Condor with GEANT4

- 55 machines support GEANT4 10.2.1 with multithreading support
- What you need to do:

1. Prepare submit file
2. Prepare GEANT4 application for condor (i.e. how output files are saved and returned to you – naming conventions so they don't overwrite one another)

GEANT4 example

- My proton CT simulation takes as input arguments the number of protons and integer defining angle of rotation (number between 1 and 720).
  - Initially this was in a macro file, but I would have to make 720 different files with 720 different parameters for condor usage otherwise
- To run my geant4 app from terminal: ./model <angleID> <numOfProtons>
- Saves output file as out<angleID>.csv

Steps

1) Connect (ssh) to submitter node and build GEANT4 code as normal (don't run on submitter node, except small runs to test!)
   - Submitter node address: submitter.itservices.manchester.ac.uk
2) Create submit files and run condor_submit (condor_submit submitFile.txt)
3) Wait for output!

Submit file

- Need to tell condor what computers we want to use:
  - Memory?
  - Processor/cores?
  - How to rank choice of computer, i.e. memory
- Where to save log of job (i.e. time submitted, time executed, time terminated, reason for termination), output from terminal, and error messages
- Where to find the executable file on submitter node
- What arguments to pass to executable
- Where files to transfer with executable i.e. macro files

SubmitGeant.txt

```
Universe = vanilla
requirements = (Opsys == "LINUX" && Arch == "X86_64" && HAS_GEANT4_10_2==true)
rank = Memory
request_memory = 500

Log = geant.log
Output=geant.out
Error = geant.error
notification = error

WhenToTransferOutput = ON_EXIT
ShouldTransferFiles = Yes

executable = run_geant.sh
arguments = 0 400000

transfer_input_files = model,run_geant.sh
Queue 1
```

SubmitGeant.txt – multi jobs

```
Universe = vanilla
requirements = (Opsys == "LINUX" && Arch == "X86_64" && HAS_GEANT4_10_2==true)
rank = Memory
request_memory = 500

Log = geant.$(Process).log
Output=geant.$(Process).out
Error = geant.$(Process).error
notification = error

WhenToTransferOutput = ON_EXIT
ShouldTransferFiles = Yes

executable = run_geant.sh
arguments = $(Process) 400000

transfer_input_files = model,run_geant.sh

Queue 720
```

run_geant.sh

```
#!/bin/bash

. geant4.sh

./model $1 $2

# this is needed to source the environment variables for geant4 to run on the computer the job has been submitted to, otherwise you'll get an error
```

Submitting C++ jobs

Application name: BPFMod

- Arguments: <angleID>
- Input files: proj<angleID>.csv
- Output files: out<angleID>.csv

Note: I was getting errors due to missing libraries, and had to compile with the -static option to run.

```
Universe = vanilla
requirements = (Opsys == "LINUX" && Arch == "X86_64")
rank = Memory
request_memory = 400

Log = BPFMod.$(Process).log
Output=BPFMod.$(Process).out
Error = BPFMod.$(Process).error
notification = error

WhenToTransferOutput = ON_EXIT
ShouldTransferFiles = Yes

executable = BPFMod
arguments = $(Process)

transfer_input_files = BPFMod,proj$(Process).csv

Queue 720
```

Submitting MATLAB jobs

MATLAB code: WEPL.m

- Arguments: <angleID>
- Input files: out<angleID>_nt_Model.csv
- Output files: proj<angleID>_nt_Model.csv
- Compilation:

  mcc -R -singleCompThread -m WEPL.m
- Creates run_WEPL.sh and WEPL files

```
Universe = vanilla
requirements = (Opsys == "LINUX" && Arch == "X86_64" && HAS_MATLAB_2013=?=True)
rank = Memory
request_memory = 500

Log = matlab.$(Process).log
Output=matlab$(Process).out
Error = matlab$(Process).error
notification = error

WhenToTransferOutput = ON_EXIT
ShouldTransferFiles = Yes

executable = run_WEPL.sh
arguments = /opt/MATLAB/MATLAB_Compiler_Runtime/v81/ $(Process)

transfer_input_files = WEPL,run_WEPL.sh,out$(Process)_nt_Model.csv

Queue 720
```

Notes

- I'm able to run ~ 200 jobs simultaneously during the day, not sure about evenings
- On the weekend, reached > 2000 jobs with C++ jobs
- Useful commands:
  - condor_submit submitFilename.txt
  - condor_q username (see jobs you have submitted, how long they've been running etc)
  - condor_rm username (remove all jobs)
  - condor_prio -p <priority> <processID> (change priority of job)

Useful links

- http://matchmaker.itservices.manchester.ac.uk/condorstatus/
  - shows status of condor pool
- http://research.cs.wisc.edu/htcondor/manual/current/11_Command_Reference.html
-