

Vectorized EM physics status

Vitalii Drohan
(15/05/2018)

EM physics overview

- 1) Post step action handler - 10%
but only 6% is spent in model sampling(4% if we exclude photoelectric)
- 2) ComputeIntL handler - 16.7%
It is hard to vectorize since there is not a lot of computation here, but it can be improved.
- 3) PostPropagationHandler(MSC part 1) - 23%
Approx 16.5% of it can be vectorized
- 4) PrePropagationHandler(MSC part 2) - 13%
~ half of it can be vectorized(6%)
In total we have around 27% of vectorizable code. (It is TestEM3 with B=0)

* x% means - percent of total time of execution in TestEm3 with B=0

Expectations

- 1) Vectorization can speedup math up to factor of 2(old AVX) - 3 (Skylake AVX)
But with AVX512 it can be as much as 5-6(result of measurement)

So total time reduction can be around 13% -- 30%

If we calculate all physics in floats it can be(very optimistically) up to 40%

//Here I talk about speedup compared to current GeantV

Vectorized MSC(only part 1)

PostPropagationHandler::Dolt == scalar - 24.28% | vector - 15.20%

vectorized method - SampleMSC == scalar - 18.94% | vector 11.60%

Angle sampling == scalar - 9.49 % | vector - 3.91%

Total time of TestEM3 goes down by ~8-10% when Vector MSC turned on.

Overview of work

- 1) Now we only scratched a surface of what can be done.
And already we can see the difference in overall time of execution only because of vectorized physics.
- 2) Even though result is only about 8-10% I think it can be twice of this on AVX512 Skylake and future architectures. Or even more than that if physics will be calculated in floats.

Perspectives

- 1) Add AVX512 support, currently there is no way to test AVX512.
In Skylake Platinum I tested simple math with AVX(on that machine avx is alias to lower part of AVX512 registers) and saw speedup of exp calculation around 5(compared to speedup of 2 on Sandy Bridge).
- 2) Optimize current vectorized implementations. It was done in short amount of time and they can be improved.
- 3) Improve ComuteIntL() and other parts of framework -- it will increase importance of physics in overall time
- 4) Introduce floats instead of doubles for physics. No difference in physics results, twice improvement in throughput for vectorized code. But it is not that trivial to do since something(directions for ex.) should be done in doubles.