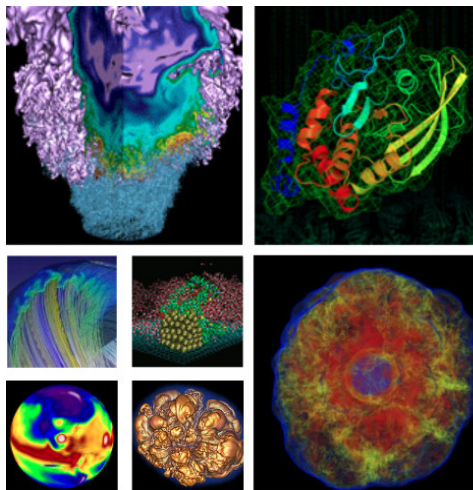


Geant4 + GPU

Geant4 + Tasking + GPUs



National Energy Research
Scientific Computing Center



U.S. DEPARTMENT OF
ENERGY

Office of
Science



Jonathan R. Madsen

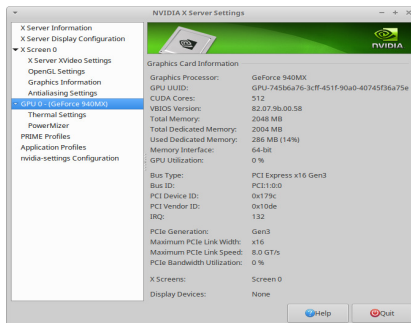
✉ jrmadsen@lbl.gov

National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory

August 30, 2018

- This is not a presentation about an existing GPU implementation
- Interested in discussion/collaboration about using tasking for efficient GPU off-loading

- Run fibonacci calculation on CPU
 - High CPU usage
- Run complex number calculation on GPU
 - Significant memory allocations
- Intel i7 CPU (4 HW threads) + NVIDIA 940MX GPU (512 cores + 2GB)
- 8 threads (4 for CPU, 4 for GPU)



- Full CPU utilization from fibonacci calculations
- GPU running a complex number calculation provided by colleague
- Concurrent GPU utilization verified
 - stdout displays output from CPU and GPU
 - CUDA visual profiler inspected \Rightarrow executing on GPU



- Tasking provides “easy” way to:
 - Dynamically off-load work to GPU
 - Work on the CPU and GPU concurrently
 - Retrieve work from GPU when it is finished
 - Queue up work on GPU without blocking CPU work

- Current idea
 - Thread-pool for CPU tasks \approx number of HW threads
 - ▷ Computations that do not off-load to GPU well
 - ▷ Sends work to GPU, when available
 - ▷ Executes work on CPU when GPU not available
 - Thread-pool for GPU tasks \gg number of HW threads
 - ▷ Bundle work via “task-groups” to run on GPU
 - Initial work
 - ▷ Dynamically decide whether to off-load based on amount of work
 - ▷ Kernel to calculate all geometrical values (safety, etc.)
 - ▷ Kernel to calculate EM cross-sections
 - ▷ Kernel to calculate ... suggestions?
- Thoughts?