Implement Event based Seeding and Multi-Threading

By Mohamed Moanis Ali
Project Information

Project: Allpix Squared

Mentors:
- Simon Spannagel
- Daniel Hynds
- Koen Wolters

Student:
- Mohamed Moanis Ali: EMSE Masters Program (University of Bolzano, Italy)

All materials can be found in the submission link here.
Allpix-Squared

- Generic framework for simulating pixel detectors
- Simulates the passage of ionizing radiation through the sensor and finishing with the digitization of hits in the readout chip
- A run consists of several sequential events. A single event refers to an independent passage of one or multiple particles through the detector setup
- The setup contains Modules; algorithm that carries out a specific task in the simulation (eg: Deposition of particles, Digitization of pixel hits, etc...)
Problem Statement

- For better accuracy users are interested in simulating millions of events
- The framework supports an experimental multithreading feature but it doesn't utilize available cores well nor it scales when using more cores
- Experimentation for a new way of multithreading was carried out in GSoC18 and found good potential in terms of speedup and scaling, but of course identified many challenges for implementing this feature
Project Goals

- Fix major challenges identified in GSoC18 and ship the feature:
  - Geant4 Multithreading
  - Buffering Writer Modules
  - Optimizing thread utilization and memory usage
  - Transforming all modules to be thread-safe
  - Refactoring the framework codebase to transparently support the new feature, eg:
    - Messenger
    - Event
    - ThreadPool
  - Testing the new approach and updating documentation and user manual

We will discuss the major goals in this presentation
Problem: Multithreading in Geant4

- Geant4 is a physics library used mainly for the deposition of charge carriers in the sensor.
- All operations must be through its G4RunManager or G4MTRunManager classes.
- These managers can’t be called in parallel.
- G4MTRunManager support multithreading but internally
- Contradicts with the modular design of the framework
Solution: Custom RunManager

- Implemented customized managers for Geant4
- Provides a sequential and multithreaded managers as Geant4
- Respects the framework multithreading design
- Unlike Geant4 implementation, both managers will produce the same results when given the exact same input.
Problem: Writing Output Files in Order

- Some modules exist to write simulation output to file
- They expect events to run in order and to write them in same order
- When using event multithreading, the order of running the events is not guaranteed
Solution: Implement Event Buffering

- Implemented a buffering mechanism where out of order events are stored to be executed later.
- This guarantees the modules will still support multithreading and they will not compromise the correctness of the output.
- The buffering mechanism happens transparently behind the scenes.
- Module's author only needs to inherit from a special class.
Results

- New multithreading feature ready to be shipped in Allpix-Squared v2.0
- Huge performance boost and near linear scalability
- Almost all modules support multithreading
- Refactored some of the major components of the framework
- Added new test to gauge framework multithreading capabilities
- Performance analysis using vTune Amplifier profiler
Execution Time comparison between old multithreading and new multithreading approaches
Future Work

- Reduce the overhead of buffering the modules
  - Re-submit events to threadpool again!
- Investigate the dependency on ROOT TRef
  - Latest versions of ROOT (especially TRef) show performance issues with Allpix-Squared
  - Problem was reported to ROOT forum
Thanks!