Generic Tasking Framework for Geant4

Author: Jonathan R. Madsen
Institution: Lawrence Berkeley National Laboratory
Email: jrmadsen@lbl.gov

This letter describes the generic tasking framework which will be implemented in the upcoming release of Geant4. Although the typical Geant4 simulation is, in general, embarrassingly parallel, there are certain workflows which can benefit from an asynchronous, load-balancing execution scheme. In particular, workflows seeking to leverage heterogeneous architectures and workflows which are heavy on analysis relative to transport will benefit. In the former case, these workflows will be able to create separate thread-pools whose threads are dedicated to managing the communication to and from the secondary execution device (e.g. GPU, FPGA) while minimally impeding the performance of the computations on the CPU. In the latter case, a tasking system will improve load-balancing by enabling the user to utilize CPU cycles which would otherwise be idle.

The tasking system is designed to be simplistic and generally compatible with other tasking systems, such as Threading Building Blocks (TBB). The tasking system is separated into 3 primary categories: (1) the thread pool which contains the set of threads which will be responsible for executing the tasks provided by the scheduler, (2) a task queue which manages the set of tasks which must be processed by the queue, and (3) a task group, which is a collection of tasks artificially separated from other tasks in the same queue. When compatibility mode is enabled with another tasking framework, the external tasking framework is responsible for (1) and (2) and the system only provides (3).