FE-Geant4

Ecole doctorale – Geant4

Session 7 : Exercise

Code for download: session7_start.tar.gz

Exercise 7a:

- Migrate the exampleED application to MT. After each step rebuild the program. You can also try to run and observe the behaviour (breaks) after each step and understand how the next step fixes the observed break.
  Steps:
  1. Update main: add `G4MTRunManager`.
  2. Update `ActionInitialization` class: add `BuildForMaster()` function.
  3. Update Hit classes: declare `G4Allocator` thread local.

  After these steps the migration is complete. Rebuild and rerun the program.

- Increase the number of events (~300) and observe the Root output. Inspect the ntuple files generated per threads with TChain, for example:

```
TChain chain("Chamber1");
chain.Add("ED_t0.root");
chain.Add("ED_t1.root");
// Inspect tree branched (ntuple colums) from all files
chain.Draw("Zpos");
// Merge the files in one
chain.Merge("ED_Chamber1.root");
```

Exercise 7b:

- Change the default number of threads: first in the code and then via the environment variable. Observe which setting has the preference.
- Limit the output to one thread only (via a command in a macro).
- Get an experience with a data race. Add a global variable defined in a file scope in `EDChamberSD.cc` just after the headers in commented lines:

```
G4double* myGlobalValue = new G4double(1.);
```

  and in `EDChamberSD::ProcessHits()`:

```
// simulate data race
if ( (*myGlobalValue) > 0.) {
delete myGlobalValue;
myGlobalValue = new G4double(-1);
}
else {
delete myGlobalValue;
myGlobalValue = new G4double(1);
}
```

  Does the program break now?

Congratulation, the race condition was added successfully.
Use `G4AutoLock` to fix this thread-unsafe code.

Solution: session7_solution.tar.gz