Sub-Event Parallelism

Makoto Asai (SLAC)

23rd Geant4 Collaboration Meeting

Aug. 28, 2018

Sub-event parallelism

- Geant4 MT was initially designed to process events in parallel.
  - Each event is tasked a thread.
    - Independence of threads makes perfect scaling of throughput with #threads
  - But scheme assumes events are small enough to fit into the memory of one thread
- Sub-event parallelism generalizes this approach:
  - To serve the case of applications requesting large memory per event:
    - e.g. ALICE, HL-LHC, air shower
  - One event is split into “sub-events”
    - e.g. each few primary tracks = a sub-event
    - Split method is obviously user-dependent.
  - Each sub-event is sent to a thread, and merged back to the original full event later
    - Geant4 will provide tools to easily enable this feature
    - ATLAS already has this mechanism within ATHENA, ALICE is developing it.
  - All the current API’s should be preserved.

Work plan

- Constraint – all the current API's must be preserved.
  - Special G4UserPrimaryGeneratorAction class will be introduced, and will be used only in the master thread.
  - It has SplitEvent() and MergeEvent() methods.
- Constraint – Objects instantiated by a worker thread must be deleted by the same worker thread.
  - Only the objects of G4PrimaryVertex and G4PrimaryParticle will be instantiated in the master thread and associated to a G4Event object, and G4Event (sub-event) is tasked to a thread.
  - After the merger, Hit and trajectory objects created by a worker thread will be sent back to the corresponding worker thread for deletion.
- Time scale of the development :
  - Source code : to be finished by May 2019
  - An example : to be finished by September 2019