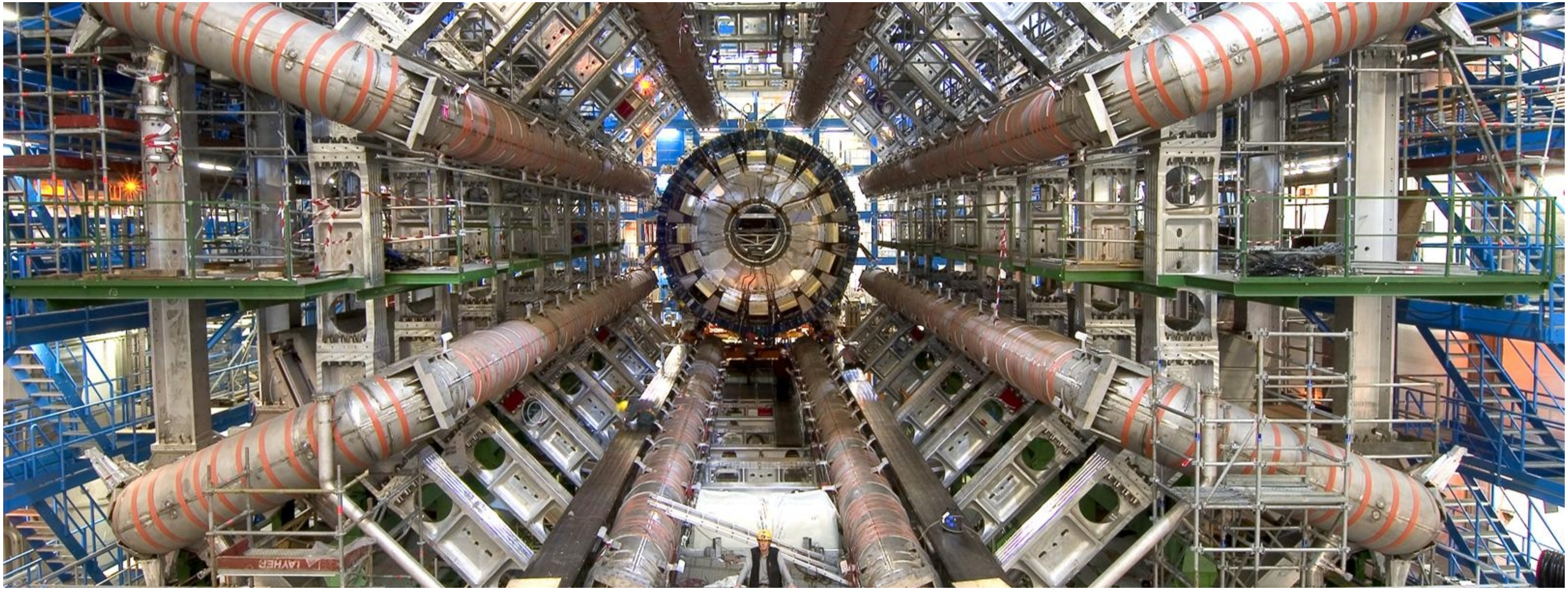


MACHINE LEARNING TO EMPOWER PHYSICS MODELING

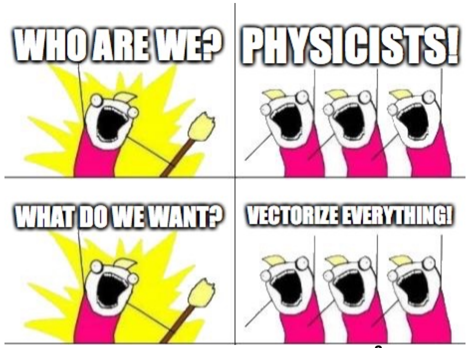
Marilena Bandieramonte for the Simulation Group (SFT)
marilena.bandieramonte@cern.ch

CERN

A SUCCESSFUL R&D

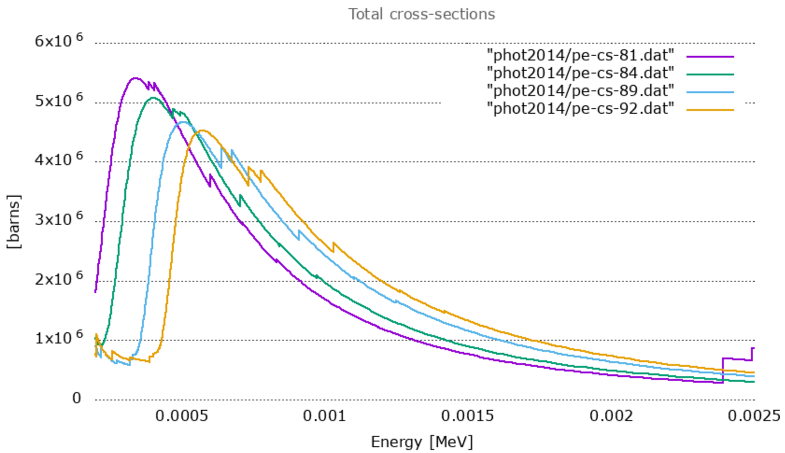
- GeantV R&D aims at showing that vectorization is the way to go to improve particle transport simulation code:
  - Engine to transport particles in parallel (v3 - Generic Vector Flow approach)
  - VecGeom
  - VecCore

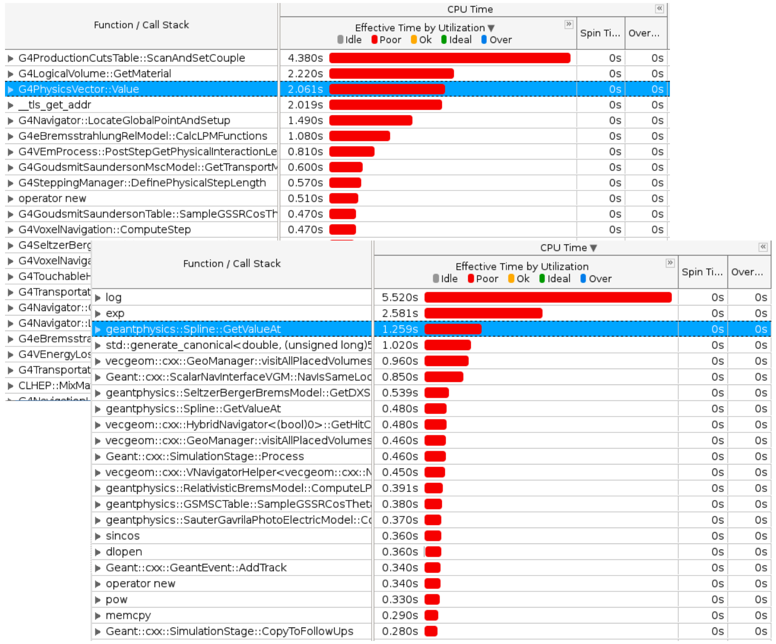
- What about physics?
  - EM physics models have been theoretically revisited/improved and implemented in a ”vector-friendly” way
    - Massive use of tables created at initialization phase to store cross-sections (log-log interpolation)
    - Alias table sampling coexisting with rejection sampling
  - Performance tests on vectorized final state sampling algorithms show promising results
  - Will this be ENOUGH?
    - Cross-section tables are not always usable
    - The cost to pay to have data ready in SIMD vectors is not negligible



EXAMPLE: PE EFFECT

- Photoelectric effect total cross-section is not an easy function
  - Fit in two different energy ranges, but not below k-shell binding energy
    - Tabulated cross-sections left for low energies
  - For the final state sampling one need to sample
    - the subshell: This is going through a binary search algorithm (not vectorizable) + linear or spline interpolation
    - the angle: described by the SauterGavrila differential cross-section correct only for k-shell



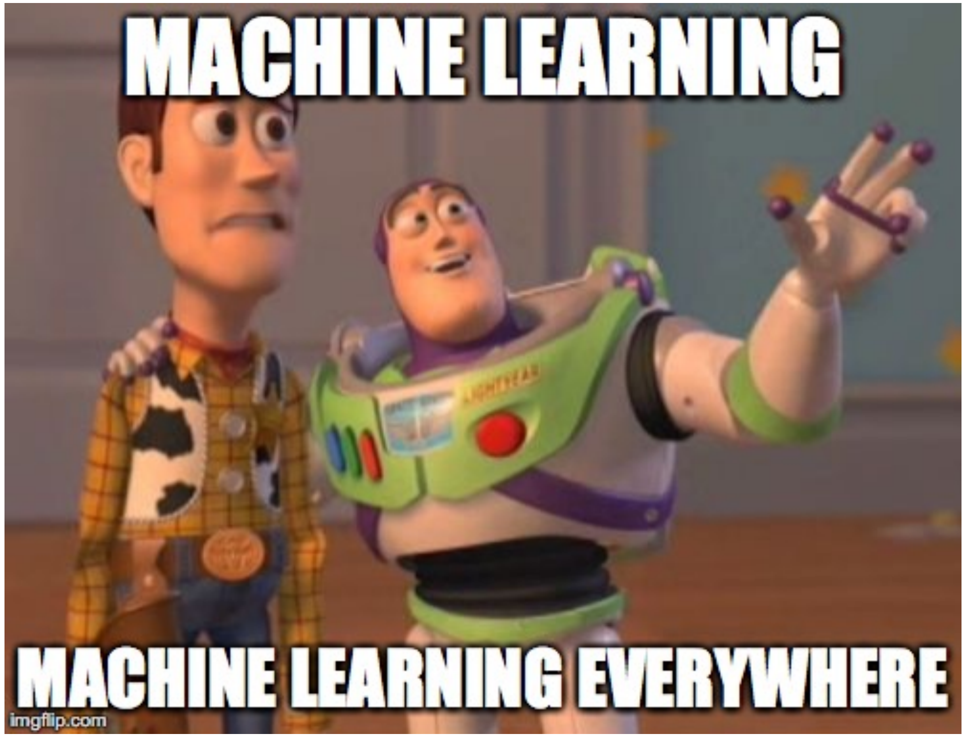


Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Ti...	Over...
G4ProductionCutsTable::ScanAndSetCouple	4.380s	0s	0s
G4LogicalVolume::GetMaterial	2.220s	0s	0s
G4PhysicsVector::Value	2.061s	0s	0s
__tls_get_addr	2.019s	0s	0s
G4Navigator::LocateGlobalPointAndSetup	1.490s	0s	0s
G4eBremsstrahlungRelModel::CalcLPMFunctions	1.080s	0s	0s
G4VEmProcess::PostStepGetPhysicalInteractionLe	0.810s	0s	0s
G4GoudsmitSaundersonMscModel::GetTransportM	0.600s	0s	0s
G4SteppingManager::DefinePhysicalStepLength	0.570s	0s	0s
operator new	0.510s	0s	0s
G4GoudsmitSaundersonTable::SampleGSSRCosTh	0.470s	0s	0s
G4VoxelNavigation::ComputeStep	0.470s	0s	0s
G4SeltzerBerg			
G4VoxelNaviga			
G4TouchableH			
G4Transportat			
G4Navigator::(			
G4Navigator::L			
G4eBremsstra			
G4VEnergyLos			
G4Transportat			
CLHEP::MixMa			

Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Ti...	Over...
log	5.520s	0s	0s
exp	2.581s	0s	0s
geantphysics::Spline::GetValueAt	1.259s	0s	0s
std::generate_canonical<double, (unsigned long)5	1.020s	0s	0s
vecgeom::cxx::GeoManager::visitAllPlacedVolumes	0.960s	0s	0s
Geant::cxx::ScalarNavInterfaceVGM::NavIsSameLoc	0.850s	0s	0s
geantphysics::SeltzerBergerBremsModel::GetDXS	0.539s	0s	0s
geantphysics::Spline::GetValueAt	0.480s	0s	0s
vecgeom::cxx::HybridNavigator<(bool)0>::GetHitC	0.480s	0s	0s
vecgeom::cxx::GeoManager::visitAllPlacedVolumes	0.460s	0s	0s
Geant::cxx::SimulationStage::Process	0.460s	0s	0s
vecgeom::cxx::VNavigatorHelper<vecgeom::cxx::N	0.450s	0s	0s
geantphysics::RelativisticBremsModel::ComputeLP	0.391s	0s	0s
geantphysics::GSMSCTable::SampleGSSRCosThet	0.380s	0s	0s
geantphysics::SauterGavrilaPhotoElectricModel::Cc	0.370s	0s	0s
sincos	0.360s	0s	0s
dlopen	0.360s	0s	0s
Geant::cxx::GeantEvent::AddTrack	0.340s	0s	0s
operator new	0.340s	0s	0s
pow	0.330s	0s	0s
memcpy	0.290s	0s	0s
Geant::cxx::SimulationStage::CopyToFollowUps	0.280s	0s	0s

Profiling of a Geant4/GeantV application, revealing their major hotspots

THE IDEA:


MACHINE LEARNING
MACHINE LEARNING EVERYWHERE
imgflip.com

YES ML, BUT HOW?

- Machine learning applied to FASTSIM looks very promising

- What if we go one level beyond and we replace computationally expensive physics models with ML blocks
  - Able to learn complex cross-sections shapes (total, differential)?
  - Able to directly generate the final-state?

→ From "physics-agnostic" to **"physics-aware"** neural networks

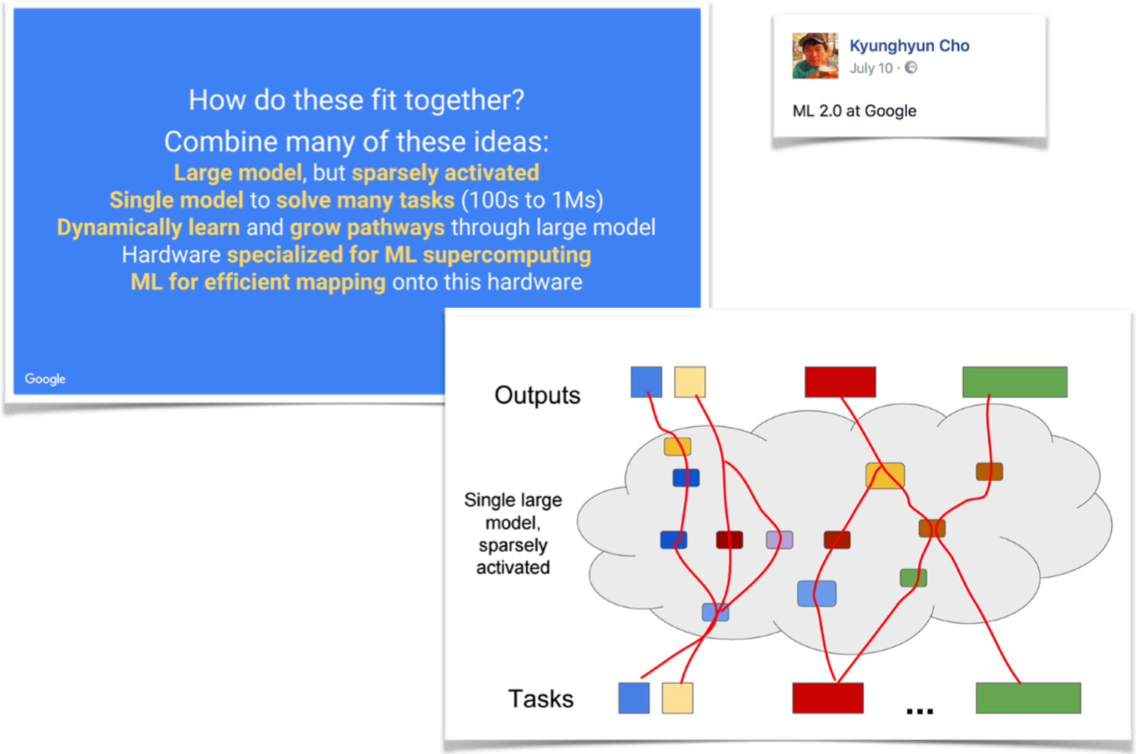
Training Physics-aware supervised neural networks[1][2]
- Embed physical-laws underlying the process
- To be used to infer physical quantities (momenta, directions, energies..)
- Both for continuous and discrete processes

[1] "QCD-Aware Recursive Neural Networks for Jet Physics", Kyle Cranmer et Al, https://arxiv.org/abs/1702.00748 - Feb 2017

[2] Physics Informed Deep Learning: Data-driven Solutions of Nonlinear Partial Differential Equations, Maziar Raissi et Al, https://arxiv.org/abs/1711.10561 - Nov 2017

ONE STEP FURTHER:

- Dynamically (sparsely) activated "**physics-aware**" ML model [3]
  - using a different neural network topology according to the different physics simulation that has to be performed
  - The different paths are activated according to a physics knowledge used to optimize the result in terms of:
    - Accuracy
    - Performance
  - Data-driven discovery?



Slides from Jeff Dean of Google Brain @ Jeju July 2017 https://drive.google.com/file/d/0B8z5oUpB2DysZWF1RTFuX1NEZUk/view

- Final (ambitious) Goal:

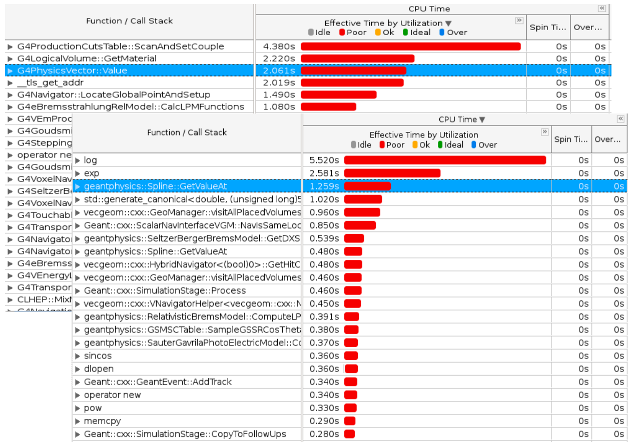
Perform a full HEP Simulation that is faster, more accurate (?) and easy to vectorize.

[3] Deep Learning and Particle Physics, K. Cranmer, https://indico.cern.ch/event/567550/contributions/2656469/attachments/1512016/2358233/ACAT-2017.pdf

BACKUP SLIDES

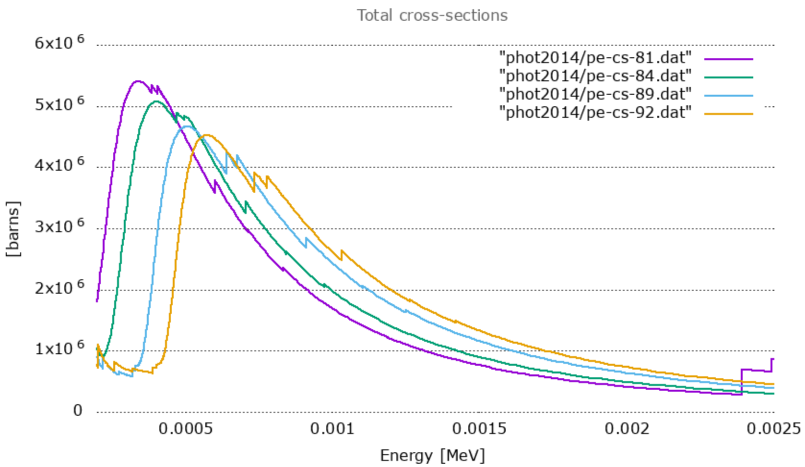
VECTORIZATION, NOT AN EASY JOB

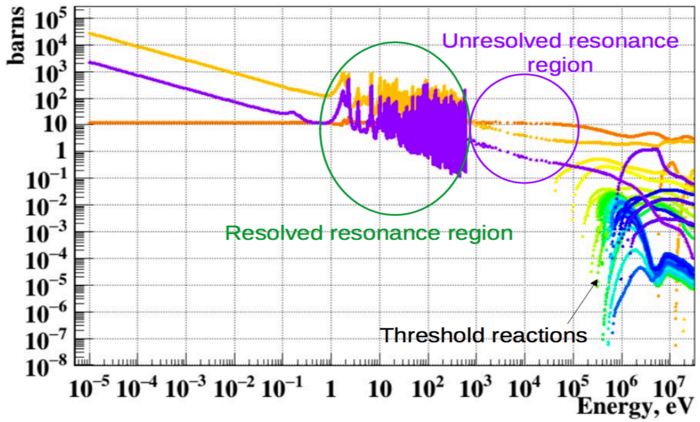
- Profiling of a Geant4/GeantV application is revealing their major hotspots (log/exp/GetValue)
  - Lot of time in the simulation is spent in GetValue from tables (log indexing/binary search + interpolation)
- Can we always use cross-section tables?
  - Photoelectric effect
  - Hadronic physics: Neutron HP (<20MeV)



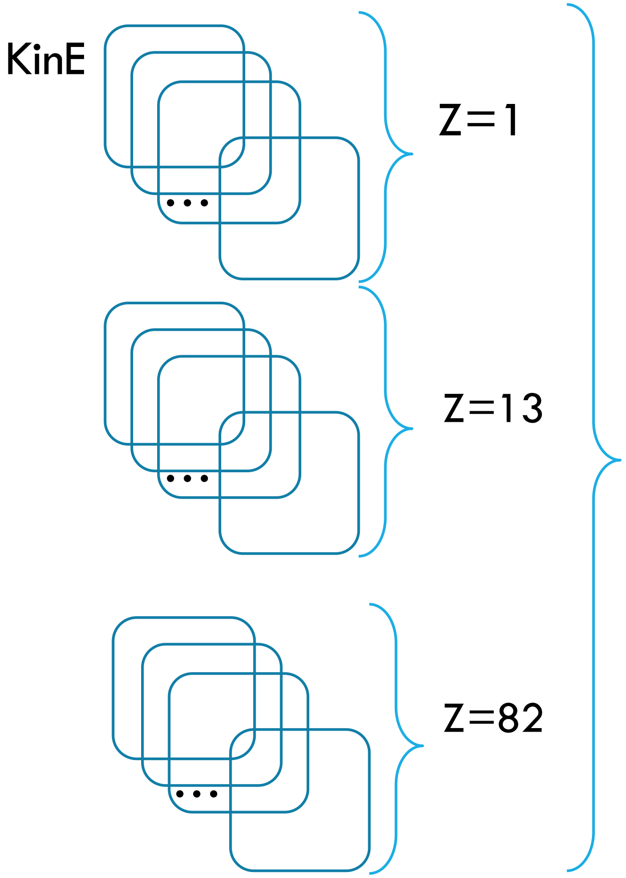
Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Ti...	Over...
G4ProductionCutsTable::ScanAndSetCouple	4.380s	0s	0s
G4LogicalVolume::GetMaterial	2.220s	0s	0s
G4PhysicsVector::Value	2.061s	0s	0s
__tls_get_addr	2.019s	0s	0s
G4Navigator::LocateGlobalPointAndSetup	1.490s	0s	0s
G4eBremsstrahlungRelModel::CalcLPMFunctions	1.080s	0s	0s

Function / Call Stack	CPU Time: Effective Time by Utilization (Idle, Poor, Ok, Ideal, Over)	Spin Ti...	Over...
log	5.520s	0s	0s
exp	2.581s	0s	0s
geantphysics::Spline::GetValueAt	1.259s	0s	0s
std::generate_canonical<double, (unsigned long)5	1.020s	0s	0s
vecgeom::cxx::GeoManager::visitAllPlacedVolumes	0.960s	0s	0s
Geant::cxx::ScalarNavInterfaceVGM::NavIsSameLoc	0.850s	0s	0s
geantphysics::SeltzerBergerBremsModel::GetDXS	0.539s	0s	0s
geantphysics::Spline::GetValueAt	0.480s	0s	0s
vecgeom::cxx::HybridNavigator<(bool)0>::GetHitC	0.480s	0s	0s
vecgeom::cxx::GeoManager::visitAllPlacedVolumes	0.460s	0s	0s
Geant::cxx::SimulationStage::Process	0.460s	0s	0s
vecgeom::cxx::VNavigatorHelper<vecgeom::cxx::N	0.450s	0s	0s
geantphysics::RelativisticBremsModel::ComputeLP	0.391s	0s	0s
geantphysics::GSMSCTable::SampleGSSRCosTheta	0.380s	0s	0s
geantphysics::SauterGavrilaPhotoElectricModel::Co	0.370s	0s	0s
sincos	0.360s	0s	0s
dlopen	0.360s	0s	0s
Geant::cxx::GeantEvent::AddTrack	0.340s	0s	0s
operator new	0.340s	0s	0s
pow	0.330s	0s	0s
memcpy	0.290s	0s	0s
Geant::cxx::SimulationStage::CopyToFollowUps	0.280s	0s	0s





VECTORIZATION, NOT AN EASY JOB




```
sampleFinalStateAlias_v(Real_v  &ekin_v,
                              Index_v<Real_v> &zed,
                              Real_v &r1,
                              Real_v &r2,
                              Index_v<Real_v> &sampledShells)
```

Other criteria
…

```
<vecCore::Gather>
or Scalar
<vecCore::Get>
operations
```

```
Real_v fAliasX(Z, kinE..)
Real_v fAliasY(Z, kinE..)
Real_v fAliasW(Z, kinE..)
```

What is the cost to pay to have data ready in SIMD vectors to be given to a nicely vectorized algorithm?