GPUs in Geant4

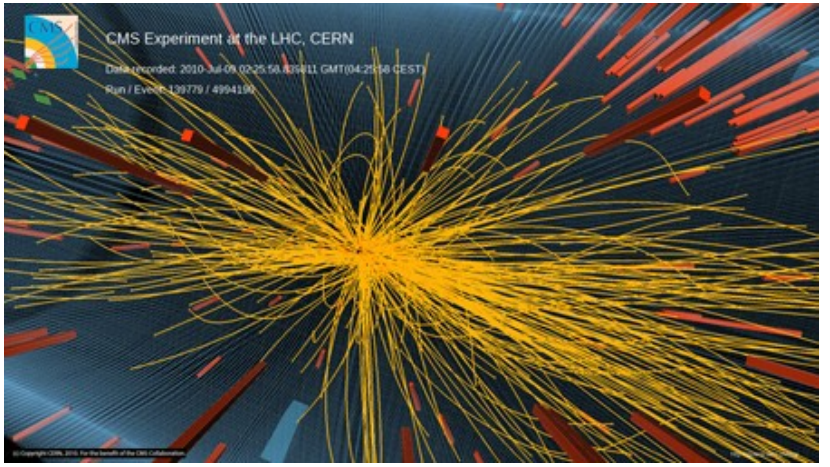
Philippe Canal, Daniel Elvira, Soon Yung Jun,
Jim Kowalkowski, Marc Paterno
Fermilab

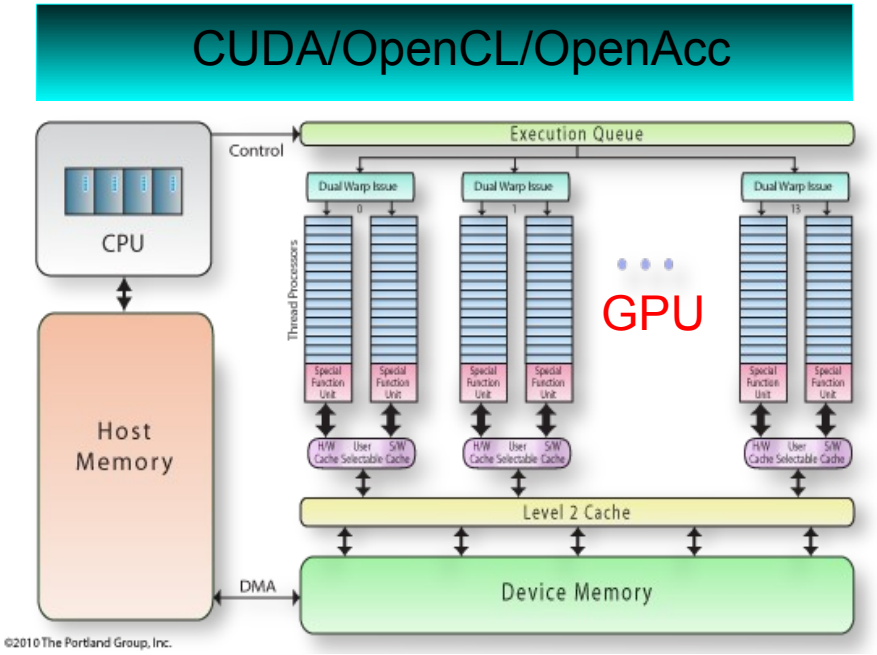
John Apostolakis
CERN

Annual Concurrency Meeting
February 4-6, 2013
Fermilab

Detector Simulation on GPU?

- Geant4 HEP applications
  - highly sequential/event-level
  - complex/memory intensive
- Rules of GPU computing
  - minimize data transfer
  - keep arithmetic intensity
  - re-use data
- Major challenges
  - kernel divergence
  - memory bandwidth
  - work balance (CPU-GPU)





Problem Statement

- A massively parallel particle transportation engine
- Requirements
  - precision (double)
  - magnetic field (slow varying)
  - EM physics only (electron, photon)
  - realistic geometry/data
- Algorithms, core methods and implementation
  - adaptive Runge-Kutta (4^{th} order)
  - voxelized geometry/navigation
  - electron and photon processes for HEP
  - generic codes for future flexibility (CUDA, openCL, ...)

Hardware: Host and Device

- Host: AMD Opteron Process 6136
  - CPU: 2.4 GHz, 4 Processors x 8 cores: 32 CPU cores
- Device: NVIDIA Tesla M2090
  - GPU: 1.3 GHz,16 Multiprocessors x32 cores: 512 CUDA cores
- Performance measurements in execution time (T)
  - number of threads per grid: 32 (blocks) x 128(threads)
  - 100K tracks
  - Tc = time with 1 CPU core
  - Tk = time with 512 CUDA cores (kernel execution)
  - Td = time for data transfers (memory allocations, H2D, D2H)
  - Speedup G_T = Tc/(Tk+Td)

Arithmetic Intensity

- (arithmetic instructions)/(off-chip memory operands)
  - occupancy = (resident warps)/(maximum warps)
  - latency = clock cycles for a warp to execute the next instruction
- Performance of numerical algorithms

Numerical Algorithm	Tc (ms)	Td (ms)	Tk (ms)	Tc/Tk	G_T
Classical Runga-Kutta	263	5.18	2.67	99	34
Runga-Kutta Felhberg	279	5.18	2.31	121	37
Nystrom Runga-Kutta	72	5.18	0.64	113	12

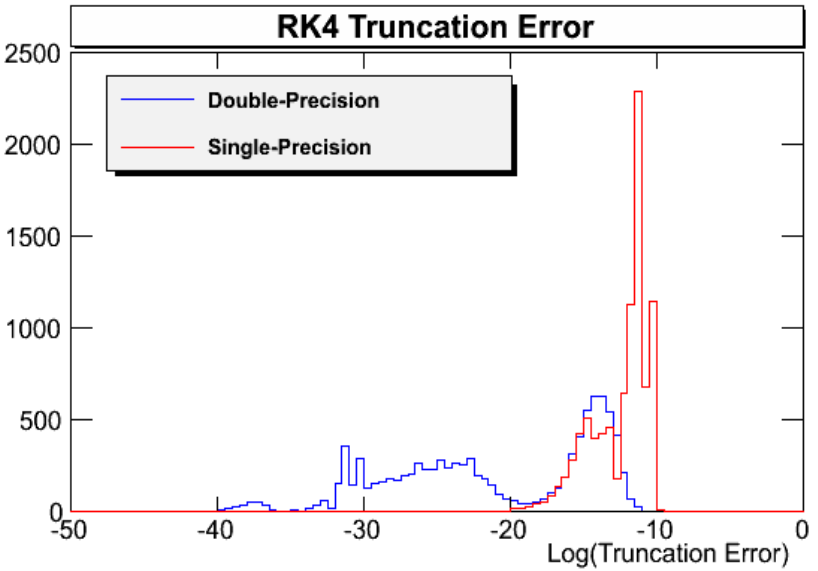
- Adaptive step control is not enough to keep GPU busy
  - kernel optimization may not be important if Td < Tk
  - multiple stepping (physics)
  - navigator (geometry)

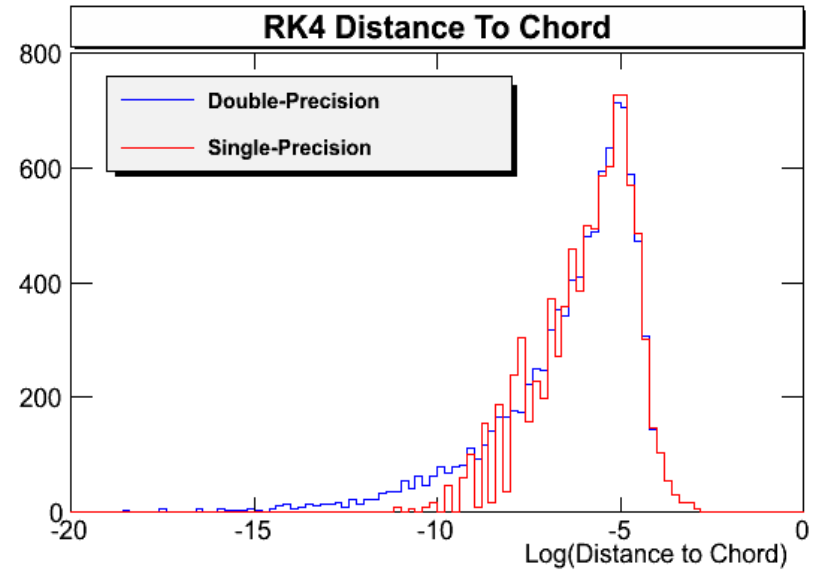
Floating Point Consideration

- float vs. double
- Cost for double
  - memory throughput (x2)
  - registers spilling
  - arithmetic instructions
- One step for 100K tracks

G_T	RK4	RKF45	Nystrom
float	76	84	25
double	34	37	12

- Robustness is not negotiable (adequate precision)





EM Physics on GPU

- Most of secondaries are electrons or photons
- Implementing EM physics on the GPU
  - increases computational intensity
  - enable to make multiple stepping possible within GPU
- Mainly converted by D. Jang (CMU, now at Nokia)

Kernel (EM process)	CPU(ms)	GPU (ms)	G_T
Bremsstralung	343	8	42
Ionization	129	6	21
Multiple Scattering	40	5	8
Combined EM kernel	297	11	28

- Challenge: handling secondaries and hits

Geometry

- Geometry is necessary for
  - EMPhysics: material (mean free path)
  - Transportation: navigator, intersection point
- Create a navigator per thread on GPU and reuse it
  - re-initializing navigator per track with device geometry on the host and copy to the device is expensive
  - avoid large latency in global memory access
- Performance depends on the complexity of geometry

SimpleCalo (nphi,nz)	CPU (ms)	GPU (ms)	G_T
Navigator::ComputeStep	309	10	31
Estimate Intersection Point	1485	32	46

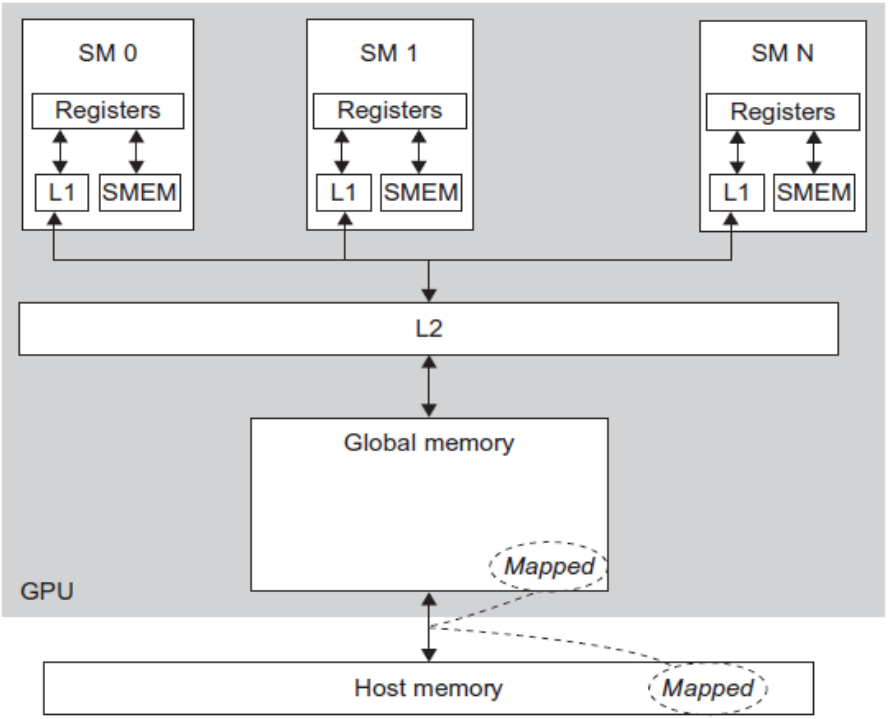
Random Number Generators

- Period, statistical properties, memory, efficiencies
- CUDA PRNG library (CURAND)
  - XORWOW (xor-shift family)
  - MRG32k3a (L'Ecuyer's Multiple Recursive Generators)
  - MTGP32 (Mersenne Twister, 32bit, period 2^11213)
  - (SOBOL quasi-random generators)
- Performance (64 blocks x 256 threads)
  - separate state setup kernels for maximum performance

CURAND PRNG	Init States (ms)	PRNG for 10K (ms)
XORWOW	4.09	7.55
MRG32k3a	6.16	22.17
MTG32	0.69	35.96

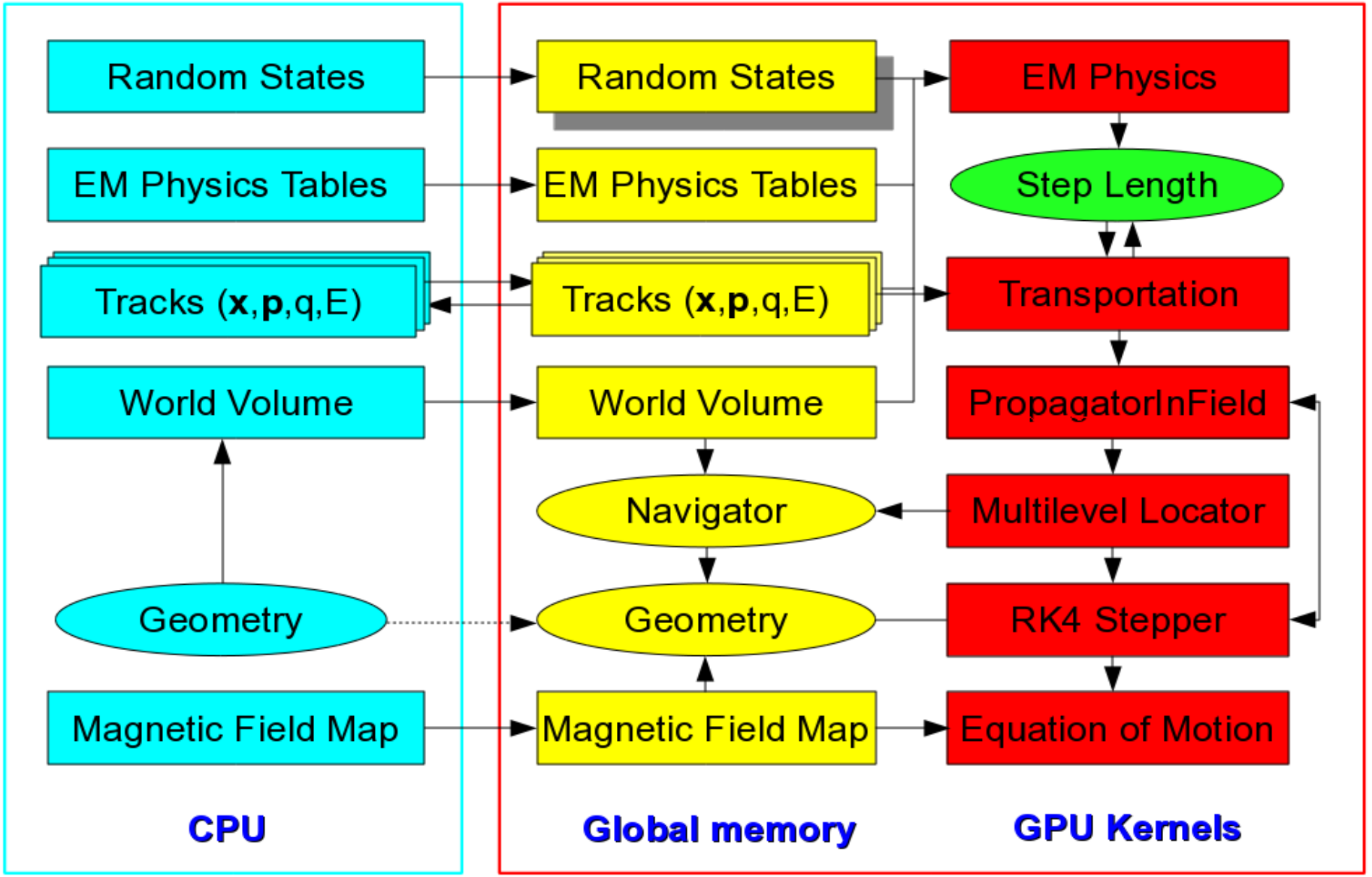
Memory Bandwidth Implications

- Data transfer (PCIex2)
  - use pinned memory
  - batching data transfer
- Global memory access
  - coalesced access
  - minimize register spills (kernel decomposition)
- Data structure
  - texture for B-field map
  - AoS vs. SoA for tracks
  - exploit data locality



Memory	Bandwidth (GB/s)
Register	8,000
Shared	1,600
Global	177
Mapped	8

EM Transportation



Streams and Concurrent Kernels

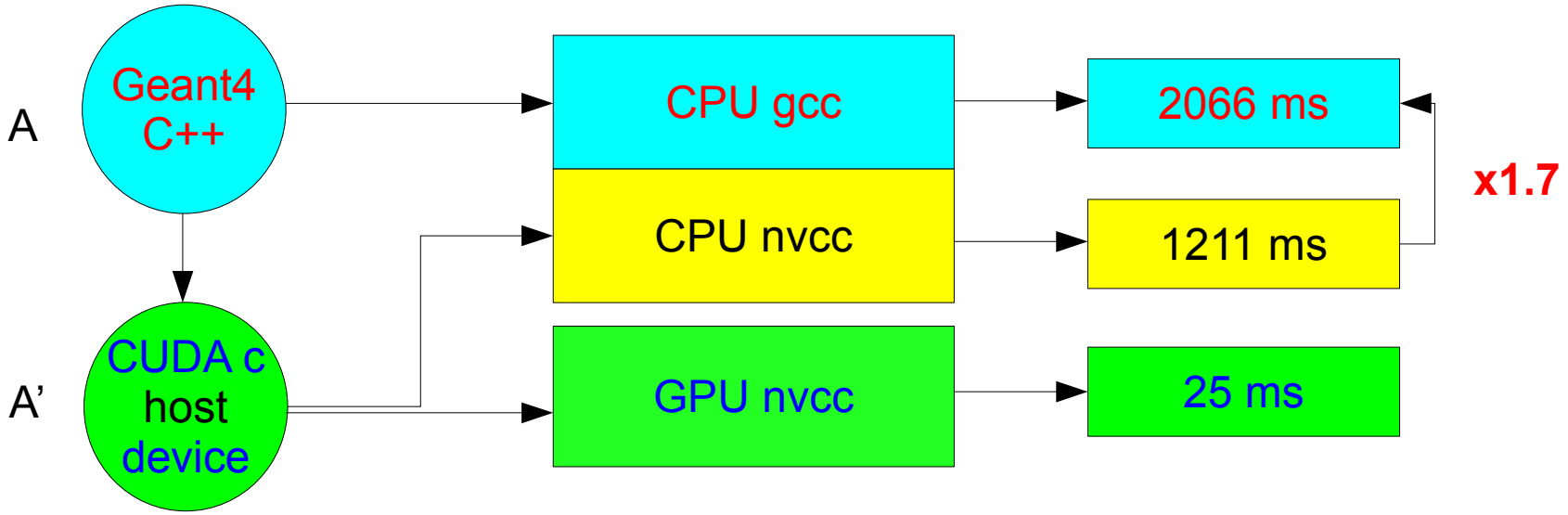
- Multiple streams: task parallelism
  - multiple CUDA operations simultaneously
  - one (two) for data transfers, others for kernel executions
- Decompose algorithms with multiple kernels
  - increase arithmetic intensity, but avoid kernel divergences
  - reduce a performance cliff from register or local memory spills
- Develop concurrent computing models
  - load balancing between CPU and GPU
  - interface to a track dispatcher

Performance Evaluation

- One step transportation for electron and photon tracks
  - a simple calorimeter (phi-z segments)
  - multiple streams and multiple kernels
- Transportation kernels
  - random states (per thread)
  - (electron + photon) kernels with asynchronous data streams
- Performance

	CPU (ms)	GPU (ms)	G_T
Transportation (1 kernel)	1201	32	37
separated e/g (2 kernels)	1211	25	49

Performance Comparison to Geant4



- CUDA host code runs 1.7 times faster than G4 equivalent
- Differences between A (Geant4 C++) and A' (converted c)
  - reduce number of temporary objects creation/destruction
  - removal of generalization, including unrolling some functions calls, removing virtual
  - reorganization of the data layout

Cost Differential

- Hardware cost
  - $(estimated NVIDA M2090) = ~$2500
  - $(estimated AMD 4-CPUs)/(CPU cores) = ~$3200/32 = $100
- Performance factors
  - G_T= 49
  - power consumption: equivalent
- Per unit of work, the GPU costs
  - (2500/100)/G_T/1.7 = 25/49/1.7 = 0.29
- Huge room to improve
  - optimization (global memory access, re-use data)
  - completed EM engines (arithmetic intensity, multiple stepping)

Future Direction

- Decompose and optimize sub-components
  - base (data manipulation, random engines, utility functions)
  - EM Physics (complete electron, photon processes)
  - geometry (explore locality and other options)
  - transportation chains
- Extend the GPU prototype to large vector machines or hybrid systems
- Connect to track dispatchers (ex: vector prototype) and demonstrate a definitive speedup