
- Samples generated:
 - Generic bb (200k) with 2TT, *dataset mbot01, fileset SA0000.0*
 - J/Psi (4.5×10^6), *dataset mbot02, fileset GI1188.0*
 - generic cc (15k) with 2TT
 - Signals (300 k)

Next steps:

- Multi-run automated procedure, weighted number of events according to the run-integrated luminosity.
- Re-ordering of tcl files in view of re-ordering of the code
- Using Glasgow farm: disk space is the main issue with multi-step.