Particle Transportation on GPU

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

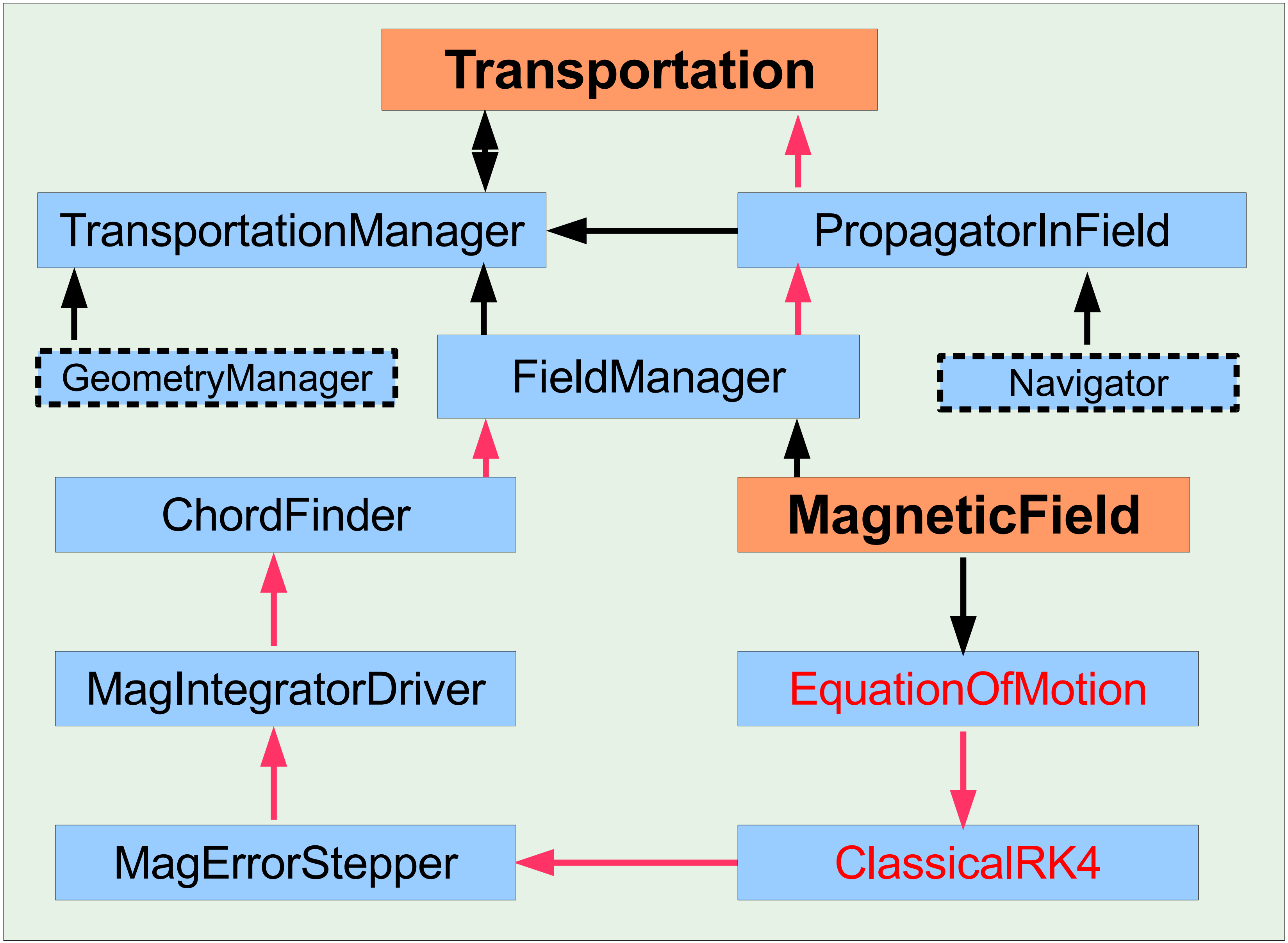
Dongwook Jang
Carnegie Mellon University

Geant4 Collaboration Meeting
September 10-14, 2012
Chartres, France

Concurrent Particle Transportation

- Hardware architecture is shifting to multiple/many cores
- Concurrency is the key word for future hybrid systems
- How can we use many-core for HEP/NP simulation?
- Geant4 performance studies with the CMS detector shows that core components of particle tracking are
  - geometry look-up (navigation, material, B-field)
  - physics models (cross section and energy loss)
  - particle transportation
- Concurrent particle transportation engine
  - study Geant4 particle transportation on GPGPU
  - optimize the transportation process for GPGPU

G4 Transportation for Charged Particles in B-Field (AlongStepGPIL)



Particle Transportation

- Particle trajectory: equation of motion in a magnetic field

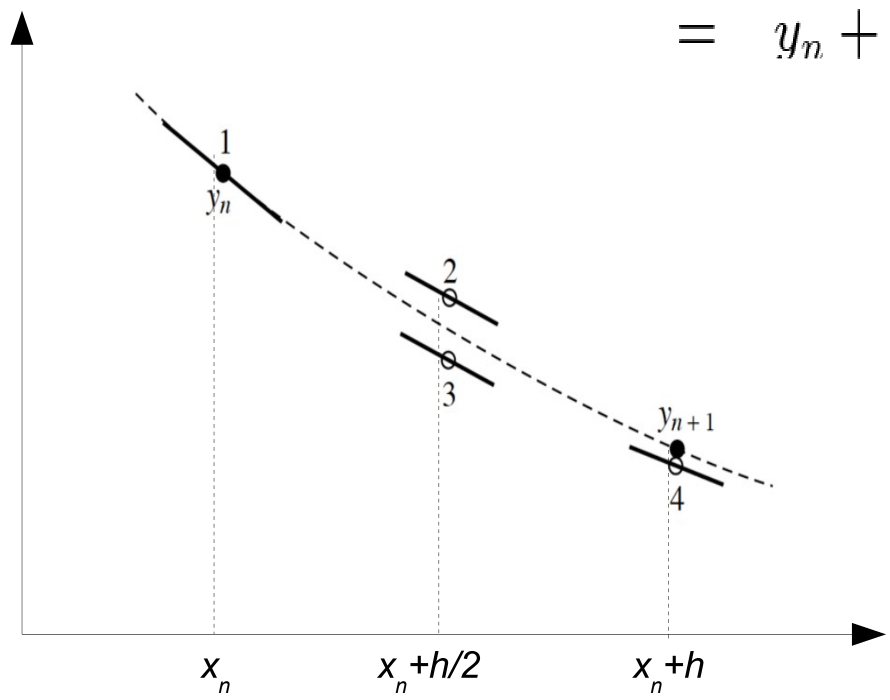
$$\frac{d^2\vec{x}}{ds^2} = \frac{q}{p}\frac{d\vec{x}}{ds} \times \vec{B}(\vec{x}) \quad \rightarrow \quad \frac{dy}{ds} = f(x, y), \qquad y(x_0) = y_0$$

- Problem definition
  - magnetic field calculation
  - ODE solver: steppers
  - decision trees for accurate advance

ODE solver: Runge-Kutta Method

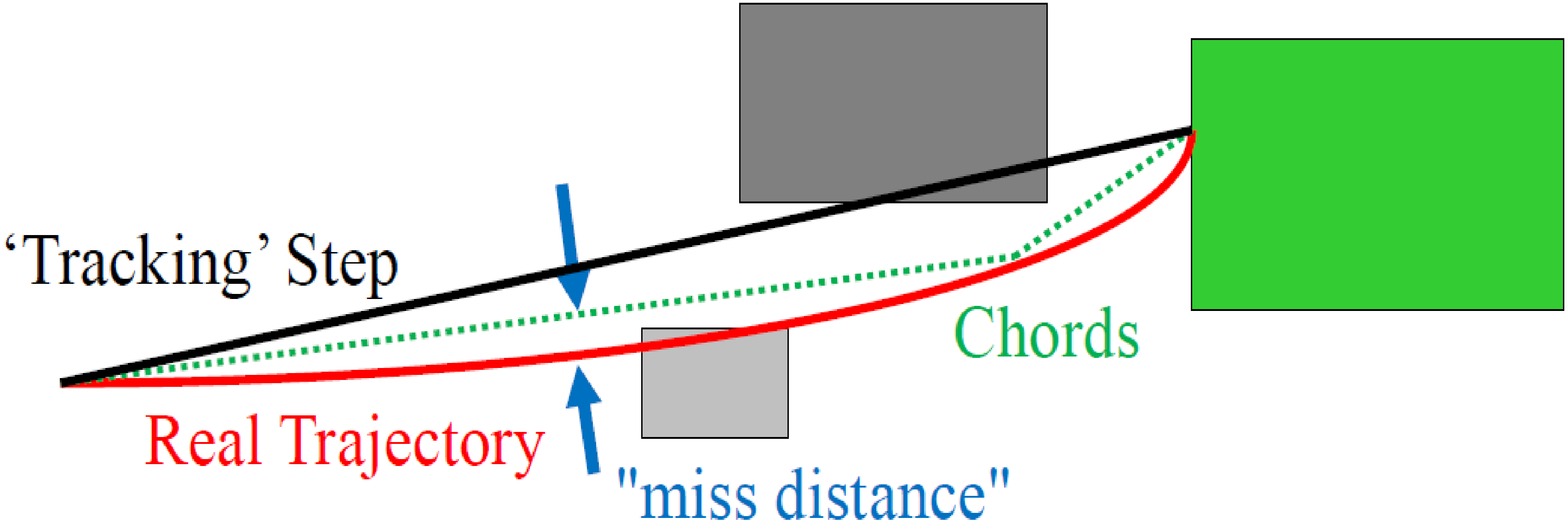
- Update a state of particle (x,p,spin,time) for each step
- 4-th order Runge-Kutta (RK4): 4 evaluations of f(x,y)

$$
\begin{aligned}
y_{n+1}(x_n + h) &= y_n + hf(x_n, y_n) \\
&= y_n + \frac{h}{6}\sum_{i=1}^{4}[f(x_n + \alpha_i, y_n + \beta_i)] + \mathcal{O}(h^5)
\end{aligned}
$$



Adaptive Step Size Control

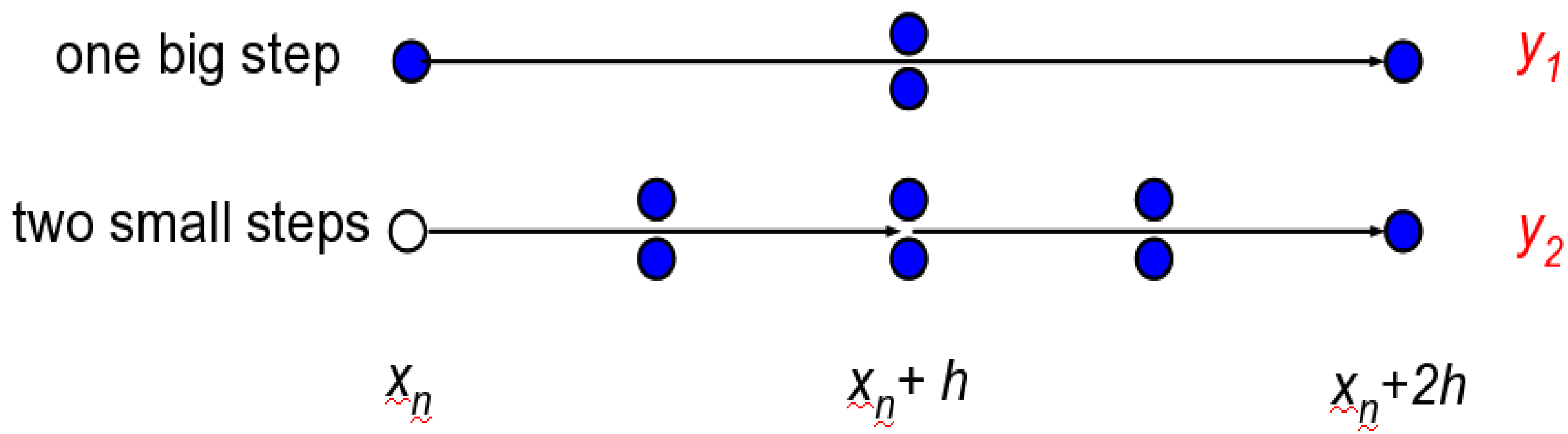
- Miss distance: maximum closest distance from the curved trajectory to the chord.



- Quick advance if miss distance < δmax otherwise, computer a new step

Adaptive Step Size Control

- Truncation errors of step doubling in RK4: difference between one big step and two small steps



$\Delta = y_2 - y_1$ = truncation error

- Accurate advance if the truncation error < $\delta\varepsilon$ otherwise computer a new step

Hardware: Host and Device

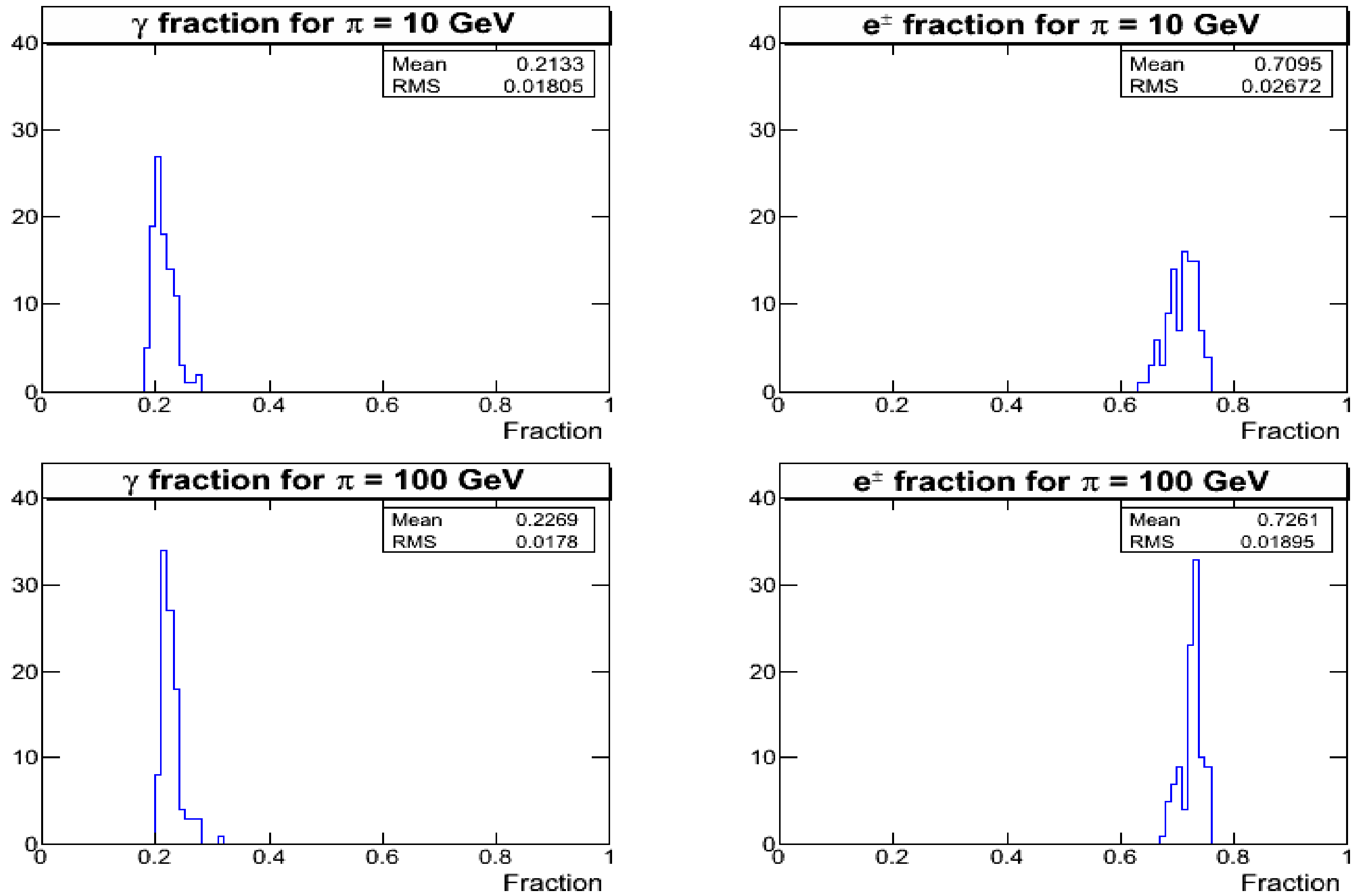
- Host: AMD Opteron Process 6136
  - CPU: 2.4 GHz, 4 Processors: total 32 cores
- Device: NVIDIA Tesla M2070
  - GPU clock speed: 1.15 GHz
  - 14 Multiprocessors x 32 CUDA Cores: 448 CUDA cores

Software: Interface and Device Codes

- cmsExp
  - Geant4 application to study event characteristics of high energy particle interactions with a real experiment
  - CMS geometry and magnetic field map
  - an interface to prepare input data and to test GPU codes
- GPTransportation
  - device/host functions invokable from GPU kernels or CPU
  - EM physics, Geometry and Transportation
  - a test bench to study and evaluate performance
- ClassicalRK4
  - heart of the transportation algorithm optimized for GPU

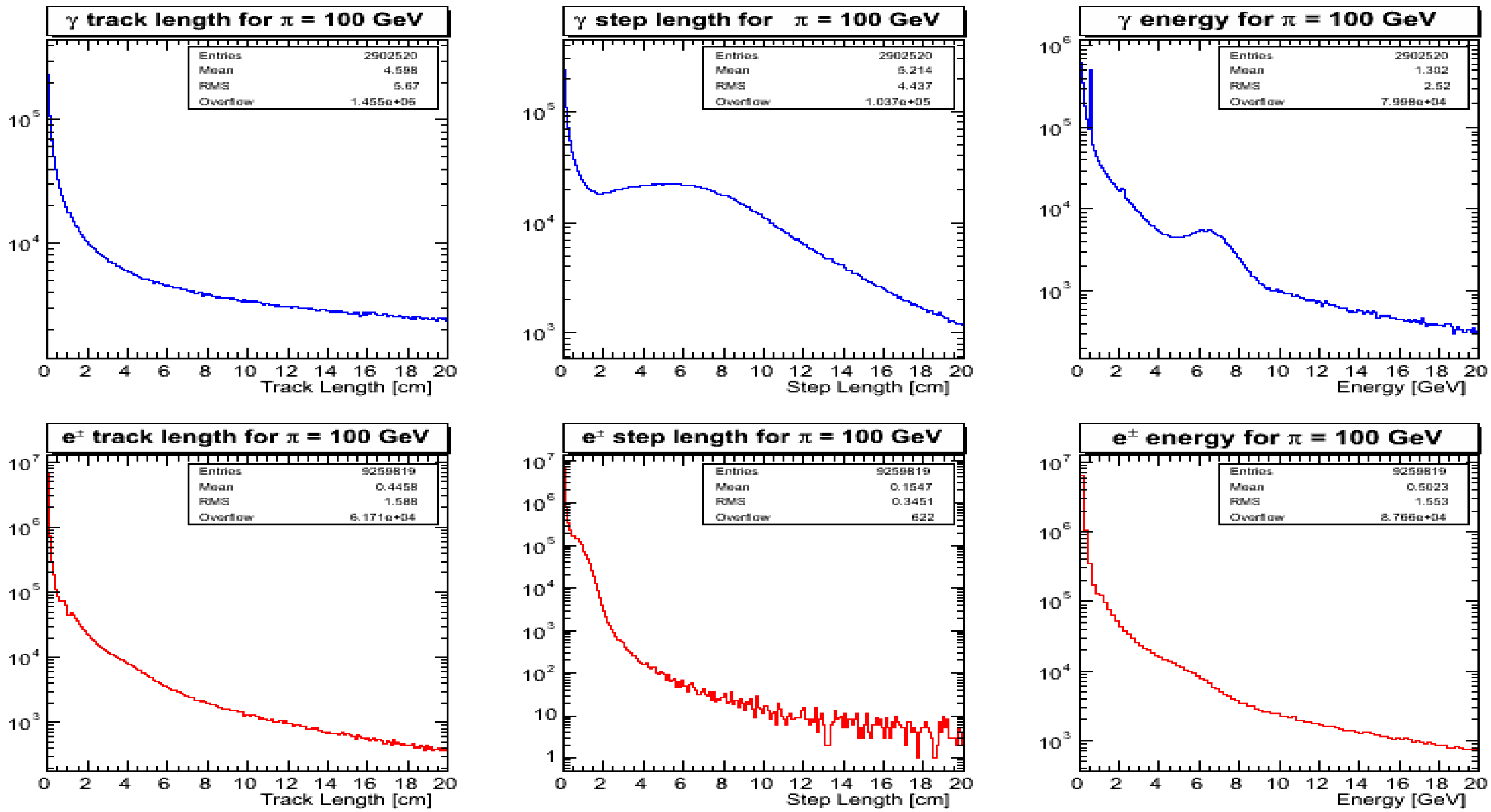
Input Data

- Secondaries produced by single pions (10,100 GeV)
- Majority are electrons (75%) and photons(20%)



Input Data: Step Length and Energy

- 90% of electron tracks have one step with size << 1cm
- Energies are very soft (E << 1 GeV)



Performance Measure

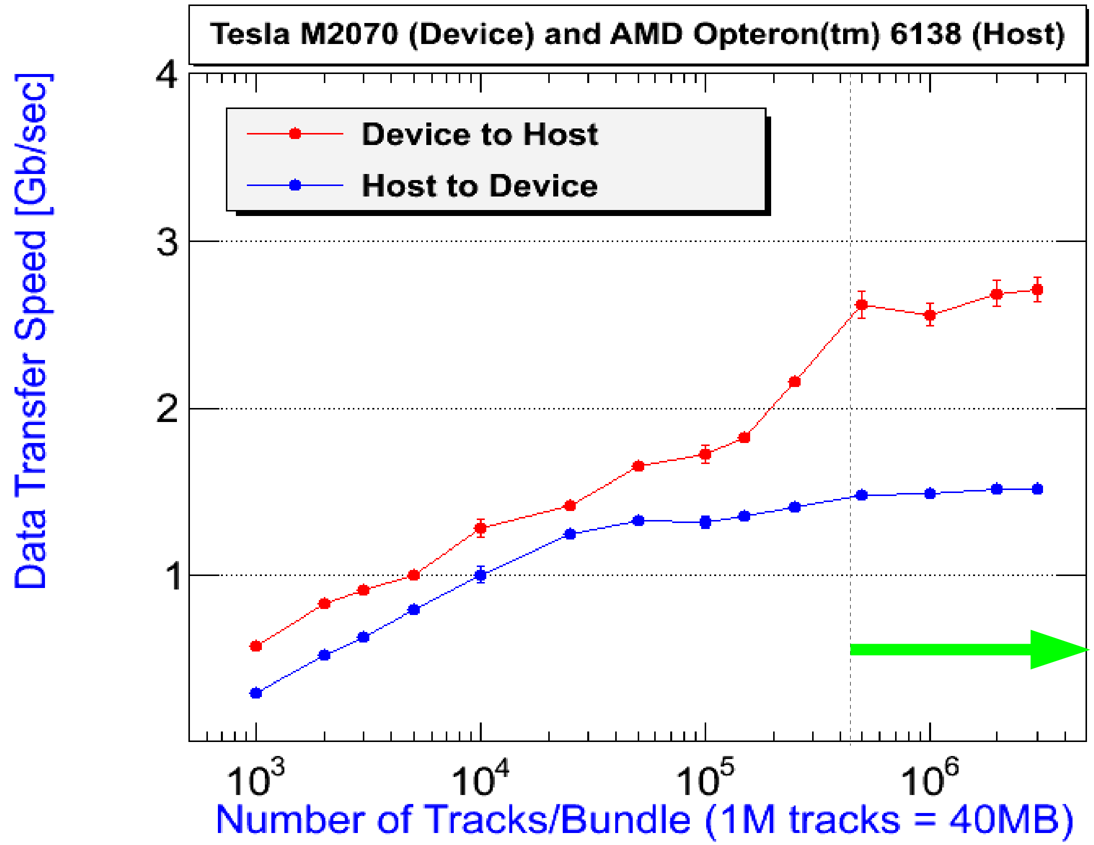
- Performance measurements in execution time
  - 1 CPU vs. 448 GPU cores
  - CUDA event timer
  - GPU time = kernel execution + data transfer
  - default cuda kernel: blocks=N, threads=128
- Performance evaluations focused on
  - data locality
  - arithmetic intensity
  - throughput
  - concurrent streamer

Performance Measure: Input Data

- CPU/GPU (ratio of processing time) of the transportation process for one step advance for all secondary particles produced by 100 GeV pions
  - electrons only
  - position and momentum
  - use the step length if nstep=1 (90%) else use the average step length (= total track length /nsteps) (10%)
- Test performance
  - Stepper: three evaluation of the4th order Runga-Kutta (RK4)
  - Geometry and EM Physics
  - full chain of transportation with all decision trees
- For the sake of comparison, performance gains with a fixed step length (1cm) are also measured

Performance: Data Transfer Rate

- Data transfer for track bundles between host and device



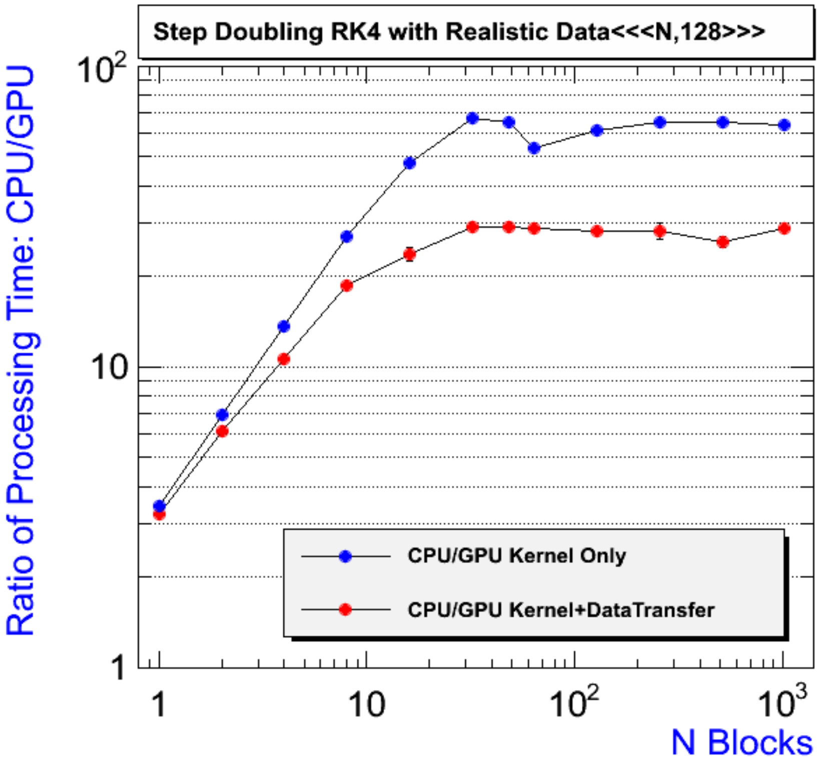
- Minimize data transfer between host and device
  - bandwidth device-device is $O(10^2)$ (GB/sec)
  - one large transfer is better than many small transfers

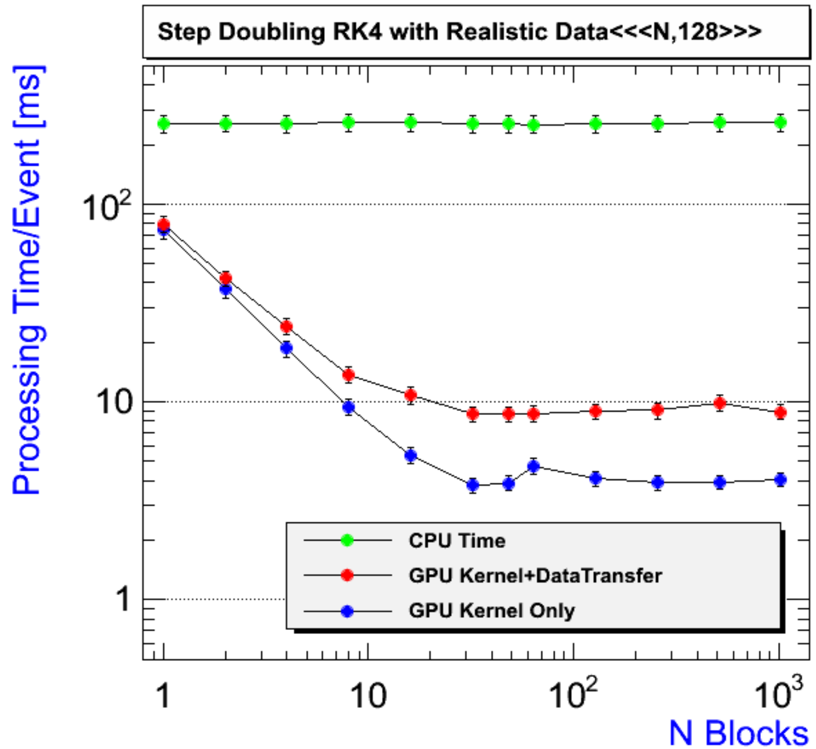
4^{th} Order Runga-Kutta

- Key components of the transportation stepper (RK4)
  - evaluation of magnetic field (B) values
  - rhs of the equation of motion for a given B
  - evaluations of the 4^{th} order Runge-Kutta
- Measure performance with the Runga-Kutta driver for the adaptive step size control (three evaluations of RK4)
- Test Geant4 transportation with realistic data
  - prepare bundles of (secondary) tracks produced by single particles (100 GeV pi-) passing through the CMS detector
  - measure processing times for AlongStepGPIL on CPU and GPU

4^{th} Order Runga-Kutta with Realistic Data

- RK4: Time (kernel only) vs. Time (kernel+data transfer)





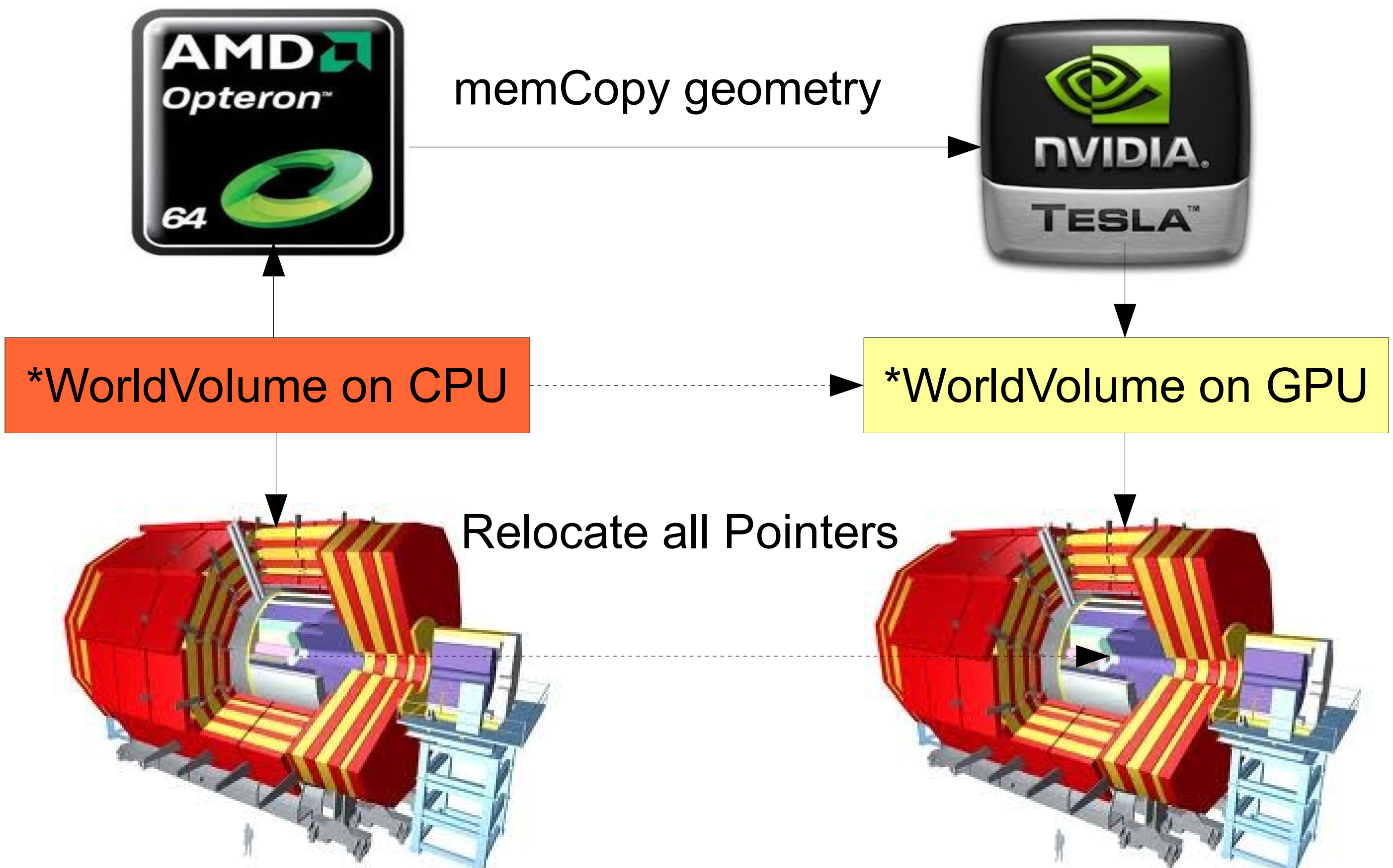
- Optimize kernel execution
  - overall (kernel+data)/kernel > 2 for adaptive RK4
  - minimize data transfer between host and device

Geometry on the GPU

- Port a small subset of the Geant4 geometry to CUDA
  - navigator and multilevel locator
  - basic solids with physical/logical volume
- The primary goals are to
  - support the transportation of neutral particles (photons) and charged particles without a magnetic field
  - complete charged particle transportation in a magnetic field including the geometrically limited step control
  - enable EM physics processes to compute the proposed step length on the device
- Construct a detector with several solids (box, tubs, trd) on CPU and transferred to the global memory on GPU

Geometry Build and Relocation to GPU

- Adapted from the previous work by CERN (Otto Seisaski and John Apostolakis)
- Relocate pointers of physical volumes onto GPU

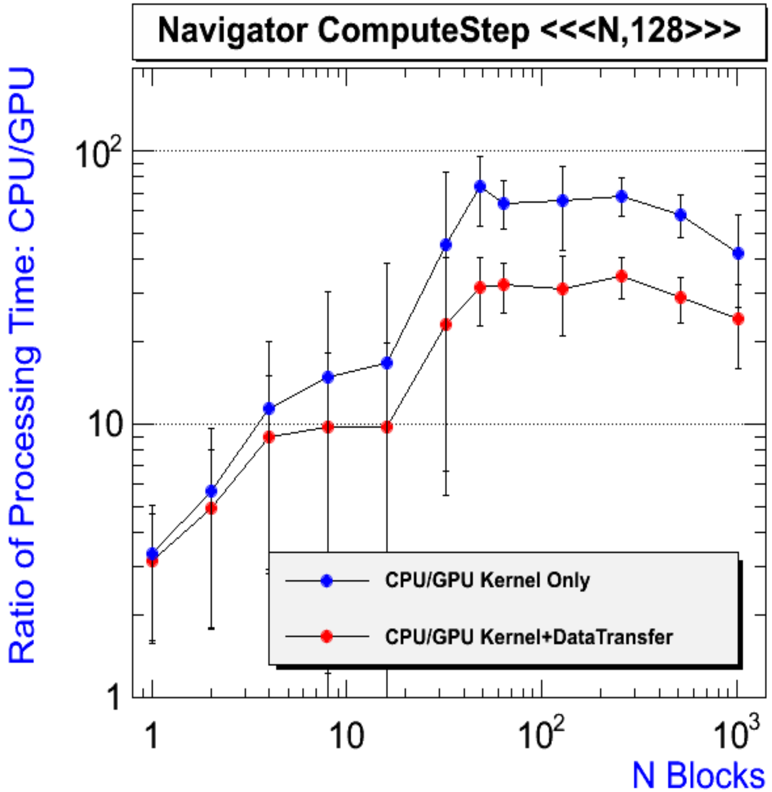


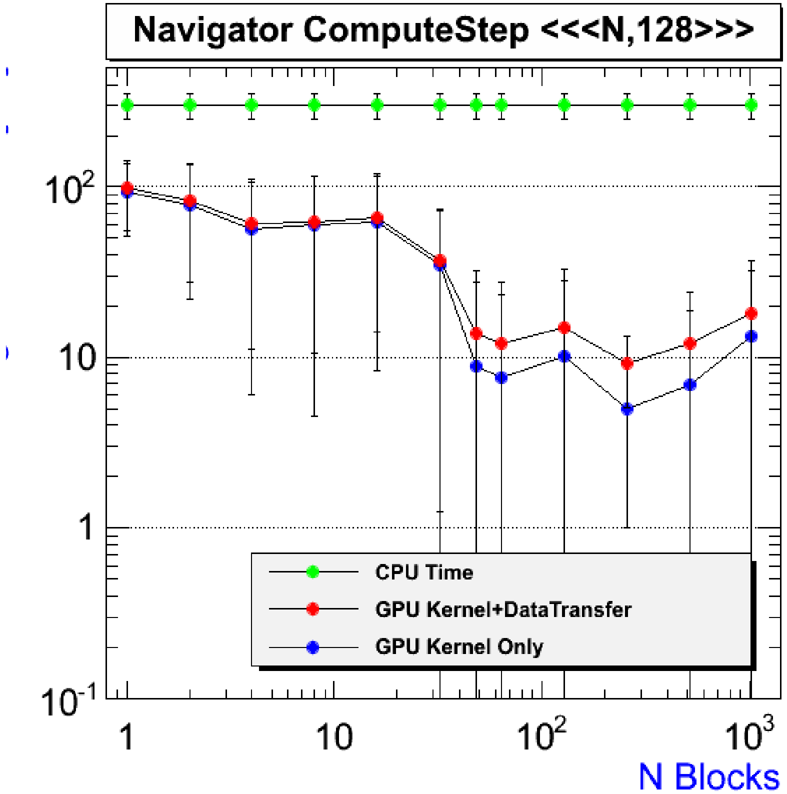
Geometry Tests

- Construct a simple detector
  - worldvolume: 1 box
  - Calorimeter: 3 dz x 4 dphi of tubs (PbWO4)
- GPU kernel includes
  - ComputeStep method of navigator
- Tests with electron (secondary) tracks from 100 GeV pi-
  - position and direction from input data
  - realistic step size
  - a fixed step (1m)
  - number of volumes

Geometry with Realistic Data

- Navigator::ComputeStep with realistic step sizes

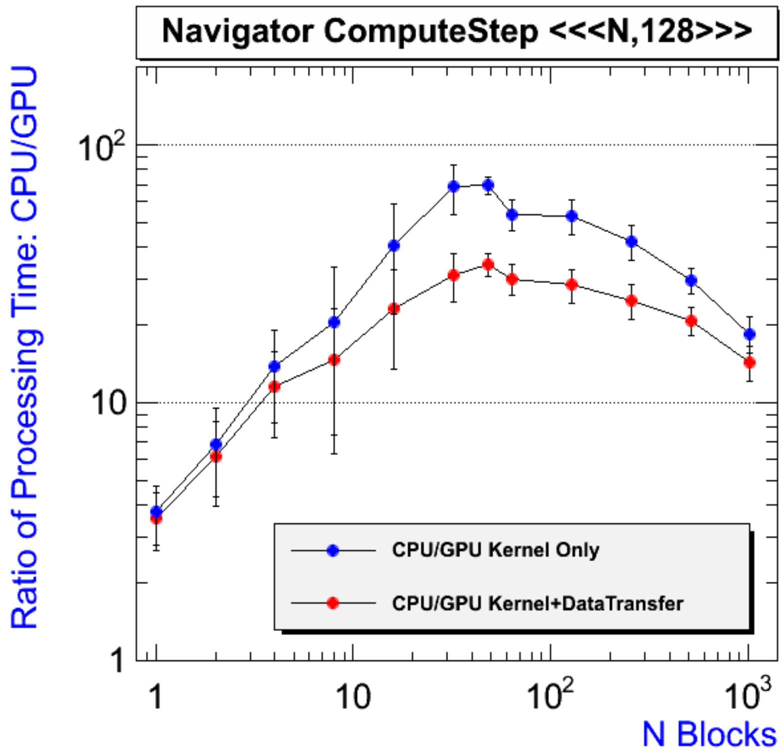


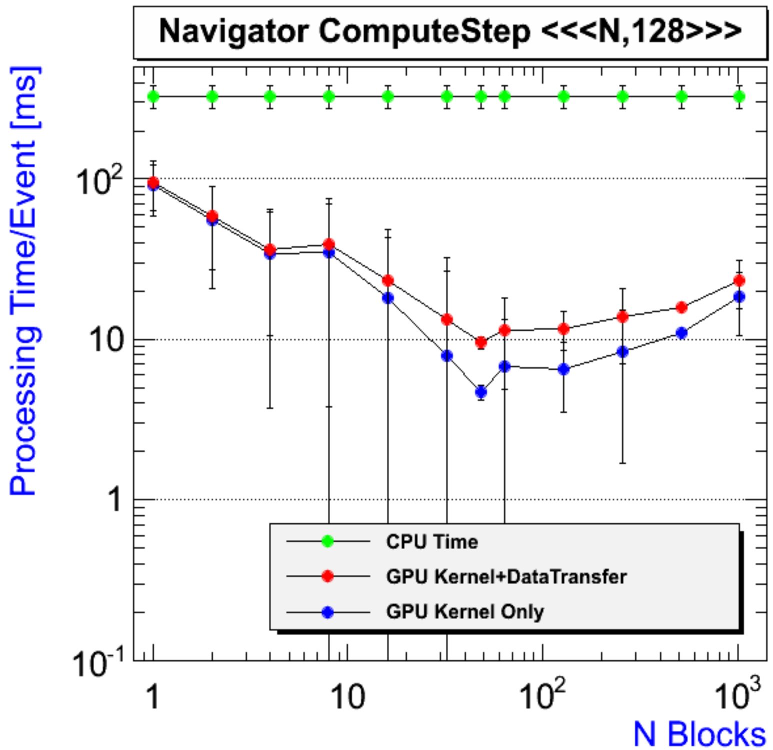


- For the more number of volumes, processing times are increased, but ratios of CPU/GPU stay at the same level

Geometry with a Fixed Step Size

- Navigator::ComputeStep with a fixed step size (1m)

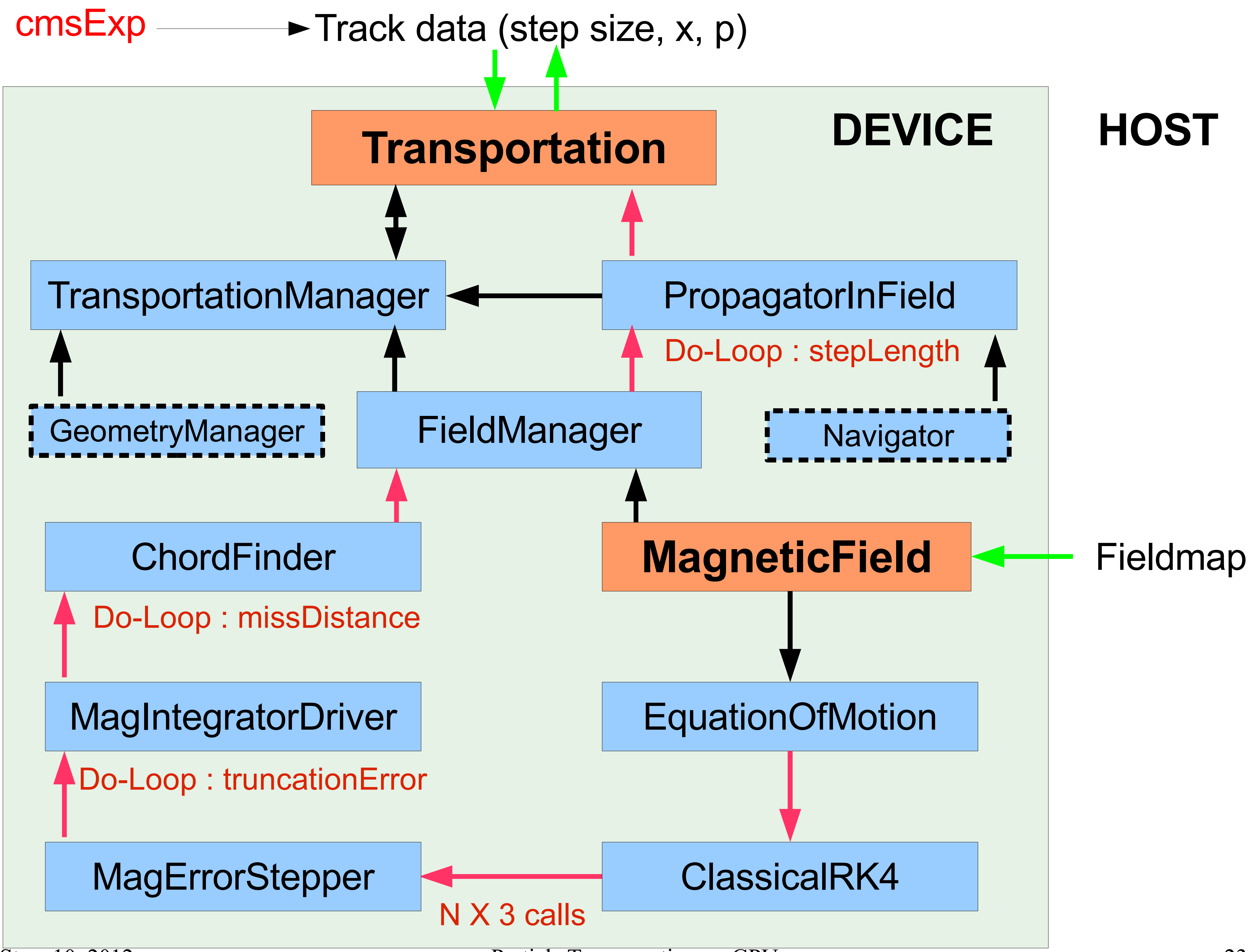




- Large fluctuations in GPU time measurements except for N Blocks = 48, 512

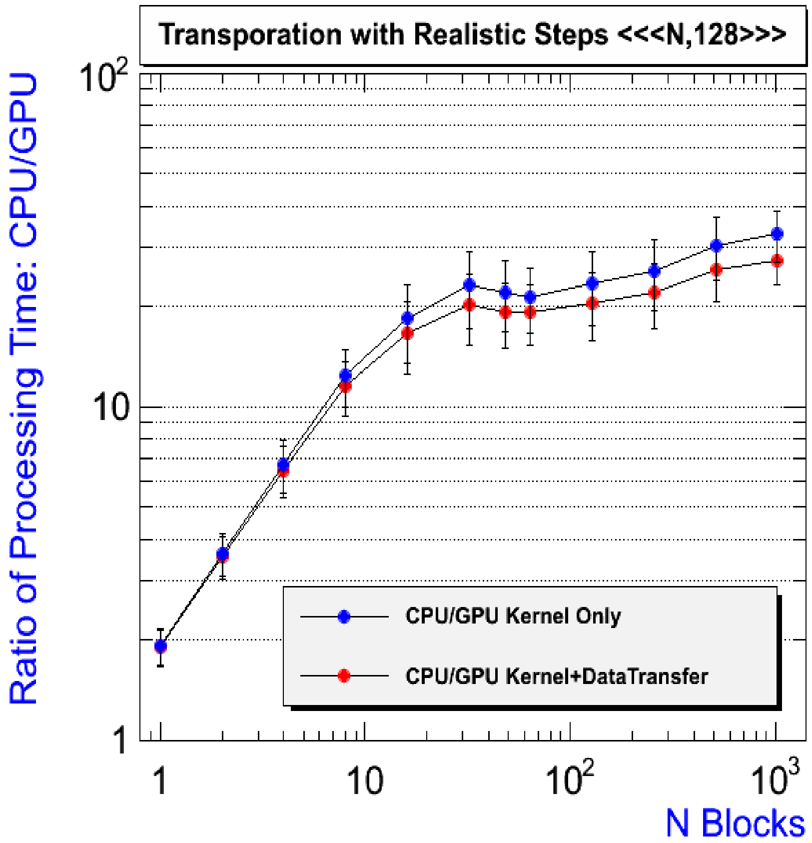
Transportation with Geometry

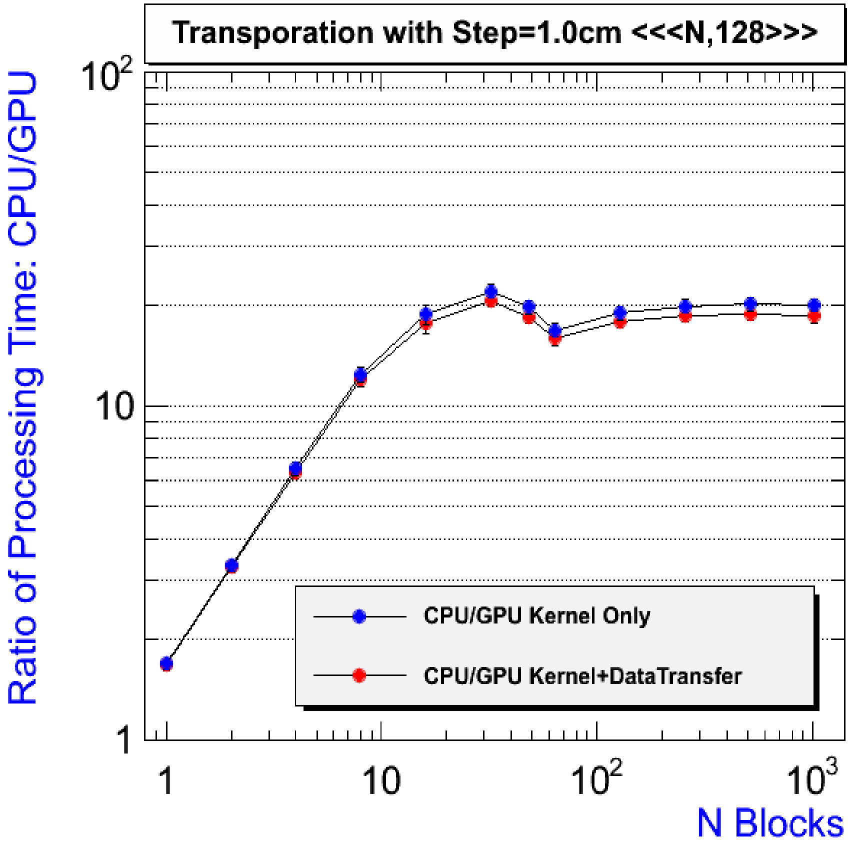
- Full chain of the AlongStepGPIL of the transportation process with a magnetic field map
  - propagator in field
  - chord finder and magnetic error stepper
  - 4^{th} order Runga-Kutta
- Limit the proposed step length with a simple geometry
  - navigator
  - multilevel locator


cmsExp
Track data (step size, x, p)
DEVICE
HOST
Transportation
TransportationManager
PropagatorInField
Do-Loop : stepLength
GeometryManager
FieldManager
Navigator
ChordFinder
MagneticField
Fieldmap
Do-Loop : missDistance
MagIntegratorDriver
EquationOfMotion
Do-Loop : truncationError
MagErrorStepper
ClassicalRK4
N X 3 calls

Transportation with Geometry

- Realistic step size (left) and a fixed step size (right)



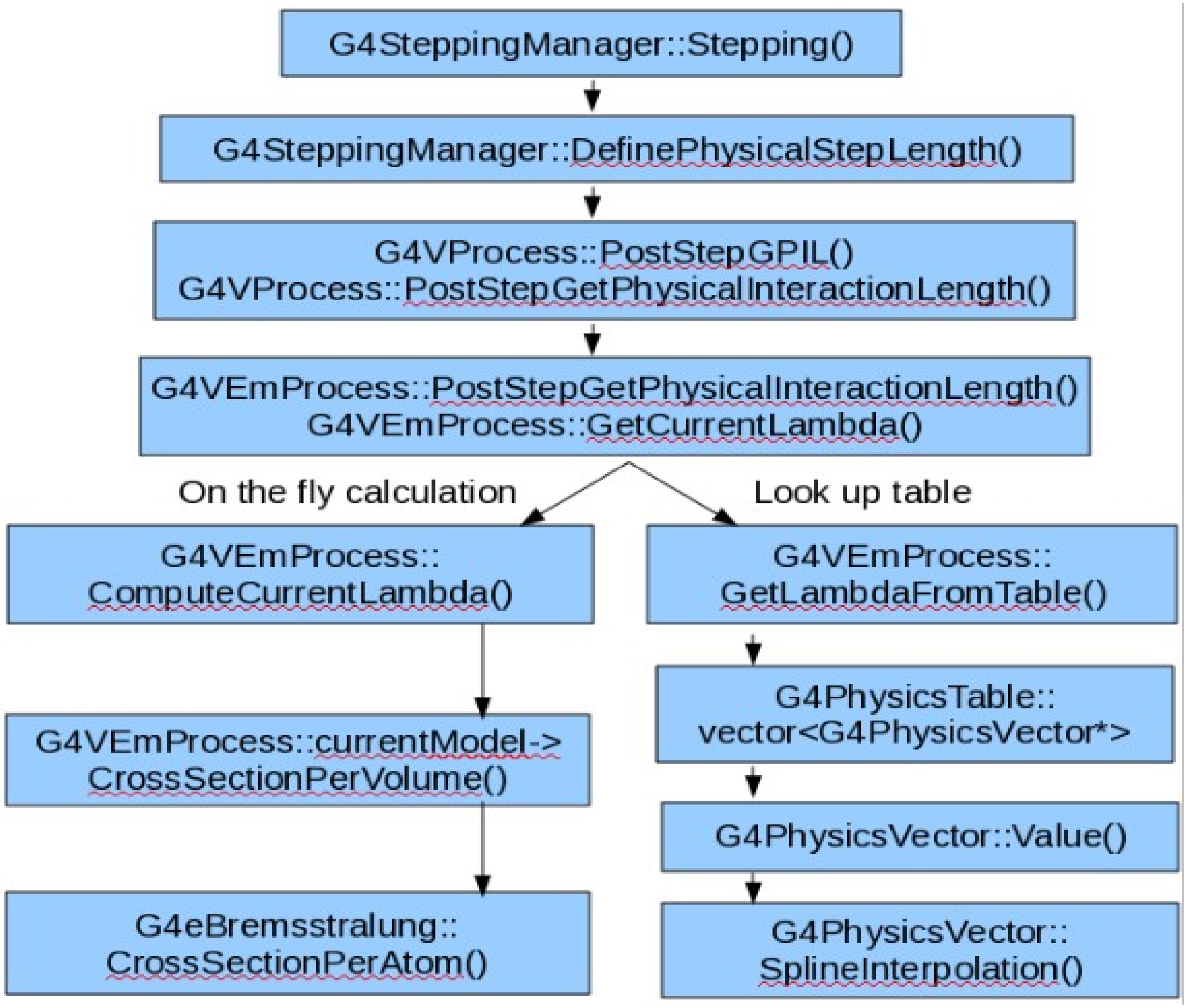


EM Physics on GPU

- Most of secondaries are electrons or photons
- Implementing EM physics on the GPU may
  - increase the computational intensity
  - Enable to make a multiple stepping possible
- CUDA codes are mainly developed by Dongwook Jang (Carngie Mellon University)
- Currently, only Bremsstrahlung is implemented
- EM physics kernel will include
  - PostStepDoIt process, handling secondaries
  - Compton, Ionization, conversion, photon electric effect

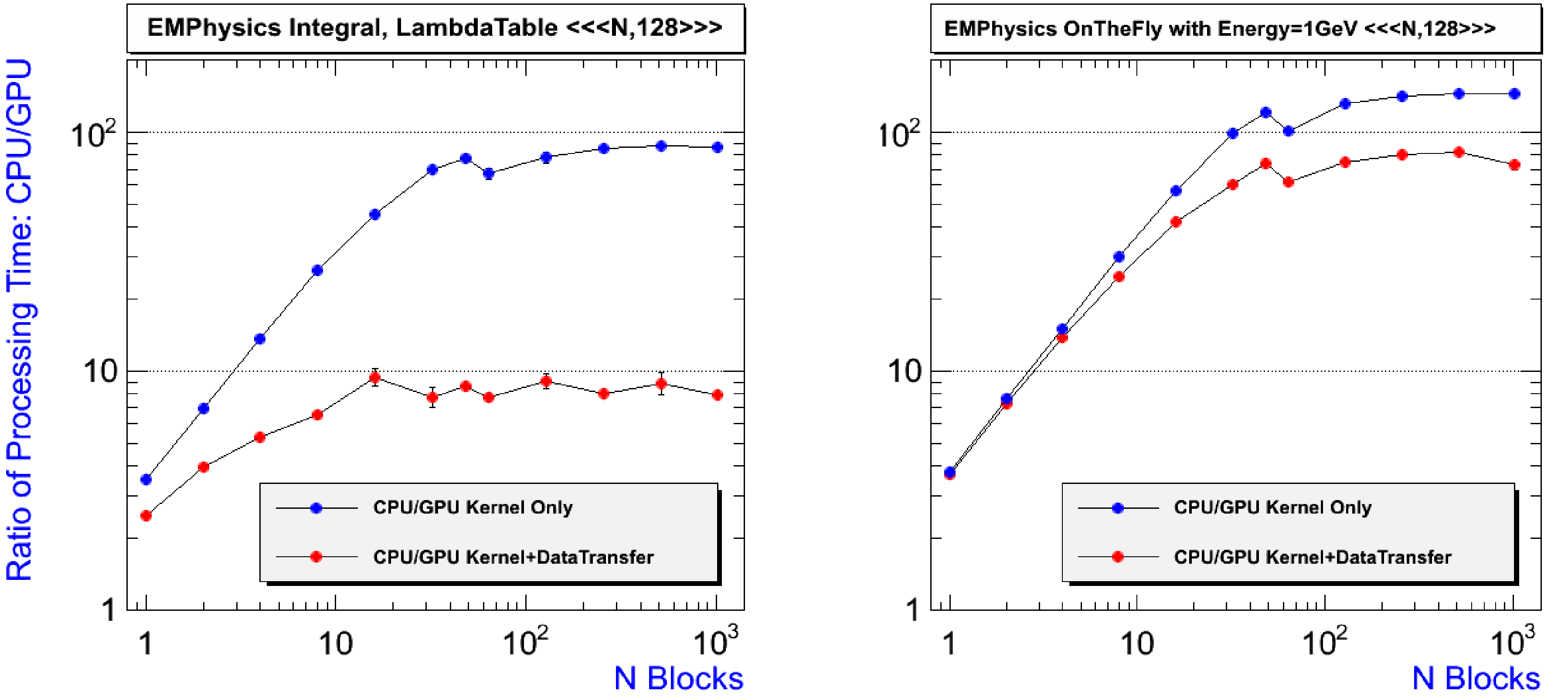
EM Physics: Bremsstalung

- See Dongwook's talk for implementation details



EM Physics: Bremsstrahlung

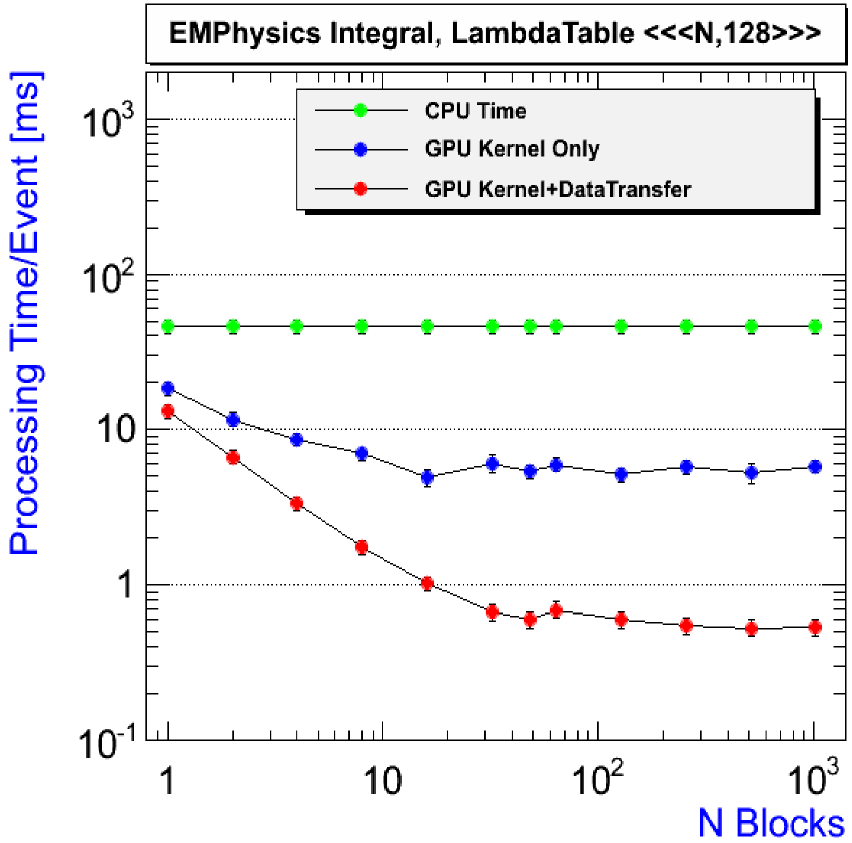
- Look-up table (left): default in Geant4
- On-the-fly calculation for electrons with E = 1GeV (right)

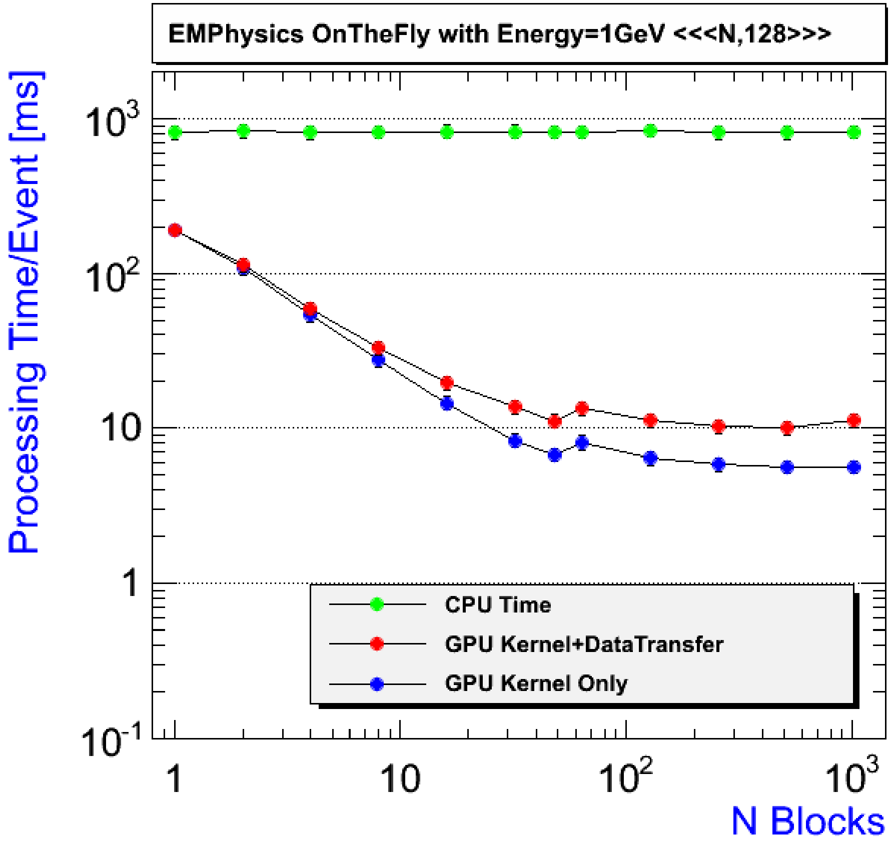


- The higher gain of the On-the-fly calculation is expected

EM Physics: Bremsstrahlung

- Processing time: Lookup table (left) vs. On-the-fly (right)

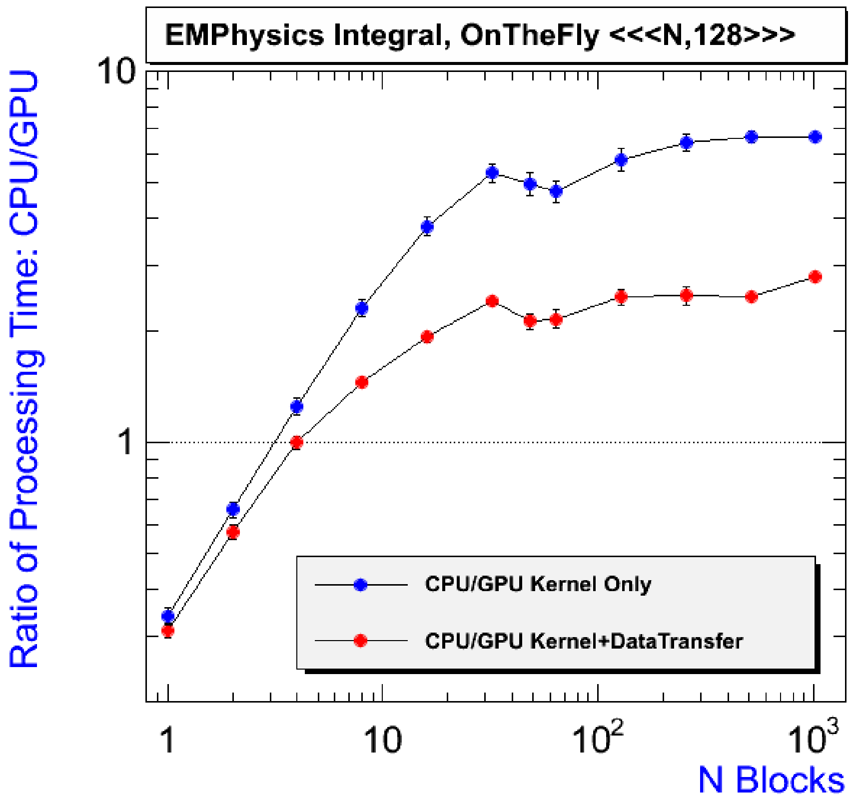


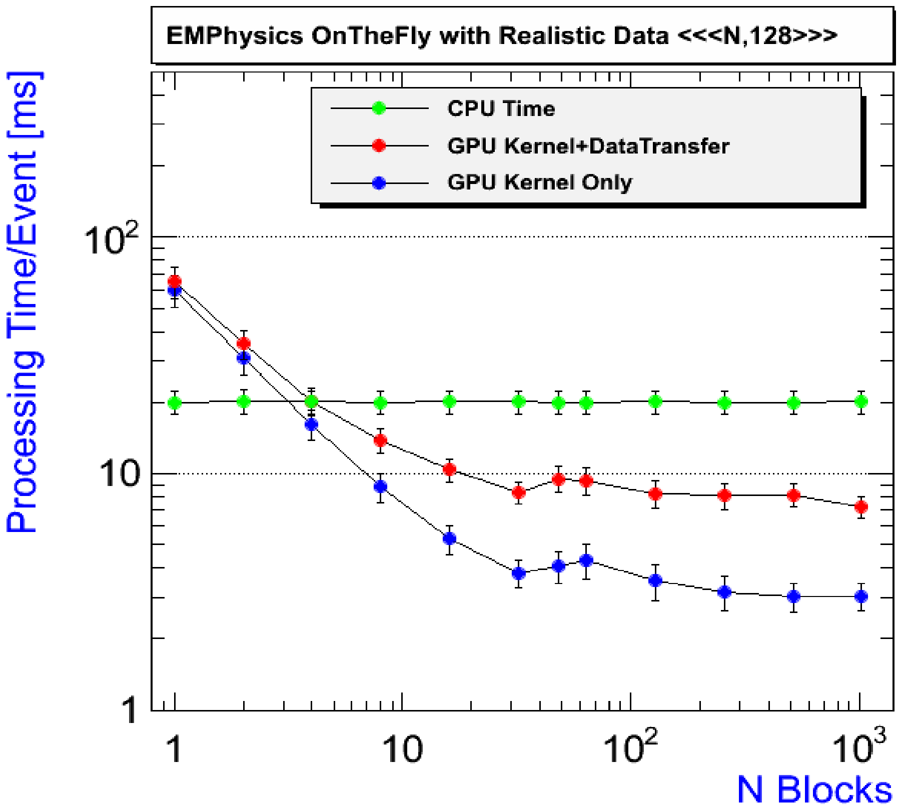


- Processing time: Lookup table << On-the-fly

EM Physics: Bremsstrahlung

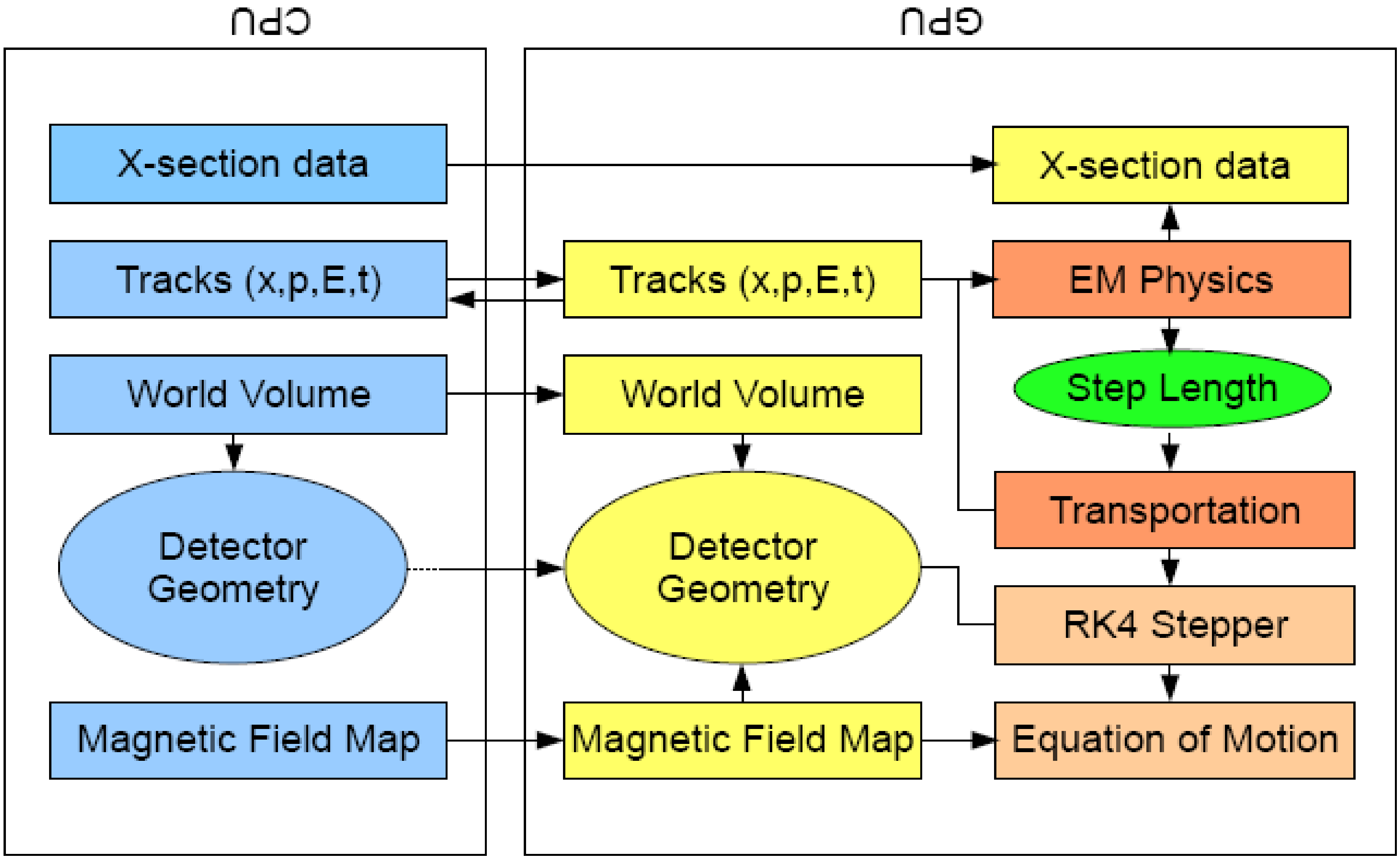
- On-the-fly calculation for realistic data (low energies)





- Low CPU/GPU gain is due to a simple calculation when E < 0.5 MeV which reduces the arithmetic intensity

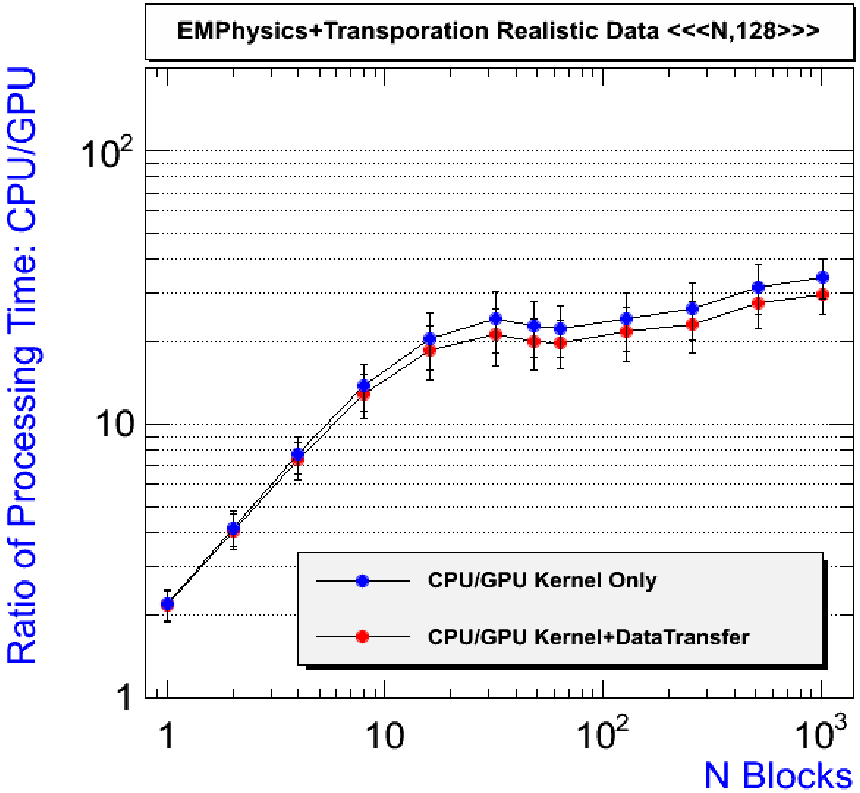
Transportation Engine

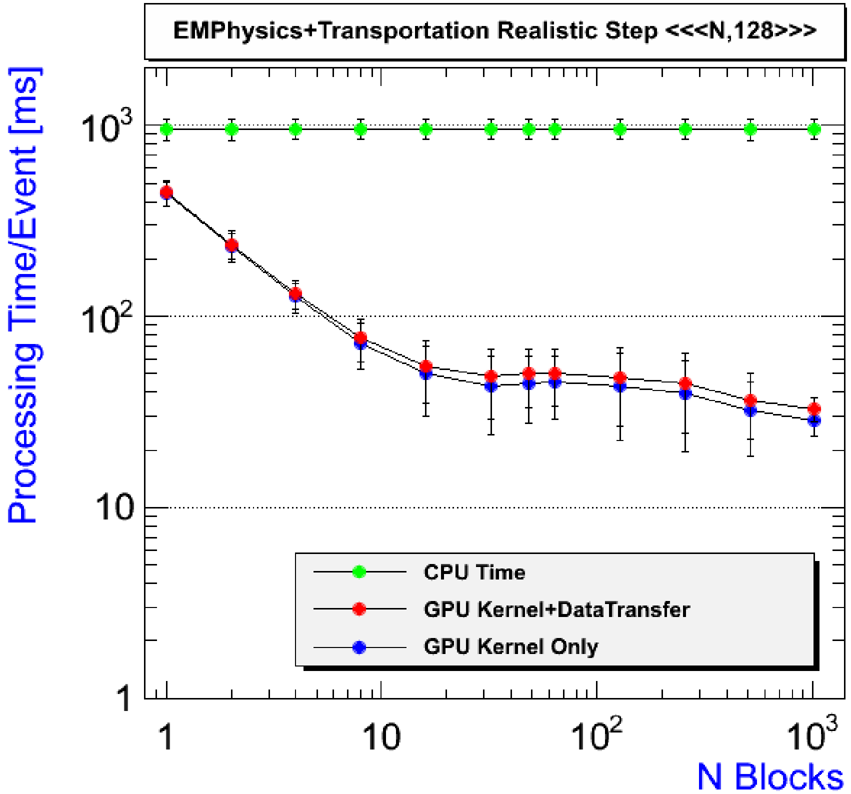


CPU **GPU**

Transportation+Geometry+EMPhysics

- For realistic steps

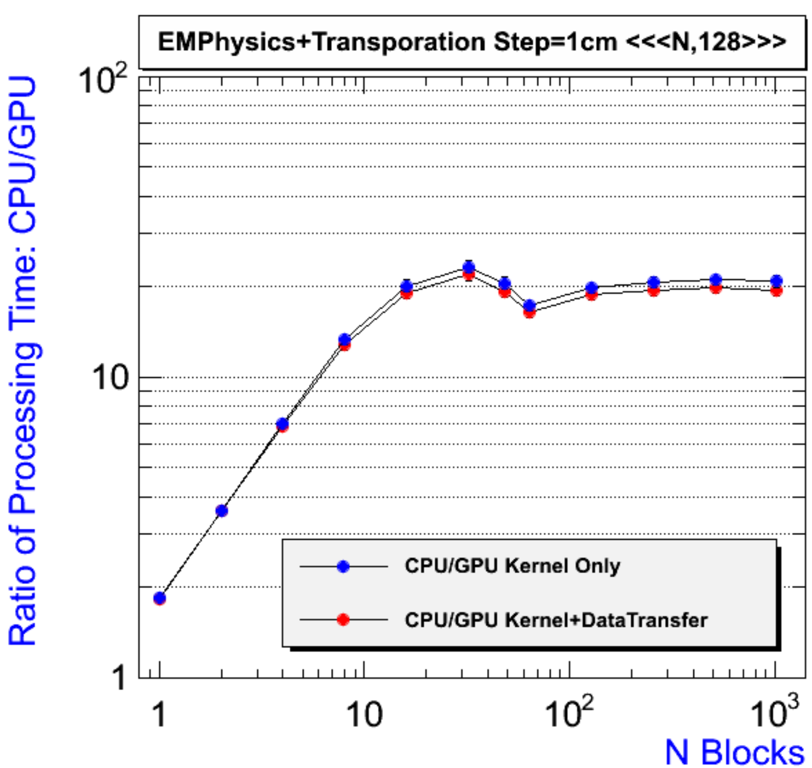


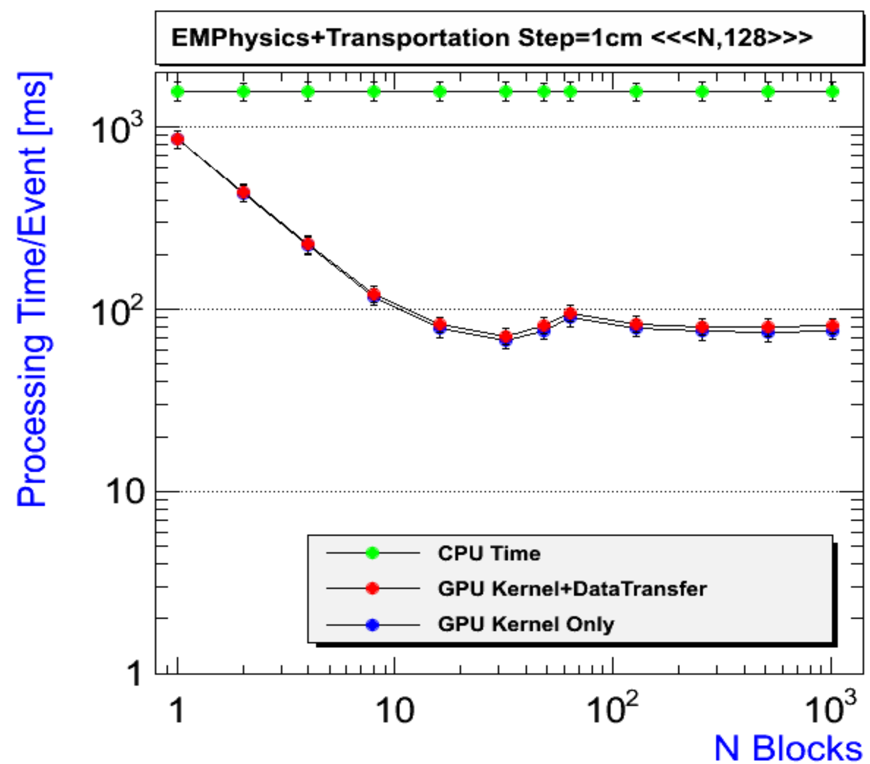


- EM Physics (Bremsstrahlung, Lookup table) improves the gain marginally. Plan to add more processes.

Transportation+Geometry+EM Physics

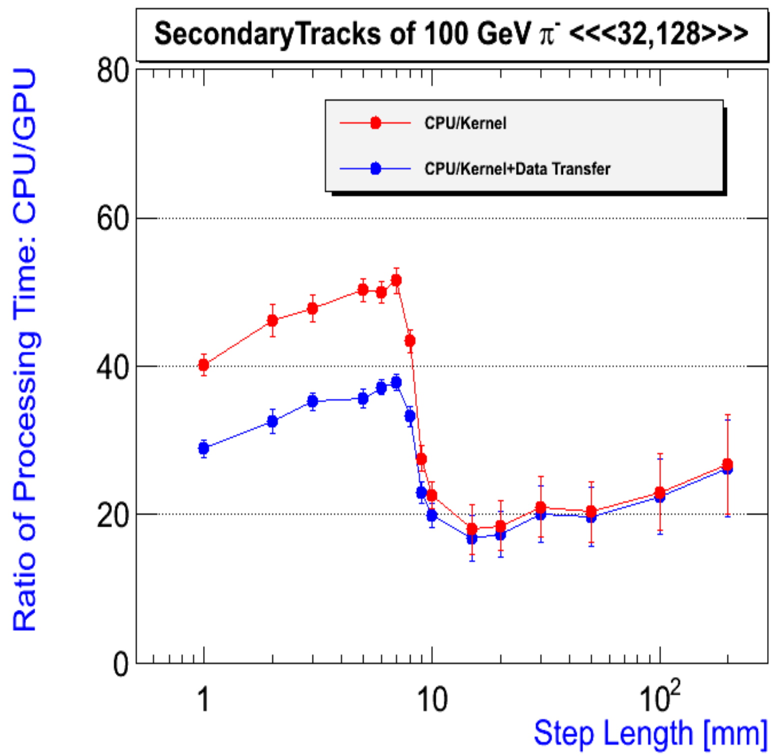
- For a fixed step lengh = 1cm

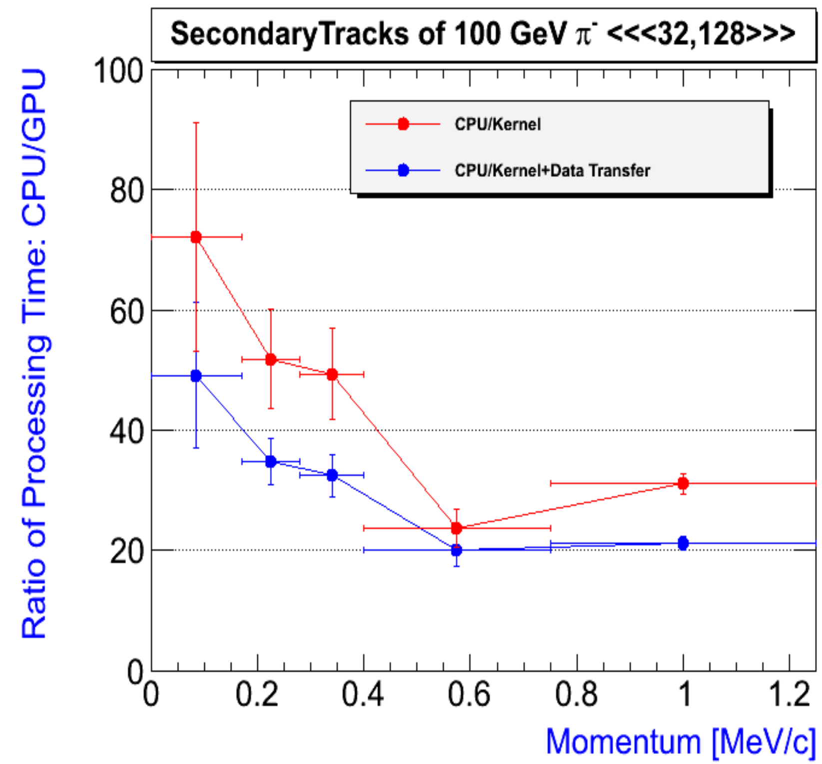




Dependence: Momentum and Step Length

- CPU/GPU for the first step of secondary tracks

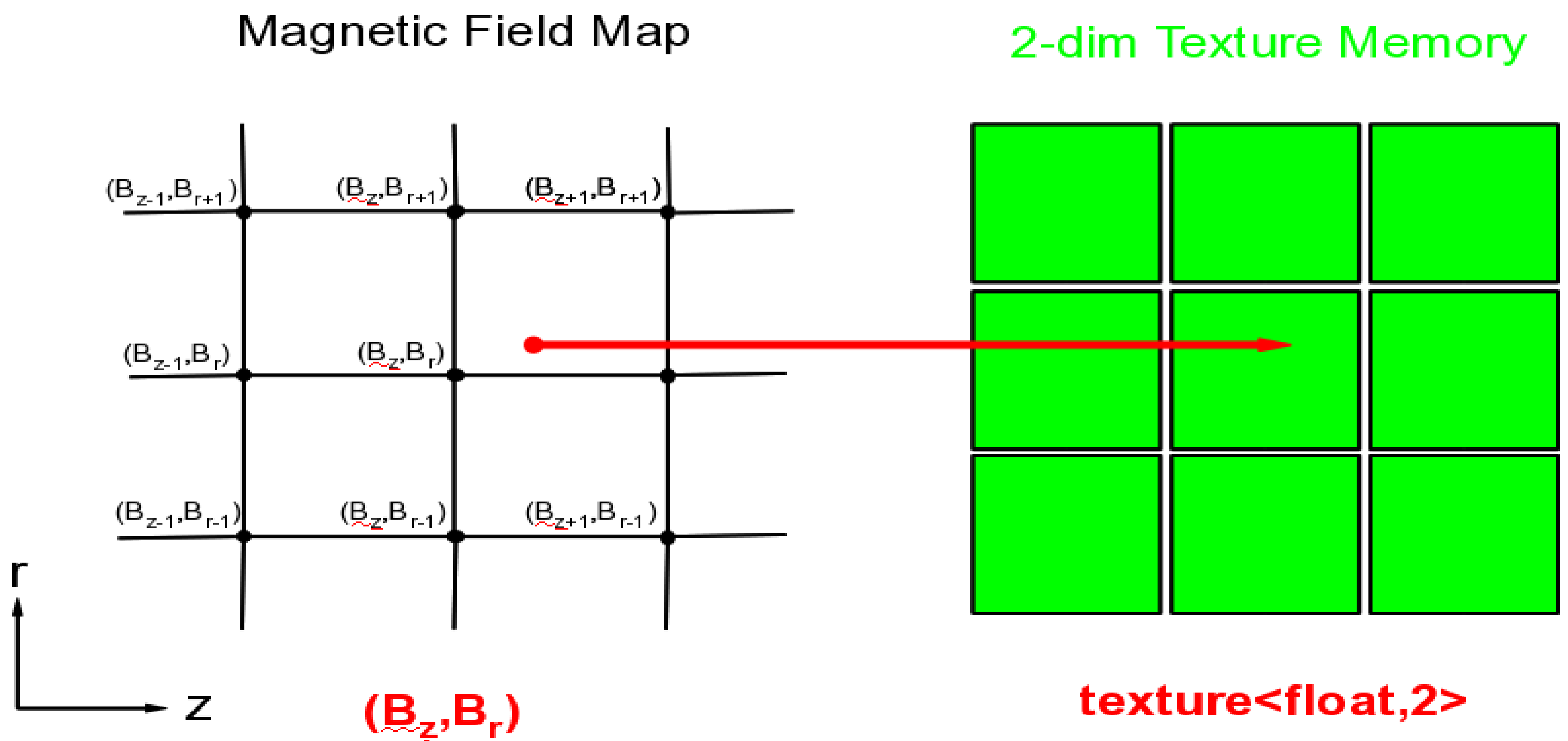




- Optimize calculation uniformity: multiple streamer
  - Keep GPU multiprocessors equally busy
  - group tracks with same number of RK4 evaluations as possible

Magnetic Field Access : Texture memory

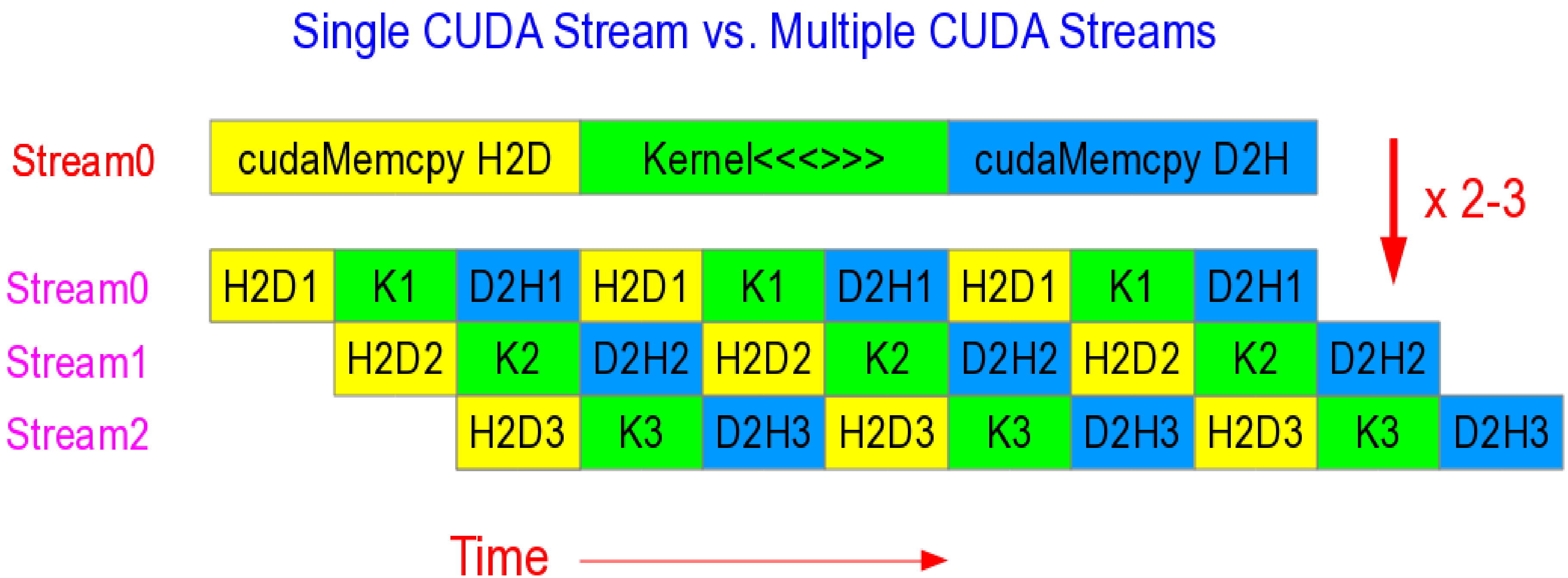
- Texture memory is cached on chip and designed for memory access with spatial locality (magnetic field map)



- Texture interpolation twice as fast as the explicit interpolation for random access

Concurrent Kernel/Stream

- Multiple CUDA streams provide the task parallelism (kernel execution and memory copies simultaneously)



- Using multiple CUDA streams for the Runga-Kutta diriver
  - no significant gain observed: balance work load evenly
  - add more calculations on device

Conclusion I

- A core part of Geant4 particle transportation has been tested on GPU
  - ratio of processing time for CPU/GPU ~ 30 with realistic data using 448 cuda cores
  - Identified key factors to maximize the GPU's ALU capacities
- Lessens learned
  - increase computational intensity on GPU
  - look for other transportation algorithms suitable for uniformity of calculations
  - organize input data for optimal efficiencies of kernel executions and data transfers

Conclusion II: Outlooks

- Geometry
  - add more solids
  - voxelized navigation
- EM physics
  - add more physics processes
  - generalize the transportation process including post step actions and pipelines for handling hits and secondaries
- Optimize GPU resources
  - more tests for multiple CUDA streams (concurrent kernel execution and copying data up/down to GPU)