Vector Physics Model

S. Y. Jun (Fermilab), J. Apostolakis, M. Novak, S. Wenzel (CERN)
for the GeantV team

20th Geant4 Collaboration Meeting at Fermilab
Sept. 28 - Oct. 2, 2015

Introduction

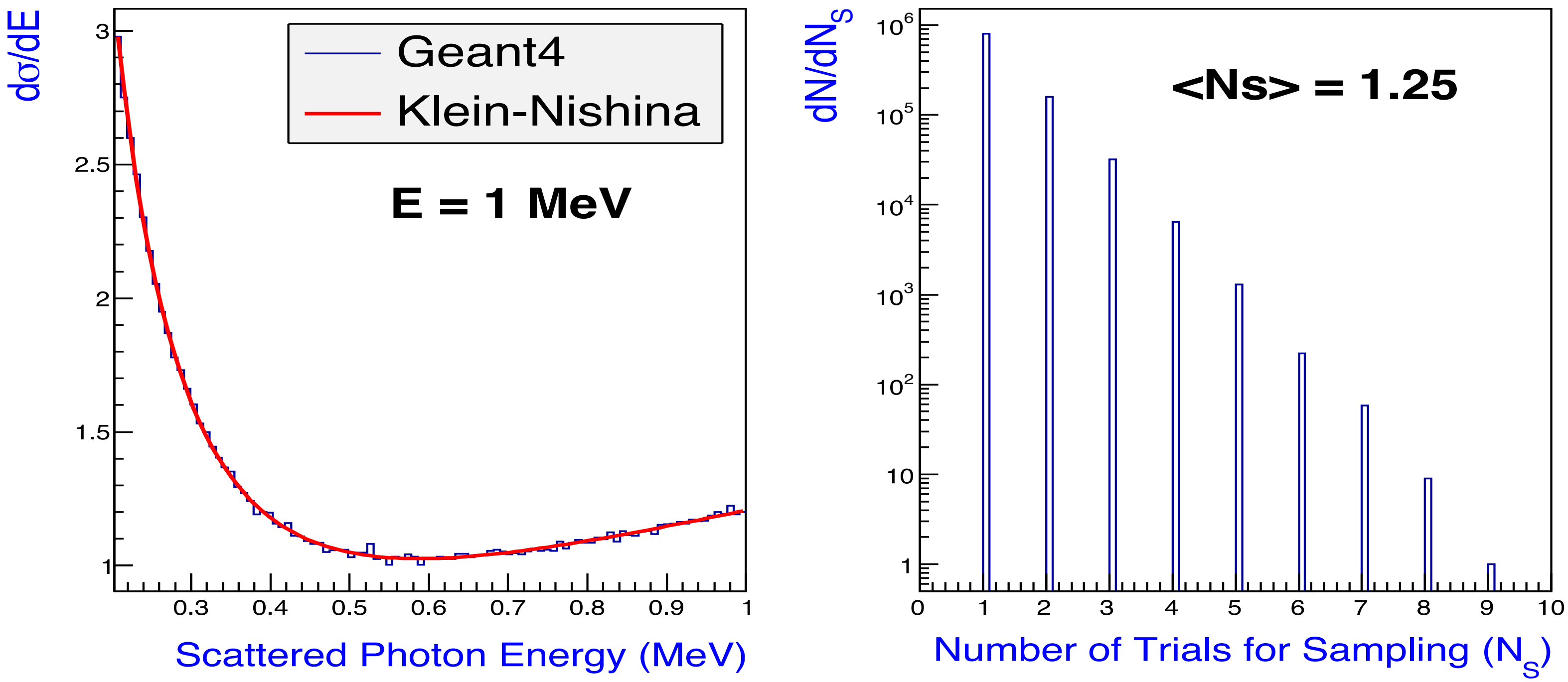
- Portable physics models for parallel architectures
  - SIMD: CPU, XeonPhi-native (vector pipeline)
  - SIMT: GPGPU (massive number of threads), (XeonPhi-offload)
- Strategies
  - Common implementation for multiple architectures
  - SIMD/SIMT friendly algorithms and data structures
  - Opportunities over challenges
- History and status
  - Based on the FNAL GPU prototype
  - Adopt the VecGeom-style backend (vectorization with Vc)
  - EM physics models for high energy photons and electrons
  - Interface to GeantV – one model (Compton) for now

Goals and Core implementations

- Intermediate targets for EM physics models
  - Final status analysis (sampling secondary particles)
  - Mean free path analysis (the total cross section)
- Explore new algorithms or techniques
  - Alias sampling
  - Efficient binning
  - Tasks decomposition
  - SIMD pRNG
- Minimize overheads
  - Gather (sequential operation): scalar + vector = scalar
  - Conversion (int to float or vice versa)
  - Memory bounded operation (avoid contention, hide latency)
  - Data transfer to devices

Why alias?

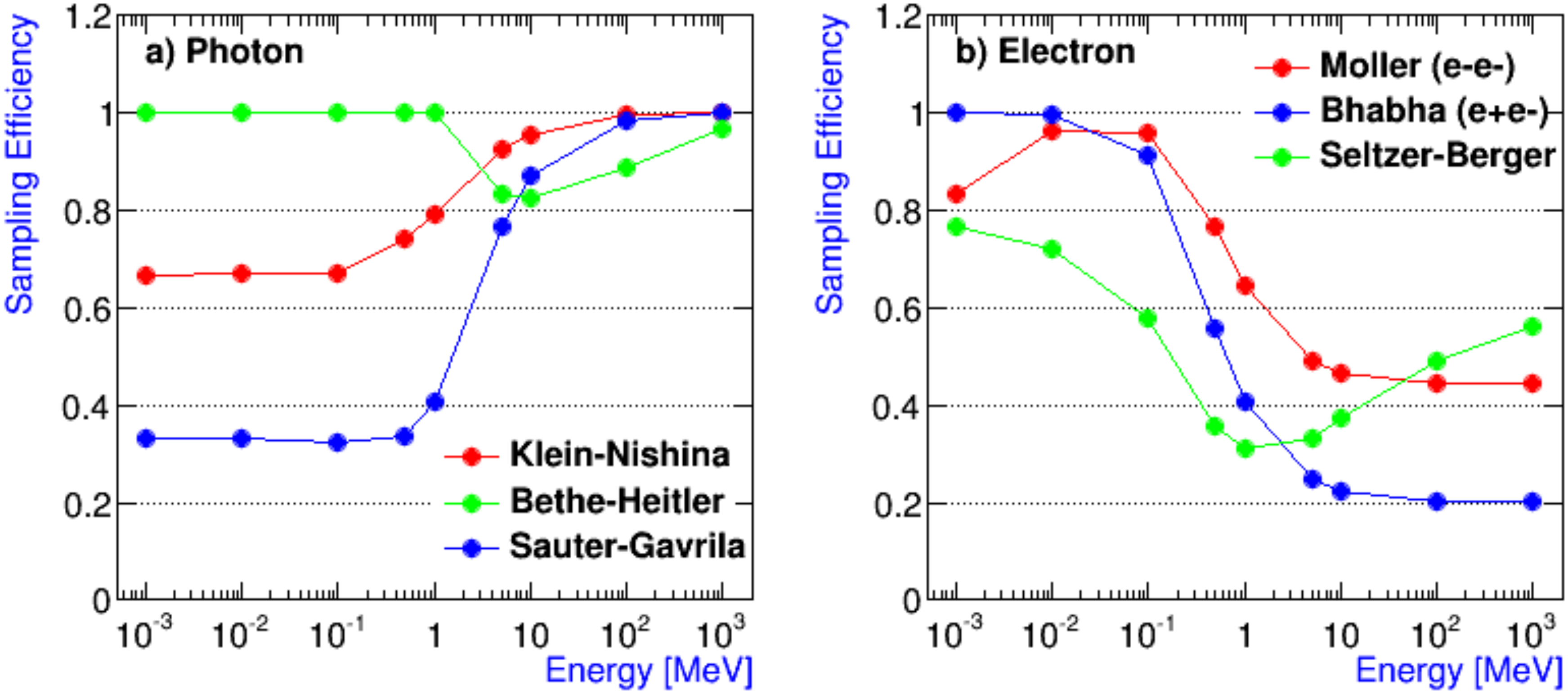
- Conventional composition and rejection techniques
  - Ex: Klein-Nishina $d\sigma/dE[\gamma(E, 0) \rightarrow \gamma' (E', \sin\theta)]$
  - Sampling efficiency $\varepsilon \sim 0.80$ at E = 1 MeV (Ns=1.25)



  - Conditional breaks (do-while)
  - Branches for vector operations or thread divergences

Review: Geant4 Composition and Rejection Methods

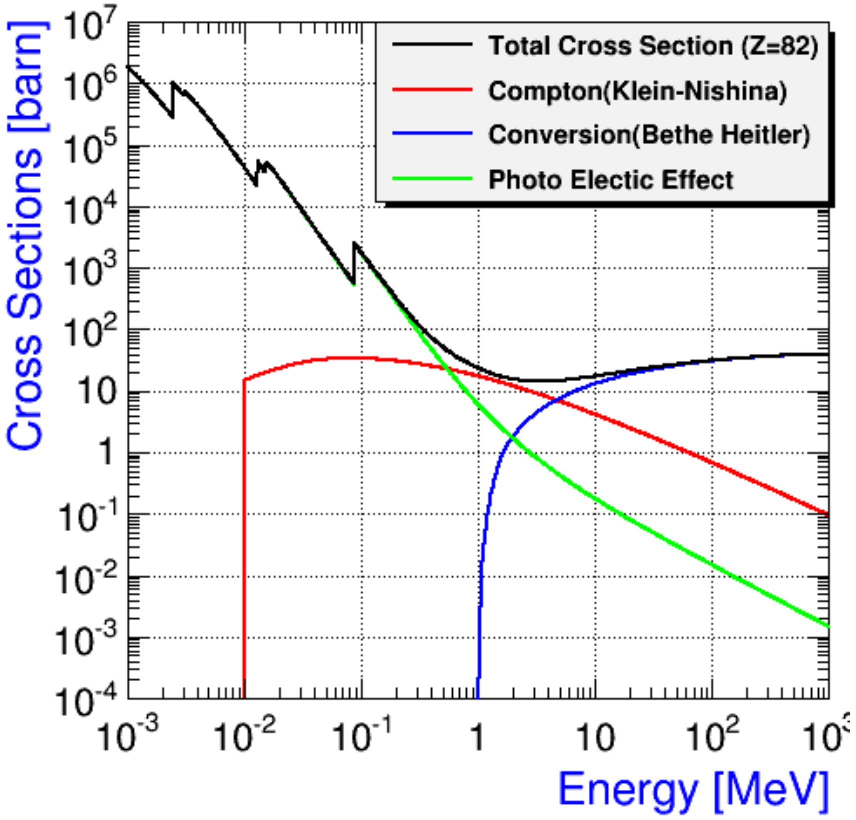
- Average sampling efficiencies (ε) of composition and rejection methods used in high energy EM physics models

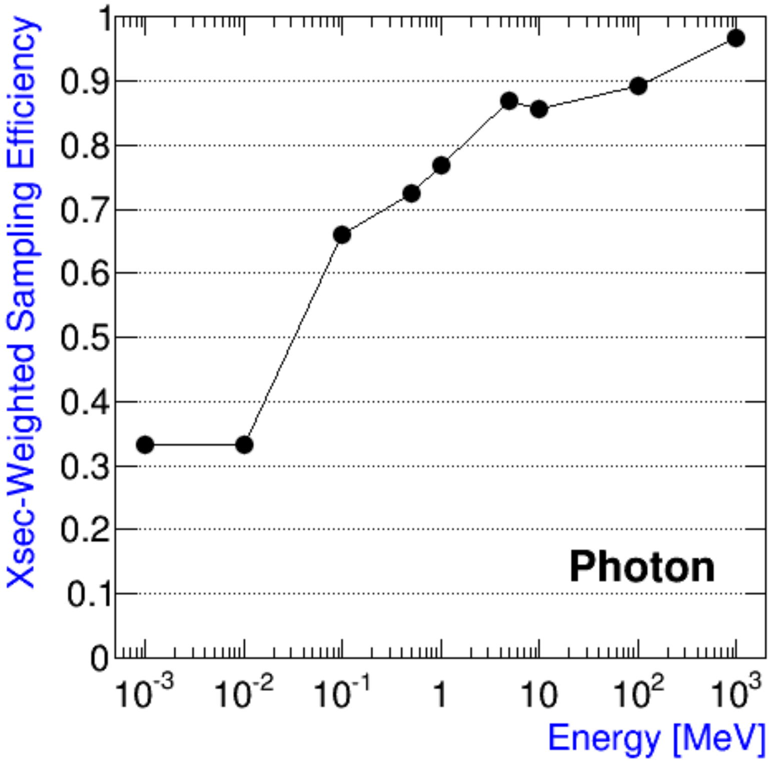


  - ε(γ): relatively low at E < 1MeV and ε(e^-): low at E > 1MeV

Review (continue)

- Impact to the overall simulation: cross section weighted sampling efficiency = $\Sigma_i \varepsilon_i \sigma_i/\sigma_{total}$ (example for photon models)





  - the probability to have the same number of trials for all n-vector operands (or n-threads) < ε^n (branch or thread divergence)

- Alternative: Alias sampling method (effectively vectorizable)

Alias Method for N Discrete Outcomes

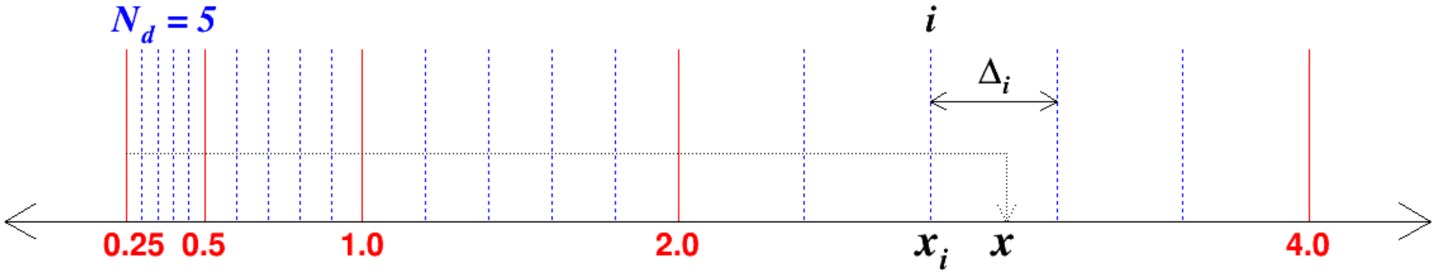
- Recast p.d.f f(x) → c, N equal probable events, each with likelihood 1/N = c (A. J. Walker, 1974)
- For EM physics models, construct Tables[Z] of 2-dimensional arrays [input-energy][sampling-variable]
  - Probability distribution of the sampling variable (p.d.f)
  - Alias, a[recipient] = donor
  - Non-alias probability
- Binning
  - Input energy -> Power2Divisor (cf. linear, log) (see next page)
  - Sample variables
    - Energy (linear or log)
    - Angle (linear)
  - Uniform sampling within a bin or the linear interpolation using p.d.f: $(f_{low} + f_{high})/2 <$ random, take x_{low} otherwise x_{high}

Binning: Power2Divisor for x(Nmin,Nmax,Nd)

- Construct bins (x_i) with a equal divisor (N_d) between the power of 2 from 2^{Nmin} to 2^{Nmax} for any integer N_{min} and N_{max}

$$x_i = [1. + (\frac{i\%N_d}{N_d})] \cdot 2^{\frac{i}{N_d}+\mathrm{N_{min}}}, \qquad i \in [0, (N_{max} - N_{min}) \cdot N_d + 1]$$

- For any x in $[2^{Nmin}, 2^{Nmax}]$, the bin number (i) and the fraction (δ)



$$\begin{aligned} M &= \mathrm{frexp}(x, \&E) \\ i &= N_d \cdot (E - 1 - N_{min}) + \mathrm{floor}[(2M - 1) \times N_d] \\ \Delta_i &= [\mathrm{ldexp}(1., E - 1)]/N_d \\ \delta &= \frac{(x - x_i)}{\Delta_i}, \qquad \delta \in [0, 1) \end{aligned}$$

Random number generators

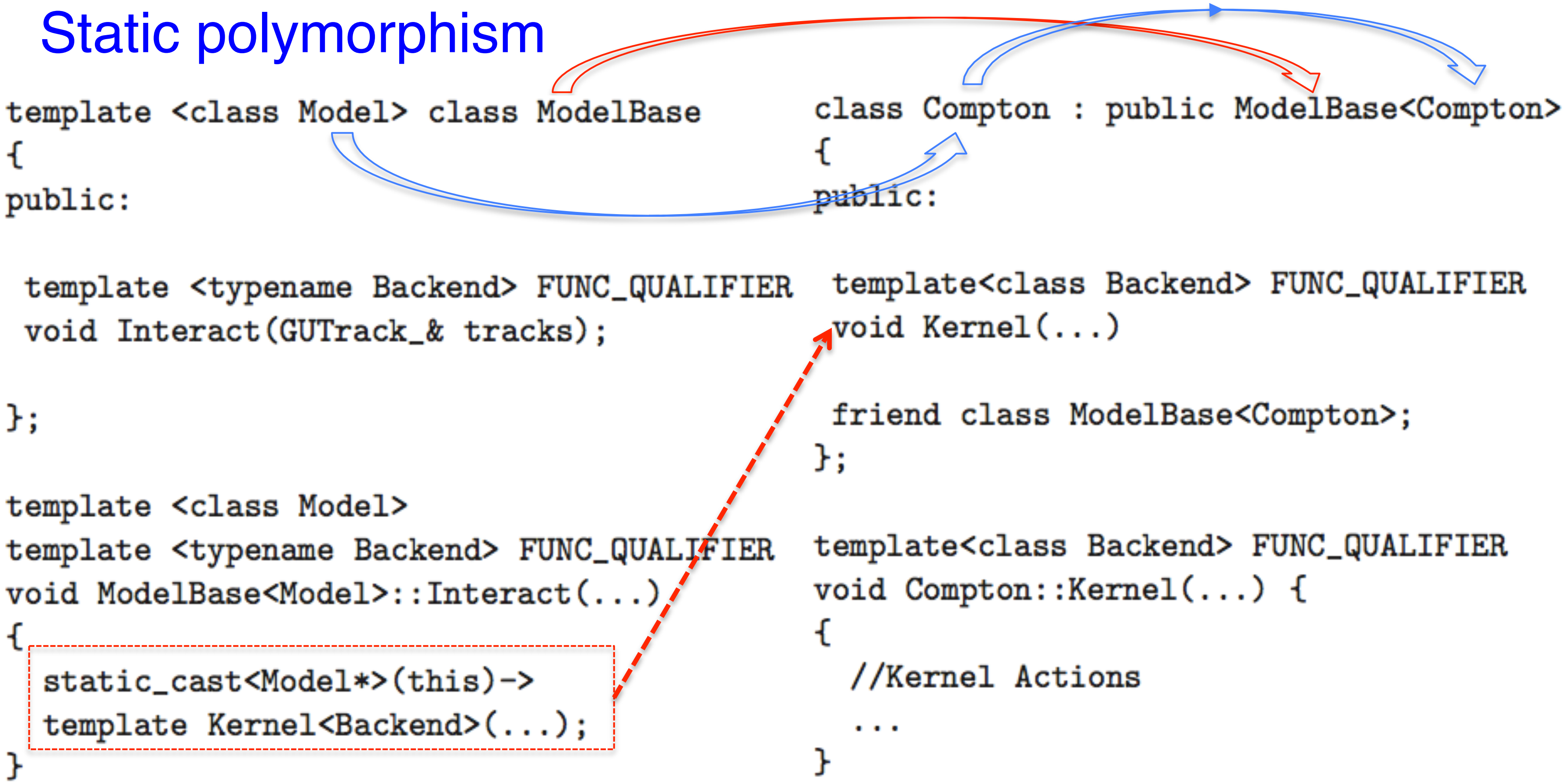
- Pseudo random number generator (pRNG)
  - Sequential: std::rand()
  - Vector: Vc Random_t
  - Cuda pRNG library (CURAND)
    - Xor-family (XORWOW)
    - L'Ecuyer's multiple recursive generator (MRG32k3a)
    - Mersenne Twister (MTG32, 32bit, period 2^11213)
- Performance: 10K pRNG generation time in [ms]
  - CPU (Xeon x5650, 2.67GHz)
  - GPU (Nvidia K20M)

pRND generator	10K pRNG [ms]
Rand()	3.66
Vc random_t	1.44

CURAND	Init States for 64 blocks [ms]	10K pRNG for 256 threads [ms]
XORWOW	4.12	7.92
MRG32k3a	5.02	21.88
MTG32	0.69	31.94

Static Polymorphism: Implementation Detail

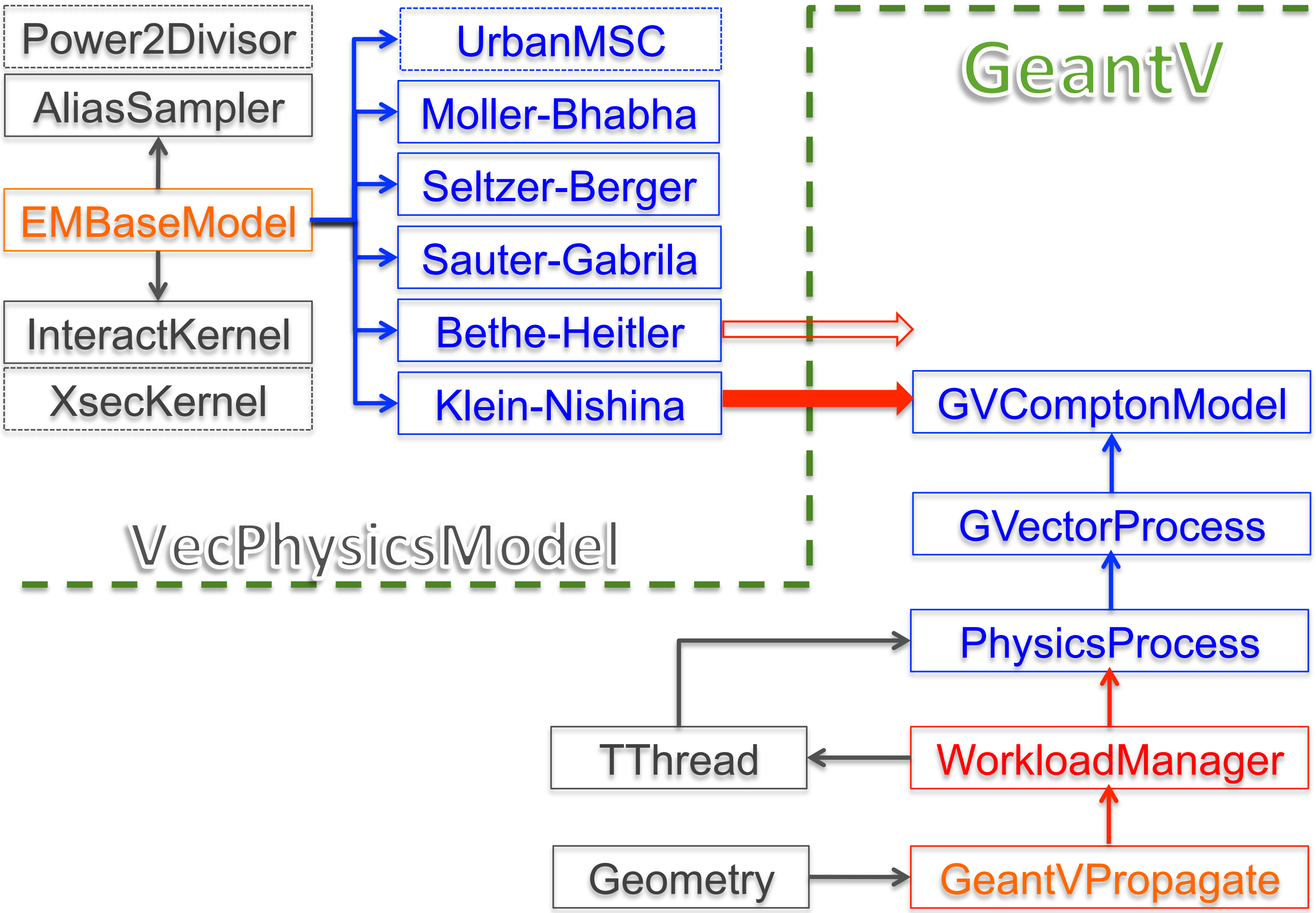
- EM models have many common interfaces. However,
  - Virtual is not allowed for Template<backend>
  - Avoid unnecessary virtual layers (performance consideration)
- Static polymorphism



```
template <class Model> class ModelBase
{
public:

 template <typename Backend> FUNC_QUALIFIER
 void Interact(GUTrack_& tracks);

};

template <class Model>
template <typename Backend> FUNC_QUALIFIER
void ModelBase<Model>::Interact(...)
{
  static_cast<Model*>(this)->
  template Kernel<Backend>(...);
}
```

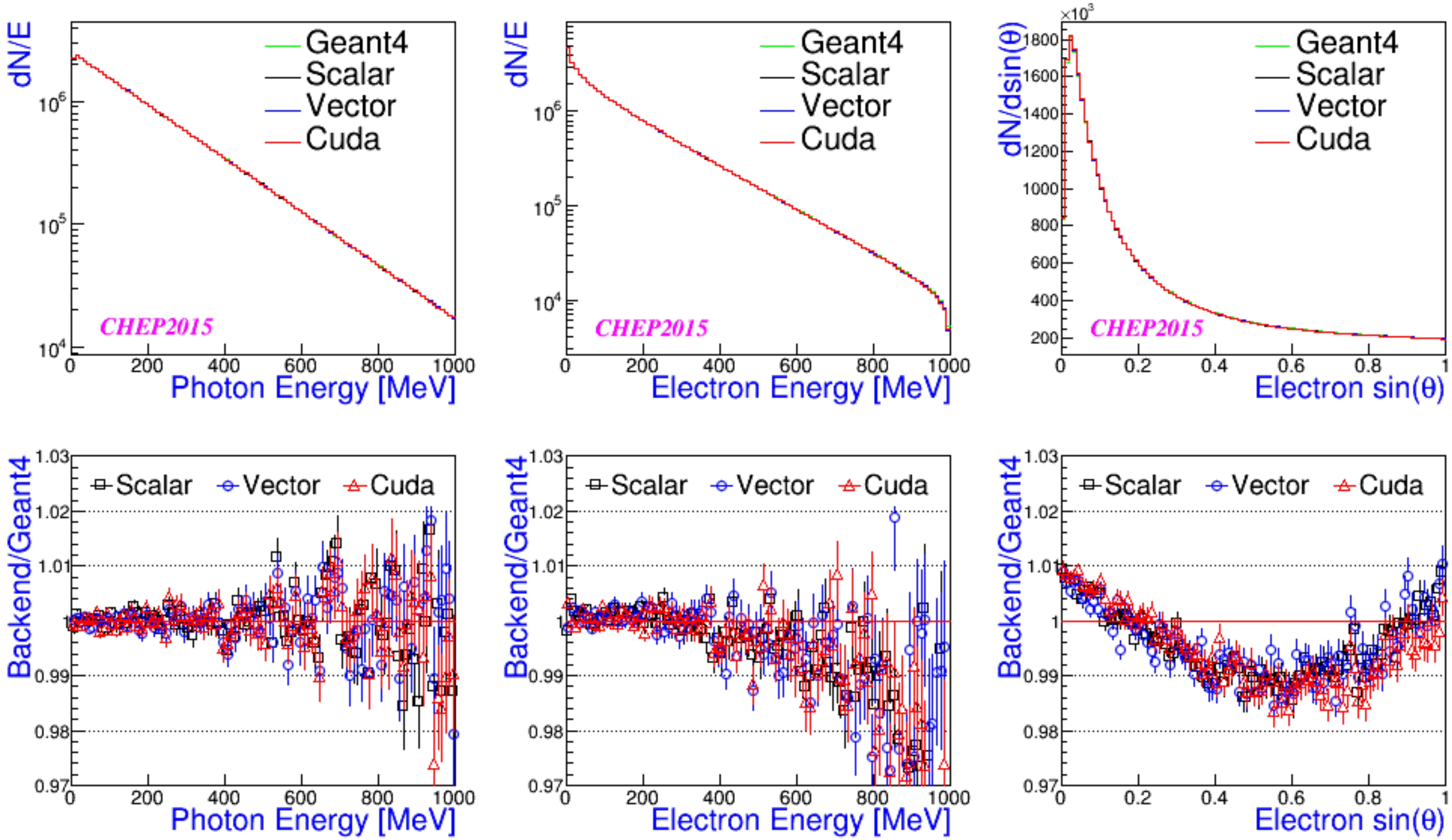
```
class Compton : public ModelBase<Compton>
{
public:

 template<class Backend> FUNC_QUALIFIER
 void Kernel(...)

 friend class ModelBase<Compton>;
};

template<class Backend> FUNC_QUALIFIER
void Compton::Kernel(...) {
{
  //Kernel Actions
  ...
}
```

Structure of Vector Physics and Connectivity to GeantV



Preliminary Validation (CHEP2015): Alias Sampling Method

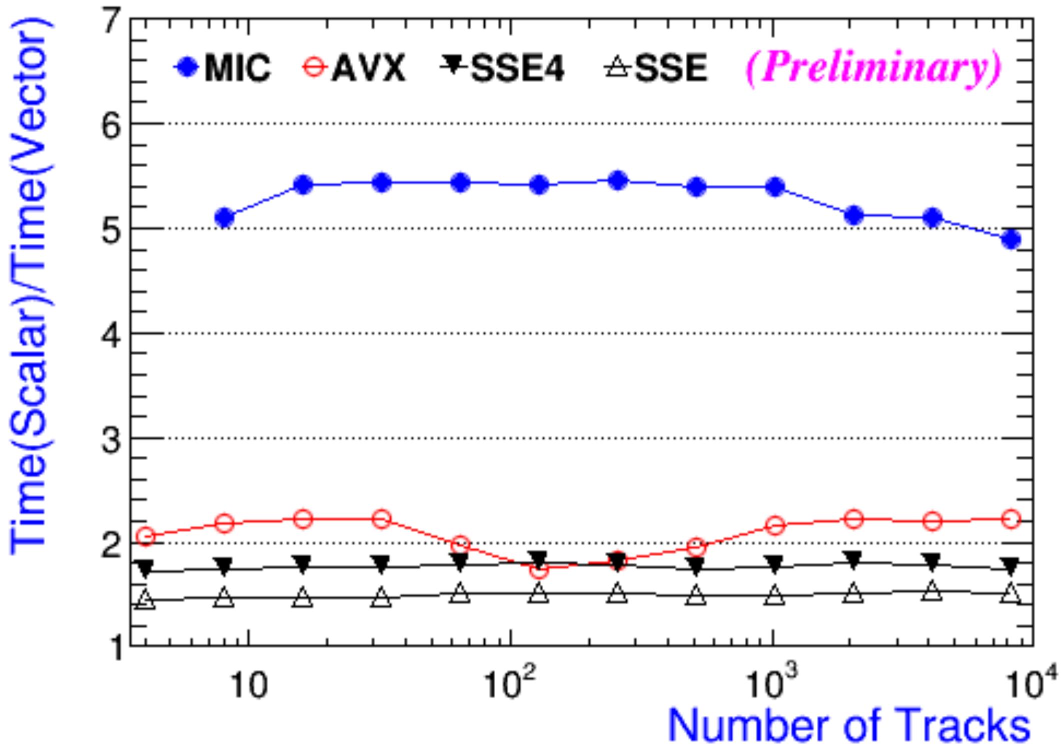
- Compton model with input photon energy [1,1000] MeV



- Sign of bias, but within ±(1-2)%

Performance: Scalar vs. Vector

- Vector speedup = CPU(scalar)/CPU(vector) for the same task
  - Sampling the secondary electron of the Klein-Nishina model
  - MIC (Xeon Phi 5110P), AVX (Xeon E5-2680), SSE(4)(Xeon x5650)



- Gain: from ~1.5x (SSE) to ~5 (MIC) - a sign of hope 🙂

Preliminary Performance (2): Vector Physics Model

- The average time (in second) for simulating 4992 interactions with input particles energy at 500 MeV (100 measurements)
  - CPU: Intel(R) Xeon(R) CPU E5-2620 0 @ 2.00GHz - SSE
  - GPU: Nvidia K20 Kepler, 2496 cores @ 0.7 GHz - <<<26,192>>>
  - Performance strongly depends on the input energy

Models	**Geant4**	**Scalar**	**Vector**	**CUDA**
Klein-Nishina	0.152	0.178	0.092	0.049
Bethe-Heitler	0.326	0.318	0.202	0.062
Sauter-Gavrila	0.067	0.169	0.094	0.047
Moller-Bhabha	0.167	0.177	0.091	0.048
Seltzer-Berger	0.481	0.288	0.146	0.055

- Scalar/CUDA = ~4-5 (the FNAL GPU prototype CPU/GPU=~40-100)
  - Binning: log vs. linear in input energy
  - Linking to an external library vs. whole-program compilation (nvcc)
  - Do not understand the performance degradation yet

Summary and Plan

- Reviewed the current status of vector EM physics models
  - Alias sampling
  - Static polymorphism
  - Efficient binning techniques
- Preliminary validation and performance evaluations
  - Physics validations w.r.t Geant4
  - Performance: scalar/vector/cuda
- Plan for ACAT (Jan, 2016)
  - Provide fully validated vector Compton (KleinNishina) and e-Ionization (Moller) models
  - Test models within the GeantV framework
  - Evaluate performance