GEANT4 Cross Section Optimizations

Robert Fowler and Paul Ruth
RENCI / UNC Chapel Hill

Pedro Diniz
ISI / USC

USC Viterbi
School of Engineering
Information Sciences Institute

renci
RESEARCH \ ENGAGEMENT \ INNOVATION

Updates to Cross Section Code

- Revised `GetCrossSection(part,mat)` Method:

```
G4double G4CrossSectionDataStore::GetCrossSection(part,mat){
  ...
  if((part == part_ref)&&(mat == mat_ref)&&(this->proc == proc_ref)){
    rslt = getCrossSectionTable(hash(part_ref, mat_ref, proc_ref, energy);
    clear xssecelem();
    return rslt;
  } else {
     // --- regular code filling in xsecelem;
  }
  ...
}

G4double G4CrossSectionDataStore::SampleZandA(part,mat){
  ...
  if(isEmptyXSECELEM()){
    // --- compute the various cross section per element of the material
    //    this is the lazy evaluation of the xsecelem
  }
  ...
}
```

Updates to Cross Section Code

- Short-Term Code Update
  - Minor modifications of the G4CrossSectionDataStore class objects and methods (.hh and .cc codes in /source/process/hadronic/crosssections)
  - Inclusion of references to Physics Process (both G4String and Enumerated data Fields
  - Initialization Method uses externally-provided table with selected particles, material and physics' process
  - Uses Segmented table-based interpolation to compute the cross section functions (lazily for materials' elements if needed)
  - Pick subset of (particle, material, process) to demonstrate performance improvement.
- Medium-Term Code Update
  - Initialization Method indicating which Particle, Material and Process can use predefined table (controlled by command file)
  - Load predefined file or generate on the fly.

Cross Section Usage

- Particle/Material Pairs
  - – 50% of calls in ~10 particle/material pairs
  - – 90% of calls in ~40 particle/material pairs
  - – Observed ~4k pairs
- Particle/Material/Process Triples
  - – 50% of cycles in ~10 triples
  - – 90% of cycles in ~85 triples
  - – Observed ~18k triples
- Implementing fast path for tens of pairs (or triples) can speedup the nearly all of the calls.

Fast Path Usage

Particle: neutron
Material: materials_StainlessSteel
Process: G4Neutron InelasticXS

Slow path only:

	Cycles	Calls	Cycles/Call
Slow Path	6,133,110,476	6,278,517	977

Fast path with lazy computation of slow path:

	Cycles	Calls	Cycles/Call
Slow Path	223,362,860	94,876	2,354
Fast Path	1,059,541,332	5,887,001	179
Total	1,282,904,192	5,981,877	214

Possible ~5x speed up of cross section calculation

Additional Observation

- Observation
  - Multiple calls to GetCrossSection with exactly the same particle, material, process, and energy
  - Results in same cross section result
- Optimization
  - Cache recent cross section for particle, material, process triple.
- Measurements
  - 17% of calls would benefit from this cache
  - 29% of GetCrossSection cycles are from these calls.
  - ~18k triples (probably can't cache all of them)
  - ~3k triples would need cache (still too many)