Multithreaded Geant4 (Geant4MT)

- Event-level parallelism to simulate separate events by multiple threads
- Efficiency for future many-core CPUs
- Testing and validation on today's 4-, 8- and 24-core nodes
- Preliminary results available based on testing on `fullCMS bench1.g4`
- Patch parser.c of gcc to output static and global declarations in Geant4 source code and add the "__thread" keyword
- Separate and share read-only data members : Geant4 parameterised geomeries and replicas, Geant4 materials and particles, Geant4 physics tables, etc.
- Custom malloc library to support thread private allocation
- Modified G4Navigator to remove unnecessary updates to G4cout and G4cerr precision (shared variables)

"Multi-core & multi-threading: Tips on how to write "thread-safe" code in Geant4", Xin Dong and Gene Cooperman, *14th Geant4 Users and Collaboration Workshop Search*, `http://indico.cern.ch/sessionDisplay.py?sessionId=68\&slotId=0\&confId=44566#2009-` and `http://indico.cern.ch/conferenceDisplay.py?confId=44566`

Experimental Results on 24-core Intel Xeon 7400 Computer

By segregating read-write data members, large read-only memory chunks are formed. Copy-On-Write does not replicate those read-only chunks. (Geant4MT + COW)

- Separate Processes: No reduction for the memory footprint
- Geant4 + COW: Share geometries (no replica or parameterized geometry)
- Geant4MT + COW: Reduce the memory footprint
- Geant4MT: Reduce the memory footprint

Tested on `fullCMS bench1.g4` with 24 workers and 4000 events per worker (electromagnetics).

Implementation	Total Memory on master	Additional Memory per Worker	Total Memory (master + 24 workers)	Runtime
Separate Processes	250 MB	250 MB	6 GB	4575 s
Original Geant4 + COW	250 MB	70 MB	2G MB	4571 s
Geant4MT + COW	250 MB	20 MB	730 MB	4540 s
Geant4MT 24 threads	250 MB	20 MB	730 MB	4510 s