Neutrino and Air Shower Simulations in IceCube

Juan Carlos Díaz-Vélez

IceCube Simulation Workshop

October 19, 2020

Wisconsin IceCube Particle Astrophysics Center

Workshop Objectives

- Review simulation chain
  - Each lead describe their project, status, challenges.
  - Discuss strategies (shish kabob)
- DYNSTACK CORSIKA development
  - biasing schemes, importance sampling
  - IceTop injector replacement
- Code optimizations.
  - Memory, CPU profiling.
  - DOM oversizing in PPC/CISim
- Resource prediction
  - Memory, runtime, etc.
- Hitspool/DAQ Trigger
- Upgrade/GEN-2 detector simulation
- Review open tickets

Agenda

Monday, 19 October

08:00 → 10:00 **The IceCube Simulation Chain: Overview of projects/modules**

Overview of the IceCube Simulation chain and the various projects/modules involved.

10:00 → 12:00 **The IceCube Simulation Chain: Computing Challenges**

Overview of the IceCube Simulation chain and the various projects/modules involved.

Tuesday, 20 October

08:00 → 10:00 **CORSIKA/MuonGun: DYNSTACK CORSIKA, CORSIKA8, IT simulations**

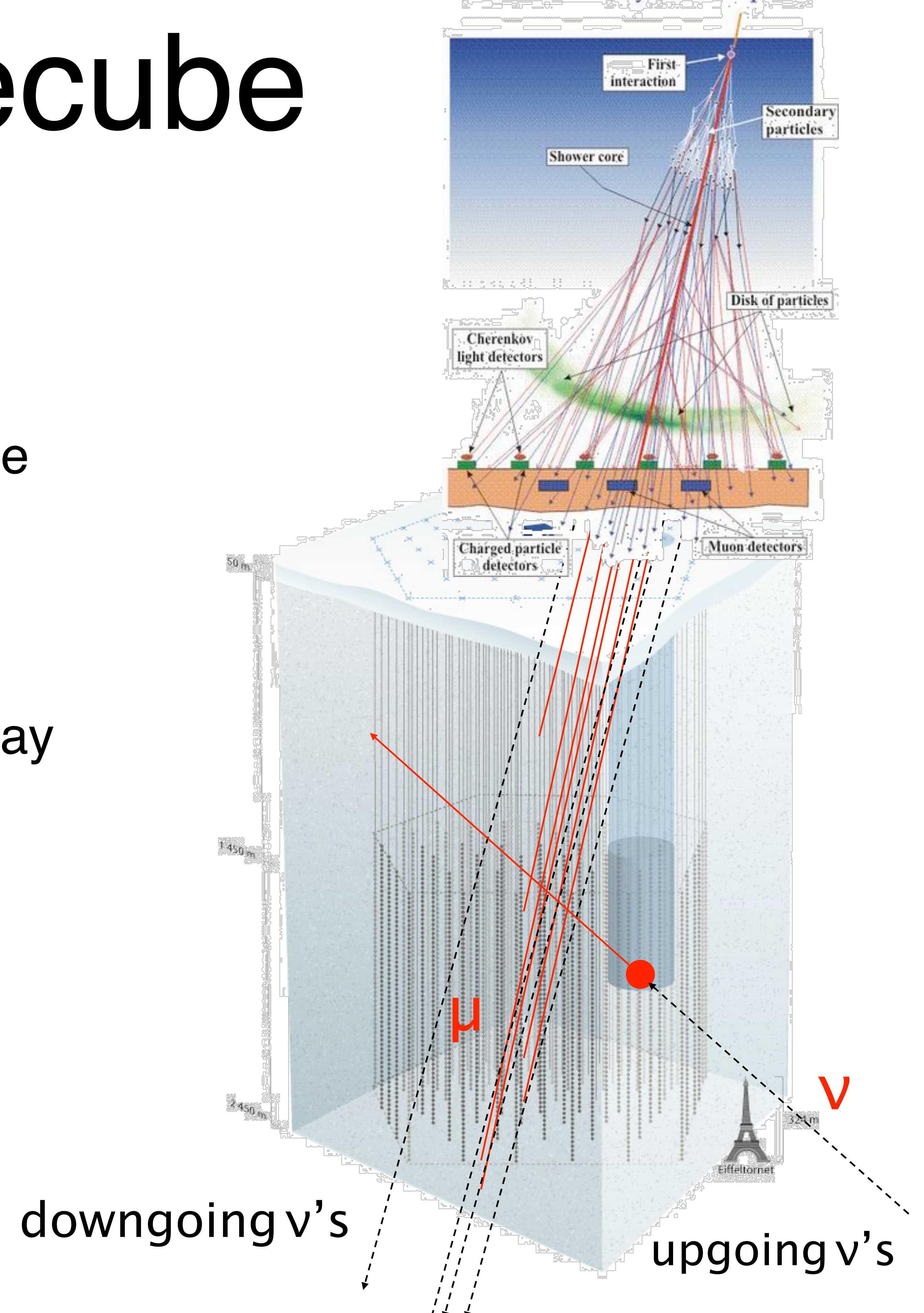
Background CR showers, IT simulation, MuonGun

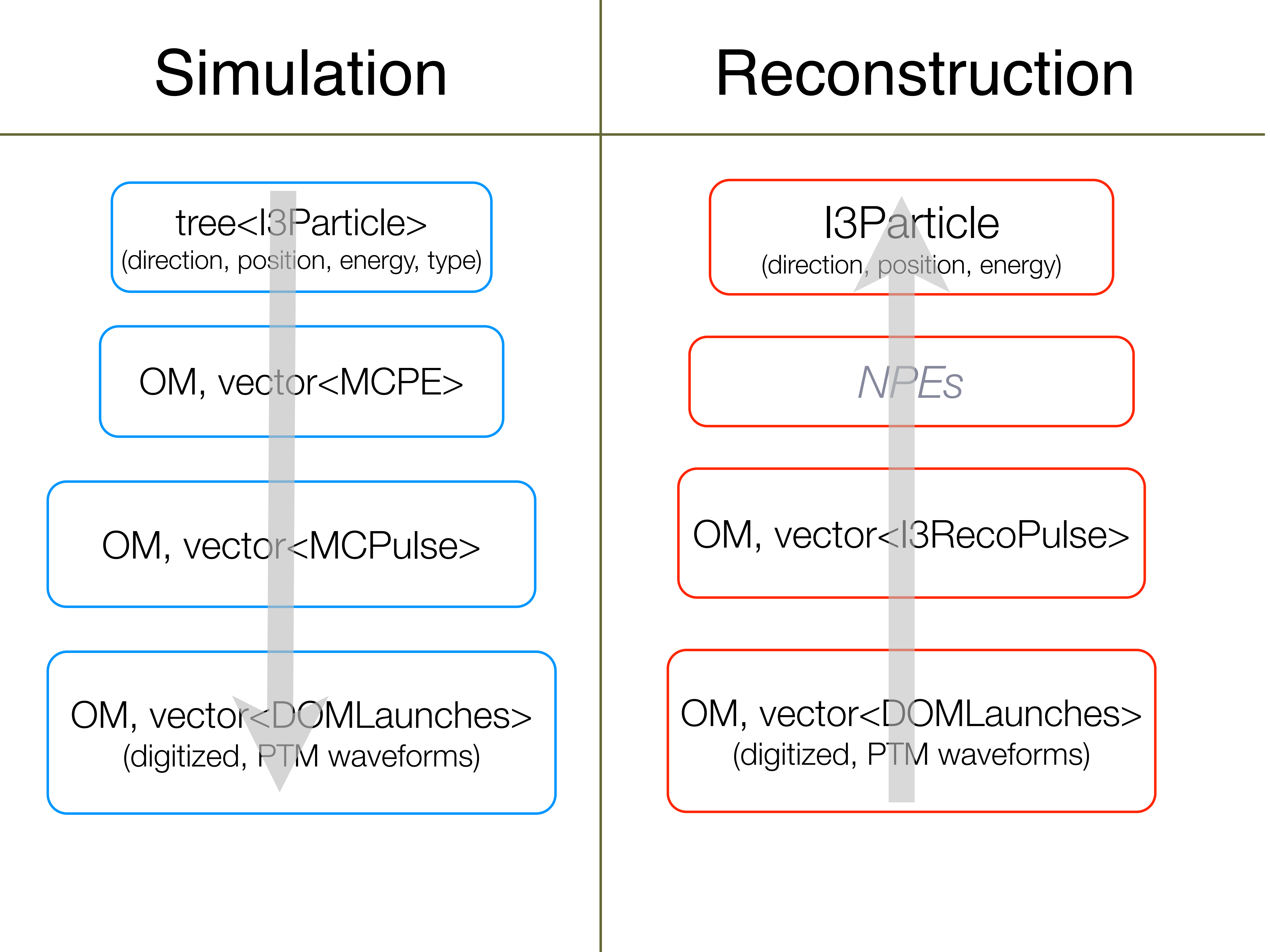
10:00 → 12:00 **Code Optimizations**

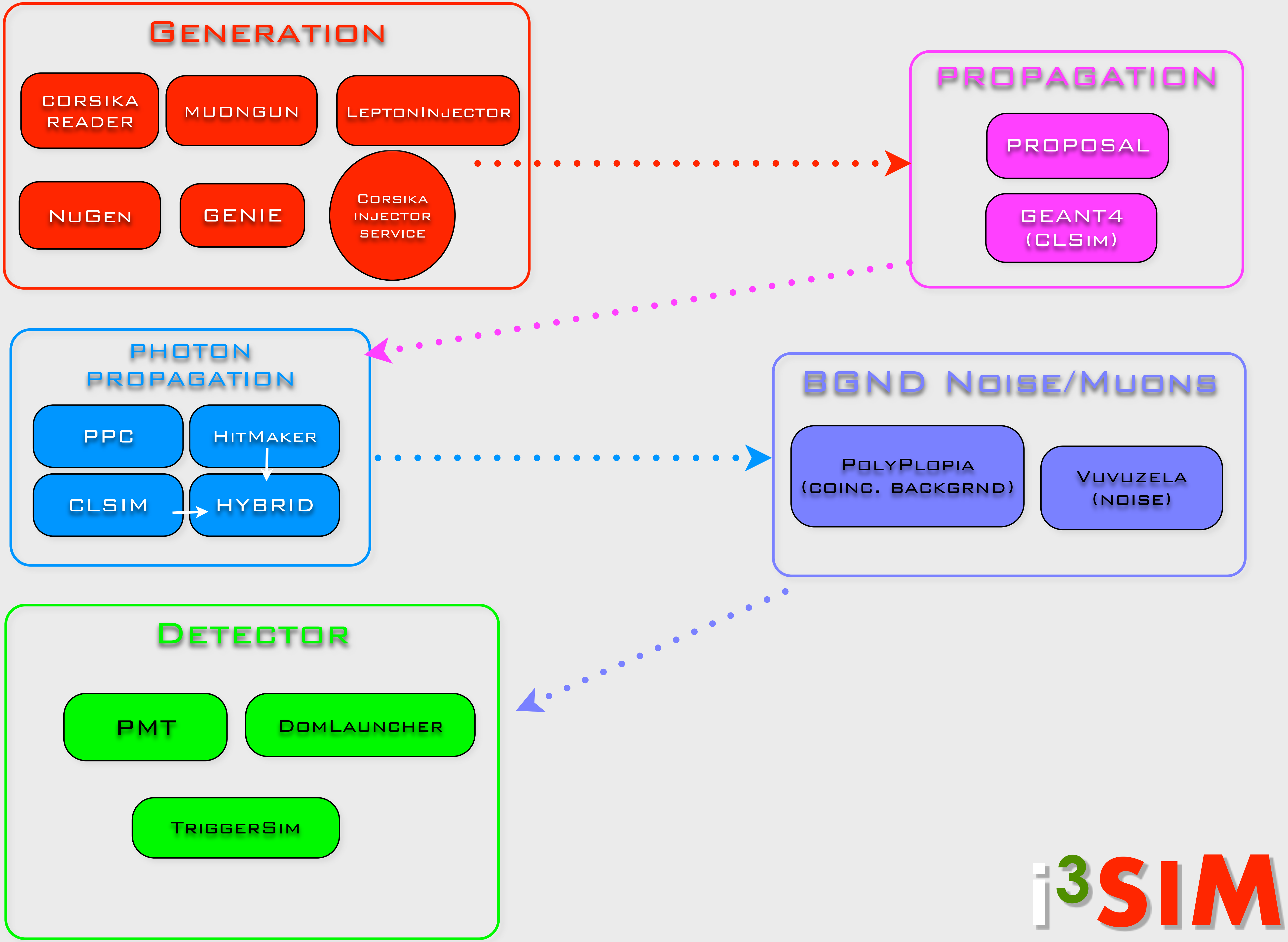
Memory, CPU profiling.
DOM oversizing in PPC/ClSim
Resource prediction
Memory, runtime, etc.

Events in icecube

- Air shower detection @ surface
- Penetrating muon detection in deep ice
- Events dominated by cosmic ray muons : 10^6 μ for every ν that interacts in IceCube
- Atmospheric ν's

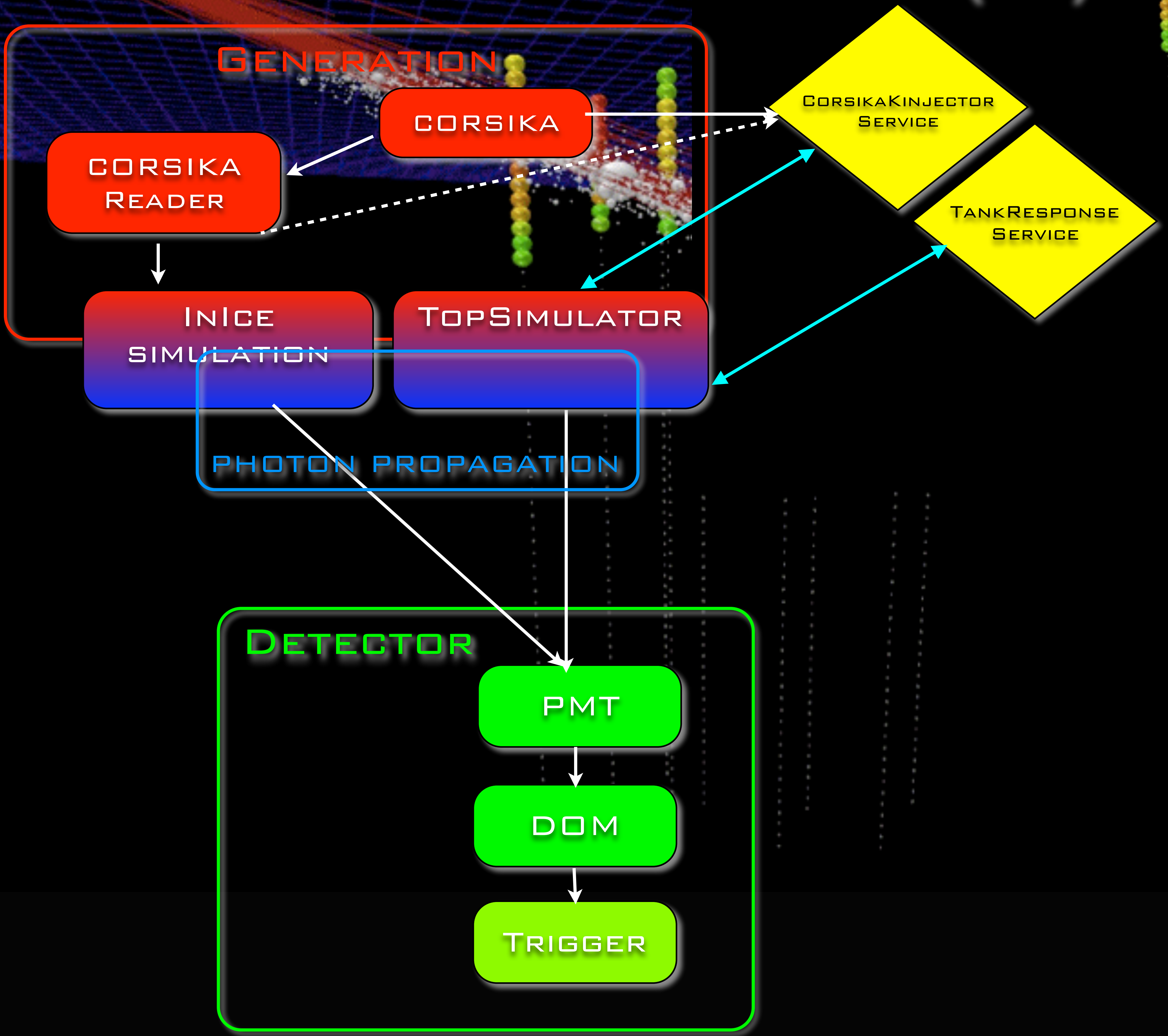



Simulation
Reconstruction
tree<I3Particle>
(direction, position, energy, type)
OM, vector<MCPE>
OM, vector<MCPulse>
OM, vector<DOMLaunches>
(digitized, PTM waveforms)
I3Particle
(direction, position, energy)
NPEs
OM, vector<I3RecoPulse>
OM, vector<DOMLaunches>
(digitized, PTM waveforms)



Generation
CORSIKA READER
MUONGUN
LeptonInjector
NuGen
GENIE
Corsika injector service
PROPAGATION
PROPOSAL
GEANT4 (CLSim)
PHOTON PROPAGATION
PPC
HitMaker
CLSIM
HYBRID
BGND Noise/Muons
PolyPlopia (coinc. backgrnd)
Vuvuzela (noise)
Detector
PMT
DomLauncher
TriggerSim
i3SIM

simulaton chain (IT)



Generators

- Cosmic-ray Air Showers:
  - **CORSIKA** (FORTRAN stand-alone)
  - **corsika-reader**: IceTray reader for standard format
  - **CorsikaInjectorService** (IceTop)
- Muons:
  - **MuonGun**: parametrization of flux of atm. muons under the ice.
- Neutrinos:
  - **neutrino-generator**: injects neutrinos, propagates them through Earth, forces interaction in detector volume.
  - **genie-icetray**: detailed simulation of neutrino interactions with GENIE. (Used for low-energy simulations)
  - **LeptonInjector/LeptonWeighter**: weighted leptons+weights to account for flux models, interaction models, in-earth propagation, etc.

Generators

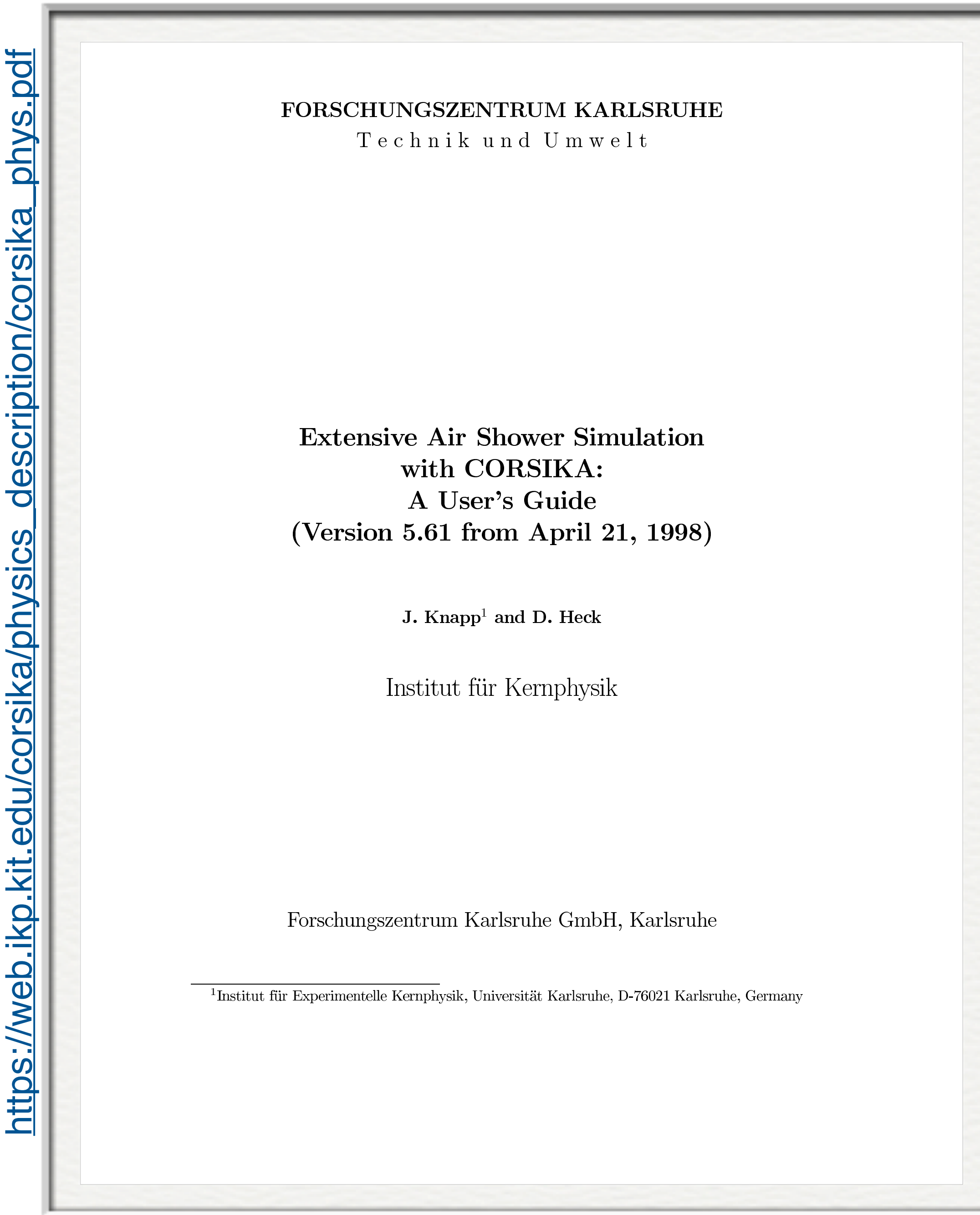
- Cosmic-ray Air Showers:
  - **CORSIKA** (FORTRAN stand-alone)
  - **corsika-reader**: IceTray reader for standard format
  - **CorsikaInjectorService** (IceTop)
- Muons:
  - **MuonGun**: parametrization of flux of atm. muons under the ice.
- Neutrinos:
  - **neutrino-generator**: injects neutrinos, propagates them through Earth, forces interaction in detector volume.
  - **genie-icetray**: detailed simulation of neutrino interactions with GENIE. (Used for low-energy simulations)
  - **LeptonInjector / NuFSGen** (not yet available): weighted leptons+weights to account for flux models, interaction models, in-earth propagation, etc.

Generators : CORSIKA

(**CO**smic **R**ay **SI**mulations for **KA**scade)

- Particles are tracked through the atmosphere until they undergo reactions with the air nuclei or - in the case of instable secondaries - decay.
- The hadronic interactions at high energies may be described by several reaction models alternatively:
  - *VENUS*, *QGSJET*, and *DPMJET* (Gribov-Regge theory),
  - SIBYLL (minijet model).
  - *neXus*, *EPOS* (combination of *QGSJET* and *VENUS*).
  - *HDPM* (Dual Parton Model).
- Hadronic interactions at lower energies:
  - *GHEISHA*, *FLUKA* , or *UrQMD* models.
- For electromagnetic interactions
  - Tailored version of *EGS4*.
  - Analytical *NKG* formulas.

https://web.ikp.kit.edu/corsika/physics_description/corsika_phys.pdf



FORSCHUNGSZENTRUM KARLSRUHE
Technik und Umwelt

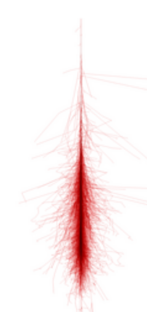
Extensive Air Shower Simulation
with CORSIKA:
A User's Guide
(Version 5.61 from April 21, 1998)

J. Knapp[1] and D. Heck

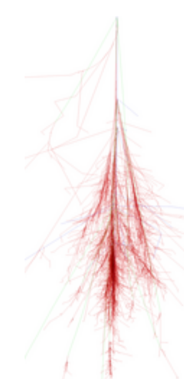
Institut für Kernphysik

Forschungszentrum Karlsruhe GmbH, Karlsruhe

[1]Institut für Experimentelle Kernphysik, Universität Karlsruhe, D-76021 Karlsruhe, Germany

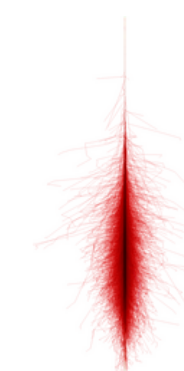


gamma shower



Proton shower

Iron shower



Muon shower

DYNSTACK in CORSIKA

Kevin Meagher & Jakob van Santen

- Replaces CORSIKA's post-reaction particle stack with a C++11 plugin
- General API for doing things like the neutrino kill threshold, plus helpful extras (take configuration from the steering card, manipulate event headers/trailers, etc)
- In mainline CORSIKA since 7.56 (modulo typos)
- Write plugins in C++11 without touching corsika.F, depend only on the standard library
- Build a better CORSIKA for in-ice background simulation.
- Reduce memory and disk requirements for high energy simulations.

Analysis-specific, targeted background simulation

Generators

- Cosmic-ray Air Showers:
  - **CORSIKA** (FORTRAN stand-alone)
  - **corsika-reader**: IceTray reader for standard format
  - **CorsikaInjectorService** (IceTop)
- Muons:
  - **MuonGun**: parametrization of flux of atm. muons under the ice.
- Neutrinos:
  - **neutrino-generator**: injects neutrinos, propagates them through Earth, forces interaction in detector volume.
  - **genie-icetray**: detailed simulation of neutrino interactions with GENIE. (Used for low-energy simulations)
  - **LeptonInjector / NuFSGen** (not yet available): weighted leptons+weights to account for flux models, interaction models, in-earth propagation, etc.

MuonGun (IceCube implementation of MUPAGE)

arXiv:0907.5563v1 [astro-ph.IM] 31 Jul 2009

Atmospheric MUons from PArametric formulas: a fast GEnerator for neutrino telescopes (MUPAGE)

M. Bazzotti[*], **S. Biagi**[*†], **G. Carminati**[*†], **S. Cecchini**[*‡], **T. Chiarusi**[†], **A. Margiotta**[*†], **M. Sioli**[*†] **and M. Spurio**[*†]

[*] *Dipartimento di Fisica dell'Università di Bologna, Viale Berti Pichat 6/2, 40127 Bologna, Italy*
[†] *INFN, Sezione di Bologna, Viale Berti Pichat 6/2, 40127 Bologna, Italy*
[‡] *INAF-IASF, Via Gobetti 101, 40129 Bologna, Italy*

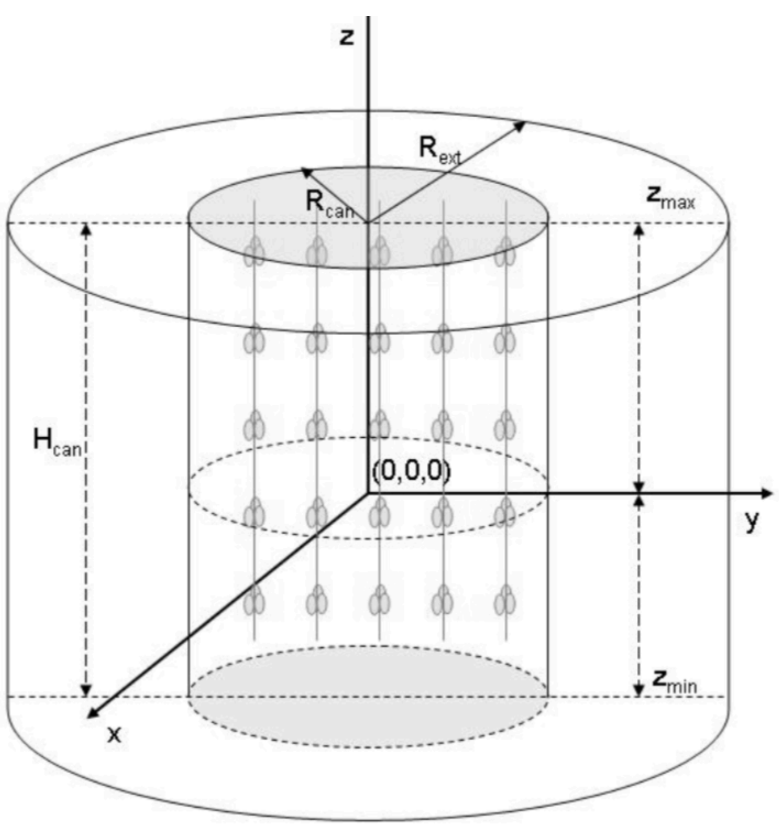
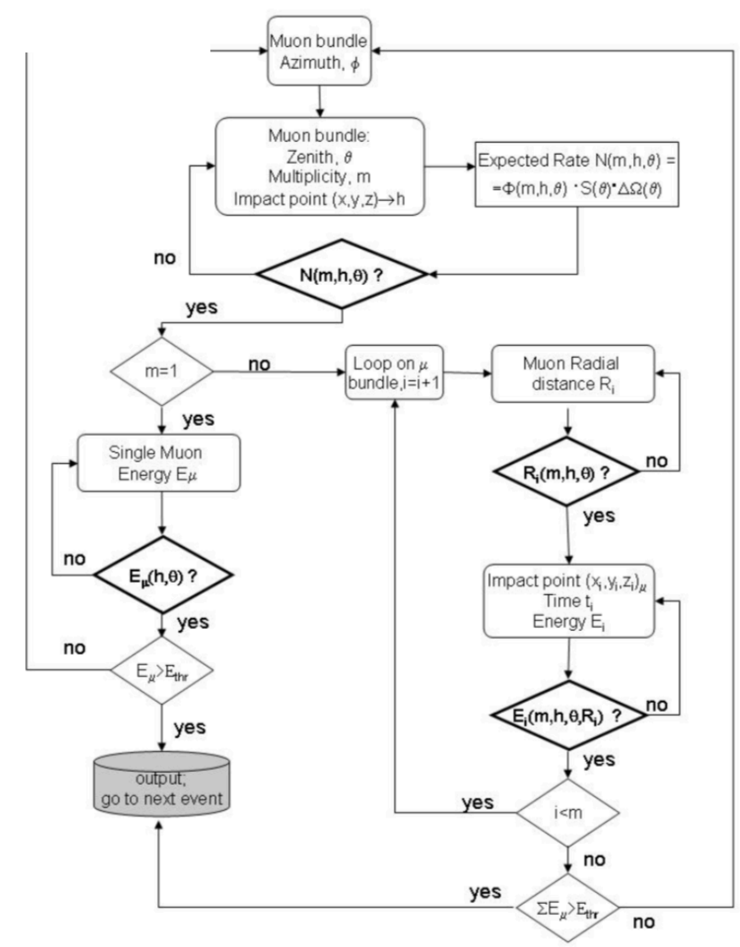


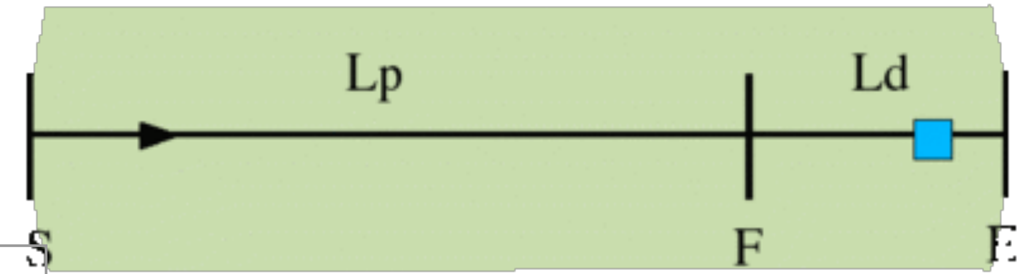
Fig. 1: Sketch of some input parameters. The cylinder surrounding the instrumental volume is the *can*, with radius $R_{\rm can}$ and height $H_{\rm can}$. The events are generated on an extended can with $R_{\rm ext}$. The origin of the coordinate system does not have to be located at the center of the detector. The lower disk is at a depth $H_{\rm max}$ with respect to the sea/ice surface.



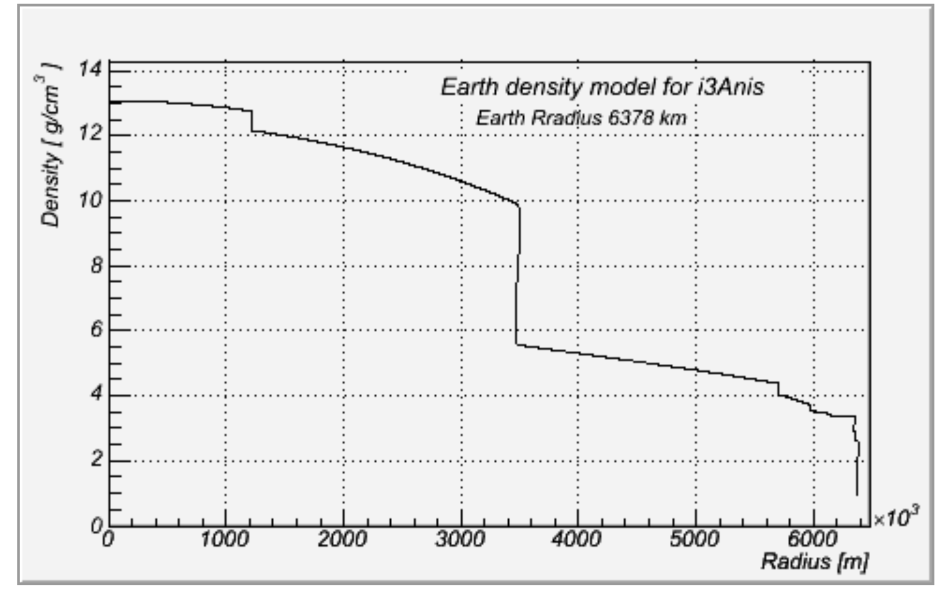
Generators

- Cosmic-ray Air Showers:
  - **CORSIKA** (FORTRAN stand-alone)
  - **corsika-reader**: IceTray reader for standard format
  - **CorsikaInjectorService** (IceTop)
- Muons:
  - **MuonGun**: parametrization of flux of atm. muons under the ice.
- Neutrinos:
  - **neutrino-generator**: injects neutrinos, propagates them through Earth, forces interaction in detector volume.
  - **genie-icetray**: detailed simulation of neutrino interactions with GENIE. (Used for low-energy simulations)
  - **LeptonInjector / NuFSGen** (not yet available): weighted leptons+weights to account for flux models, interaction models, in-earth propagation, etc.

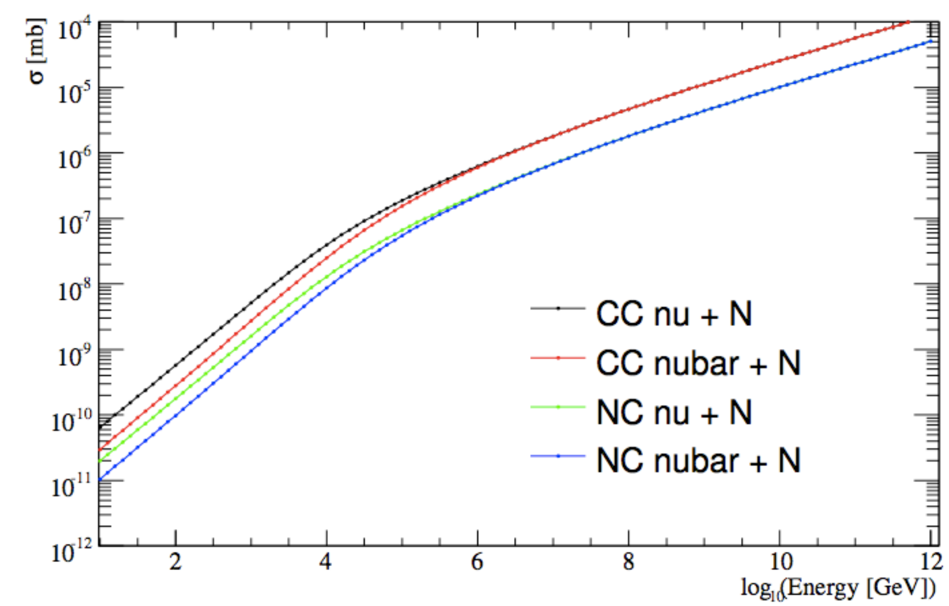
neutrino-generator (fased out)



1. Calculate total path length inside the Earth using injected neutrino geometry.
   a. Separate the total path length into propagation area (SF) and detection volume (FE).
2. Define a step length dx[m] using propagation area and step number.
3. For each step:
   a. Calculate a column depth and Earth's density at the step point.
   b. Calculate a total cross section at the step point.
   c. Calculate a probability that the injected neutrino interacts within the step. Try Monte-Carlo, decide if an interaction happened.
   d. If interaction occurred: choose interaction randomly.
      i. If CC-interaction is selected with injection particle NuMu or NuE, break (event is killed).
      ii. else, generate secondaries and continue to next step.
4. Finish propagation when injected neutrino + secondaries reach surface of detection volume (point F), then process a weighted interaction.



- produce a $E^{-\gamma}$ ν_μ, ν_e, ν_τ with
  - PRELIM Earth's density model



  - parton distribution functions
  - prop & interaction of neutrinos into a weight

Lepton propagation

https://doi.org/10.1016/j.cpc.2013.04.001

- PROPOSAL: parametrized interactions with the medium. Comp. Phys. Com. 184, 9 (2013), p2070-2090
  - Stochastic energy losses include:
    - ionization
    - electron-pair production
    - bremsstrahlung
    - photo-nuclear interaction
    - decay
- GEANT4: Detailed particle propagation in media. https://geant4.web.cern.ch/
  - 3rd-party G4 library used by CLSim to propagate leptons for low-energy simulations (CPU-intensive).

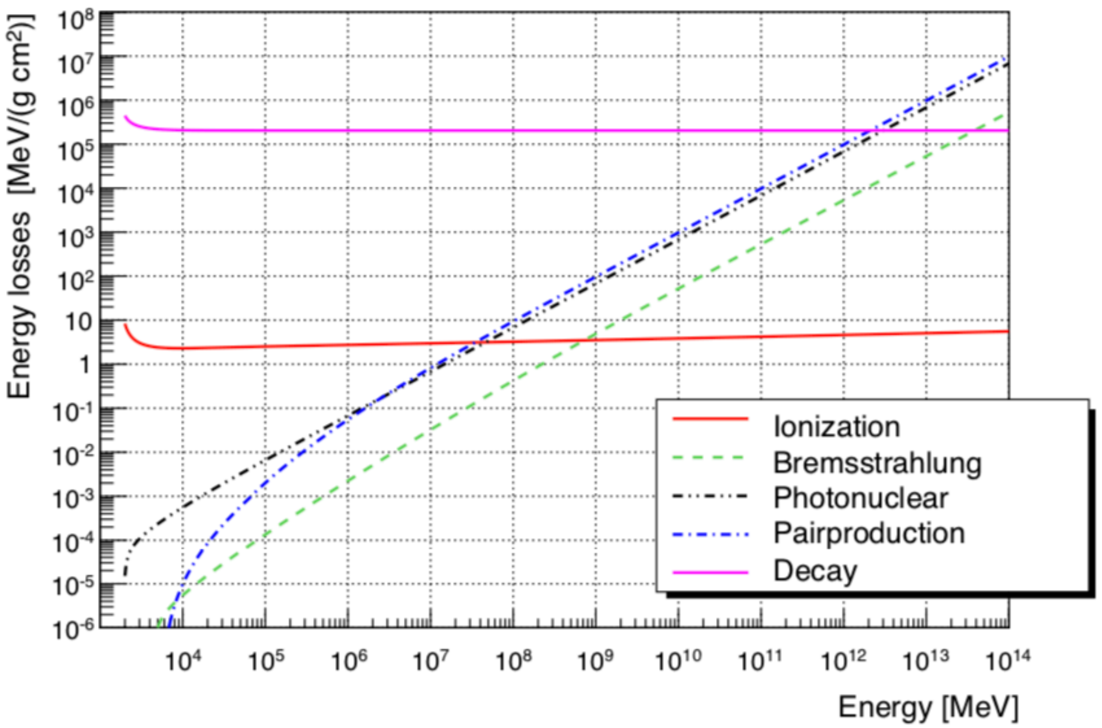


Fig. 31. Continuous energy loss of taus in ice in the energy range from 10^3 MeV to 10^{14} MeV. The graph shows the energy losses of the four interactions and the probability of decay multiplied by the primary particle energy.

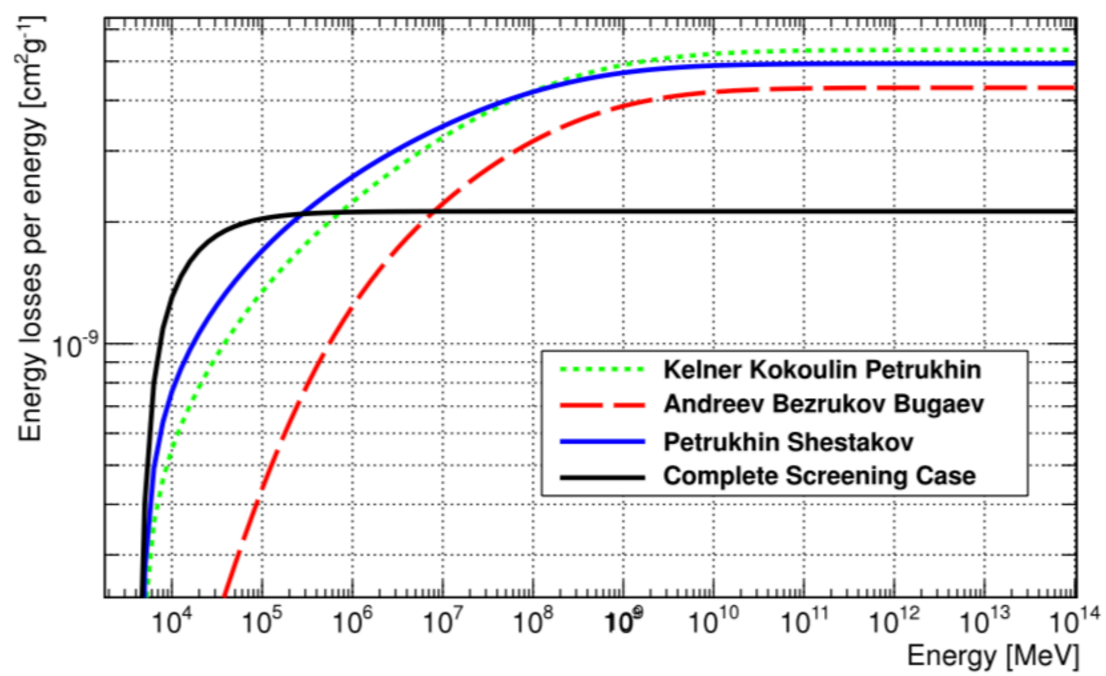
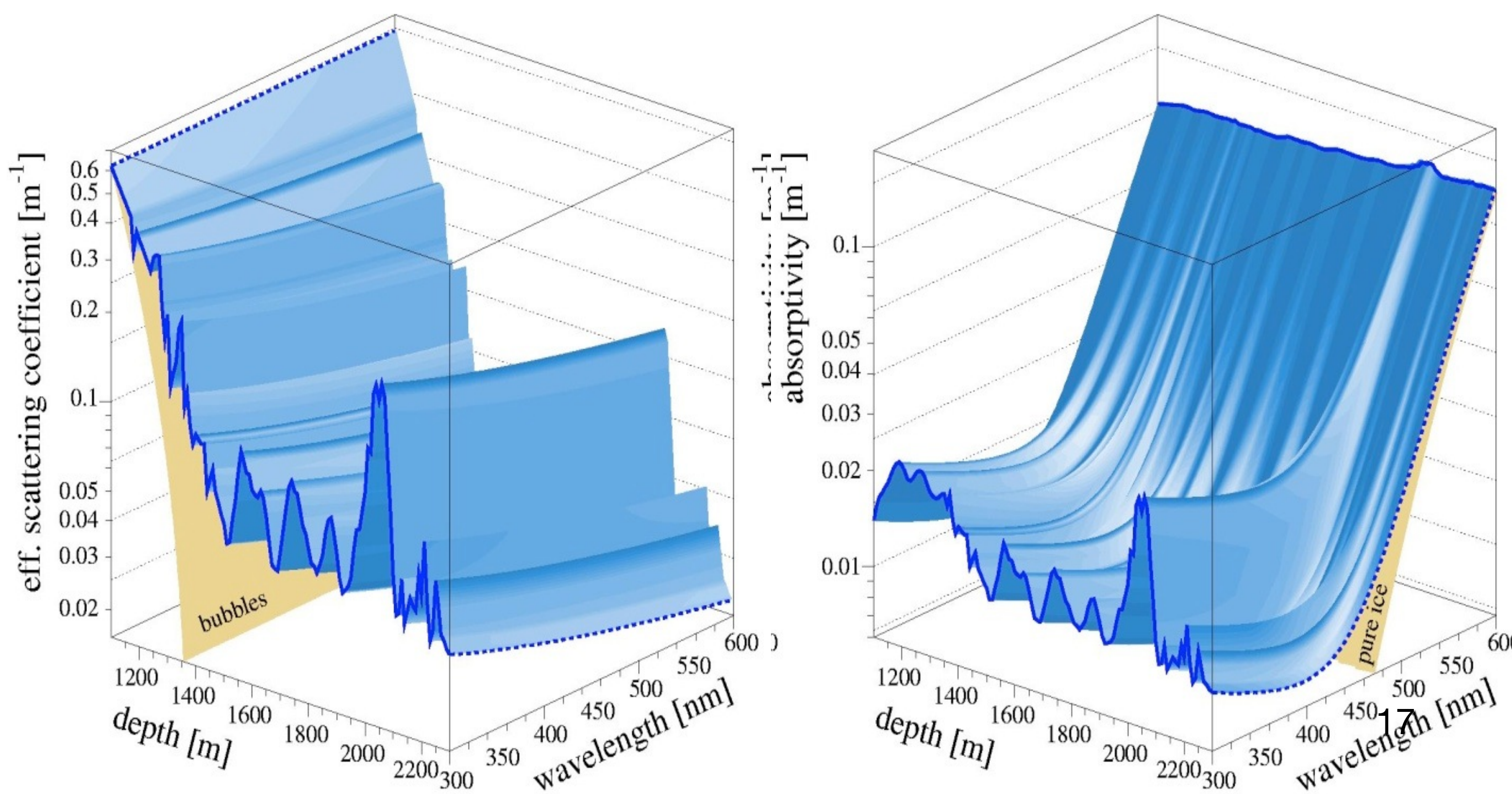


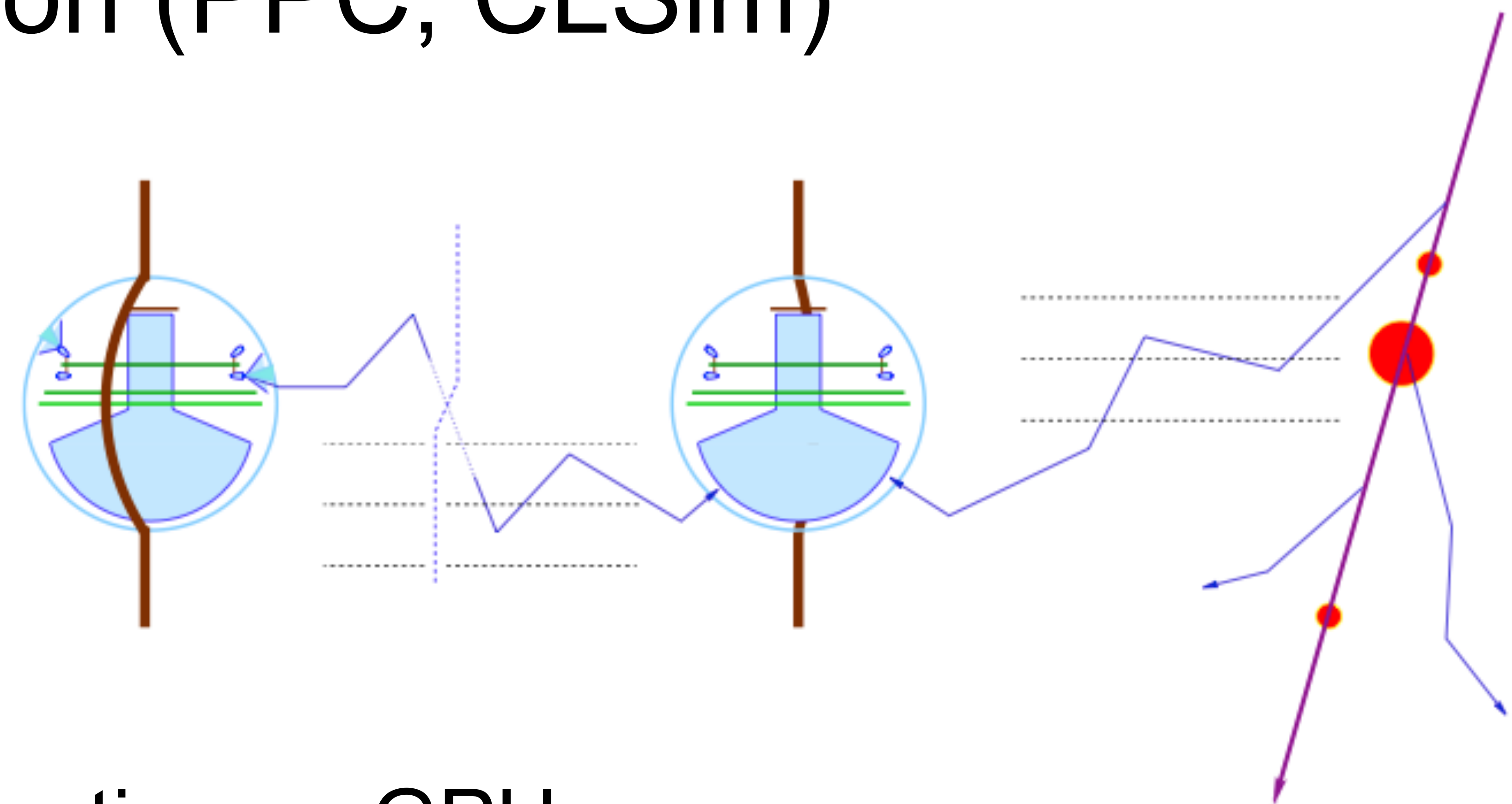
Fig. 4. Continuous energy loss of taus caused by Bremsstrahlung in the energy range from $2 \cdot 10^3$ MeV to 10^{14} MeV. The figure shows the same four possible parametrizations as Fig. 2.

Photon Propagation

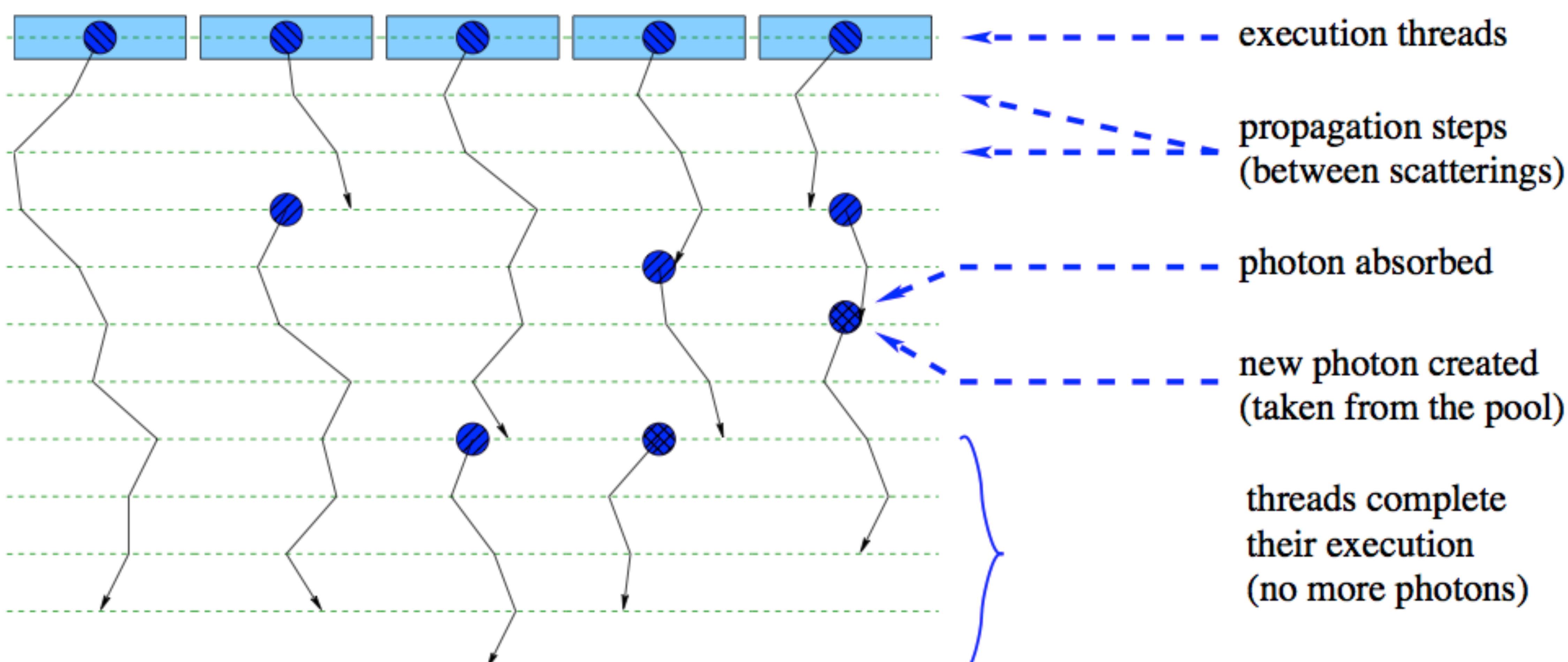
- μ energy lost + cascades ➙ photons ➙ p.e.
  - Photon propagation : ice properties + PMT response + DOM glass/gel
    - Pre-generated lookup splined table :
      - I3PhotonicsHitMaker
      - Amplitude and time distribution
    - Direct photon tracking
      - CLSim
      - PPC
    - Hybrid photon tracking
      - HitMaker + CLSim



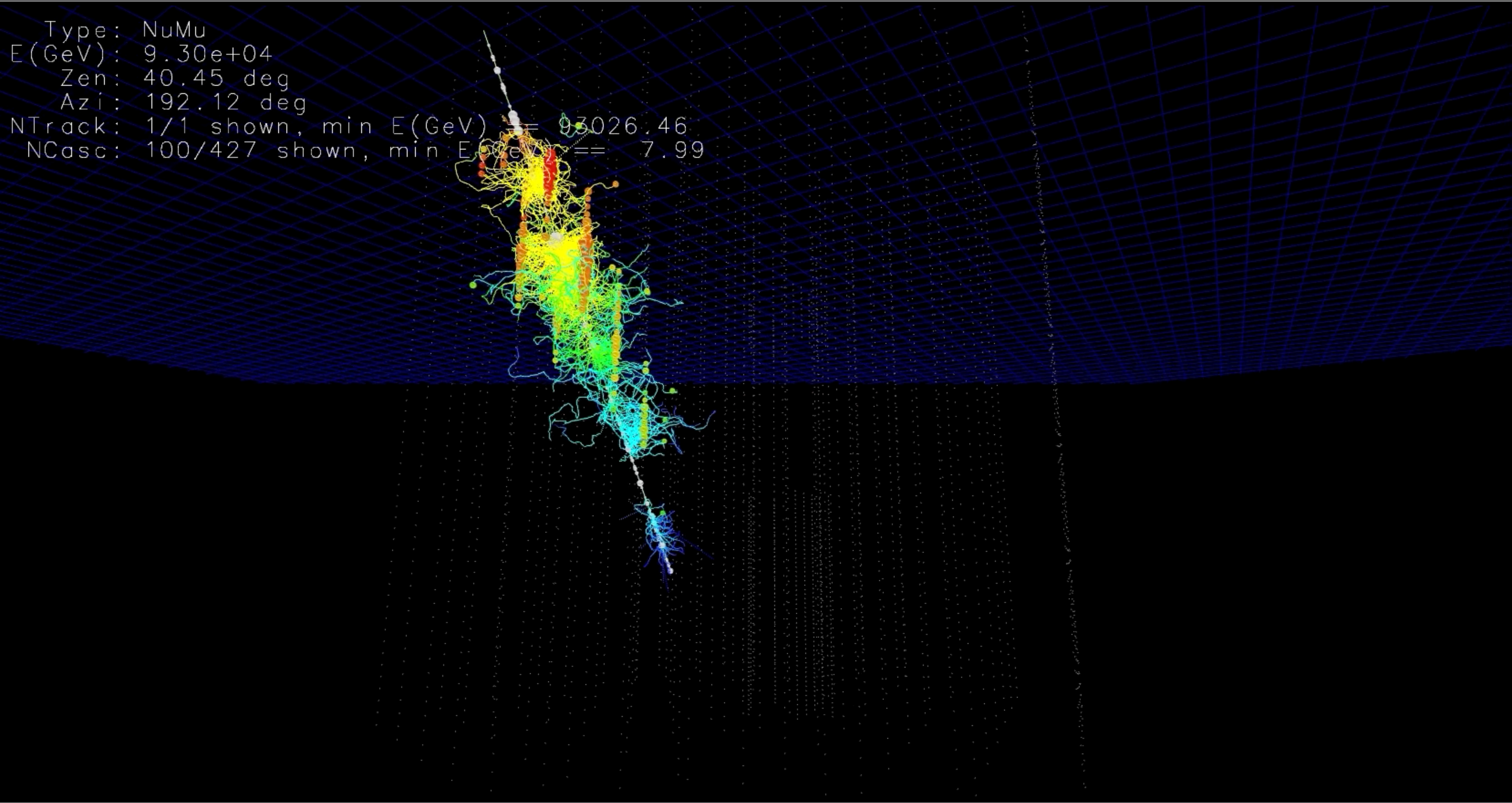
Photon Propagation (PPC, CLSim)



Direct photon propagation on GPU



Photon Propagation (PPC, CLSim)

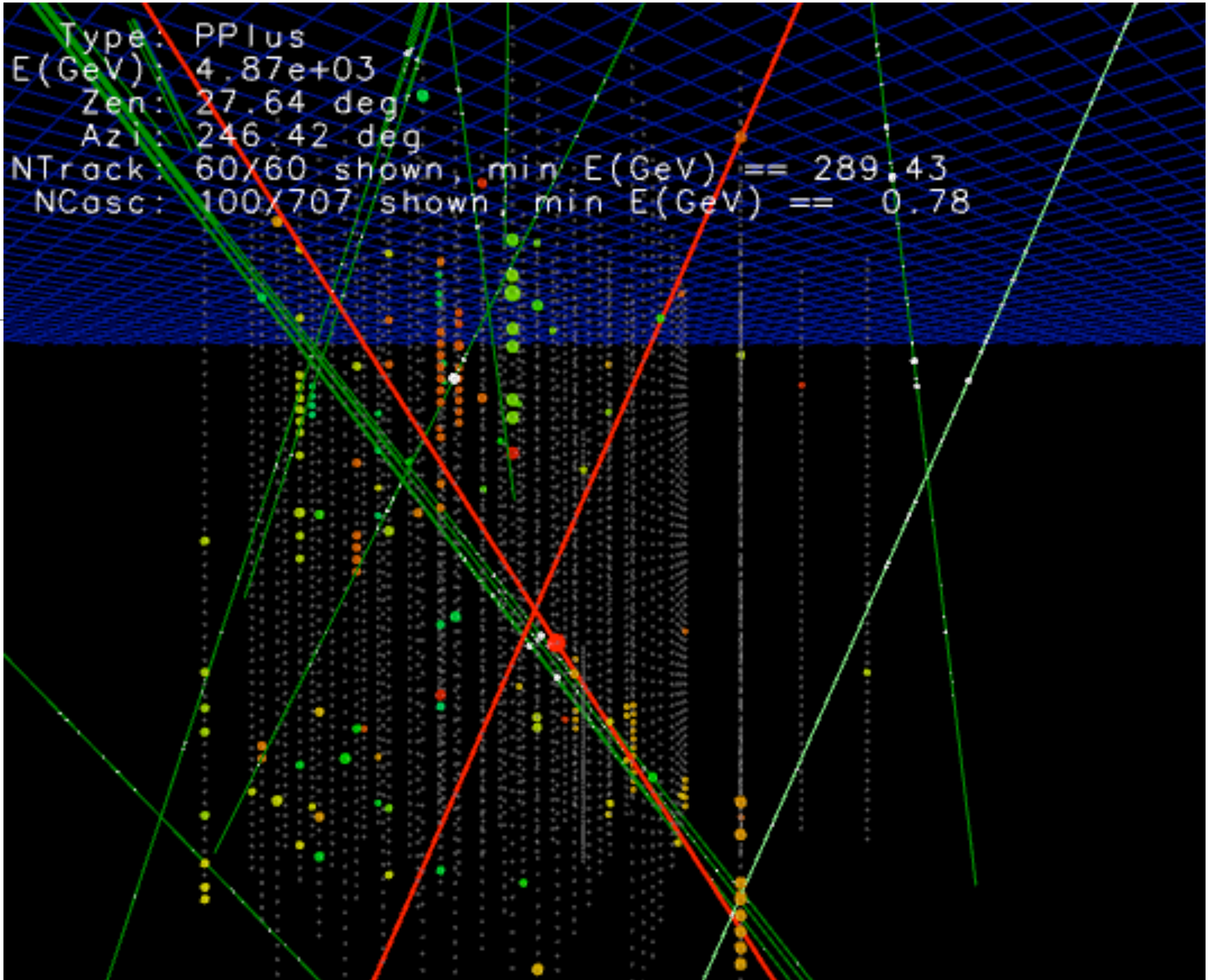


http://icecube.wisc.edu/~ckopper/muon_with_photons.mov

diplopia

(from gr. διπλόος, "double", and ὄψ, ὀπός, "vision")

coincident atmospheric shower events in IceCube



Noise Generation

—> (MCPEs)

Noise Model

Thermal Noise (~few Hz)
[Poisson process]

~ ms Timescales

+

Radioactive Decay in Glass
[Poisson process]

~ ms Timescales

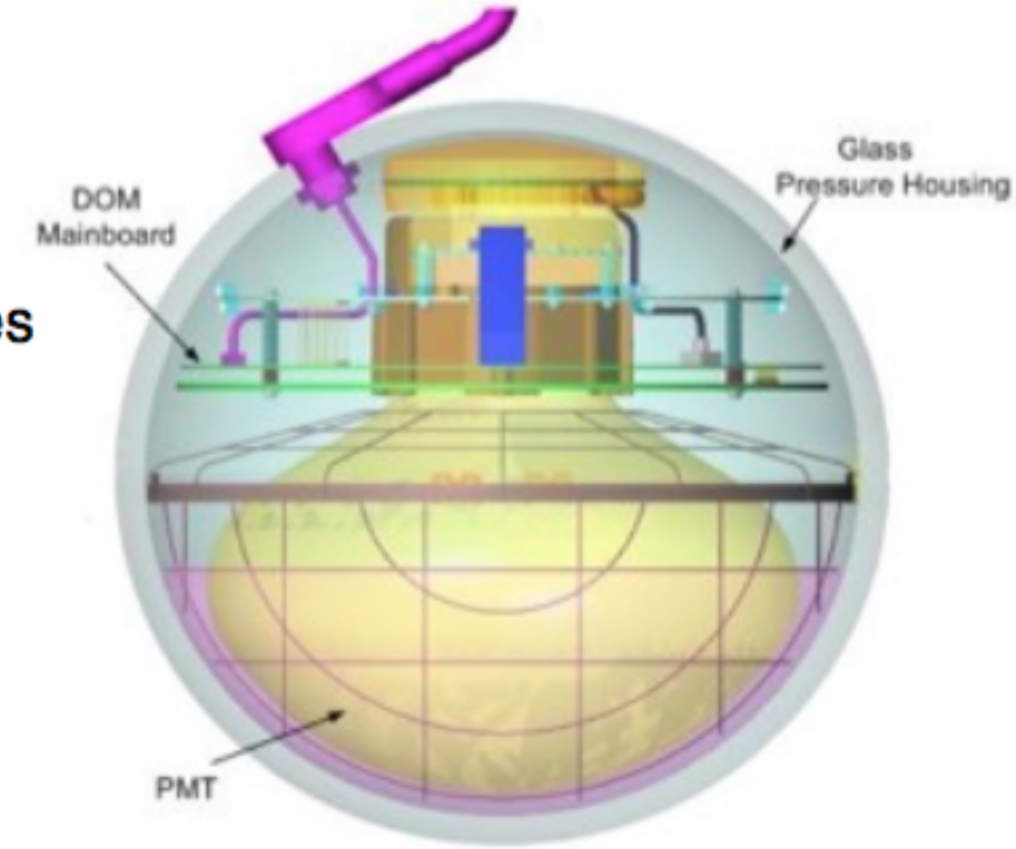
↓

Energy deposited in glass

↓

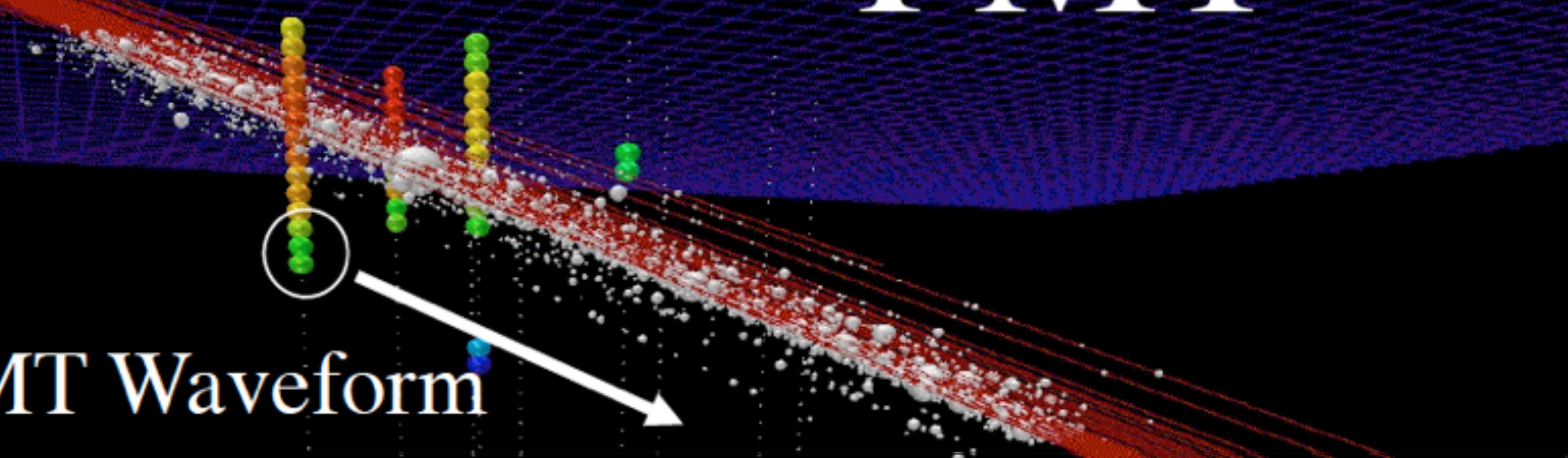
Glass scintillates/fluoresces
over long timescale
[Log-normal]

≲ 500 µs Timescales



DOMLauncher::
PMTResponseSimulator

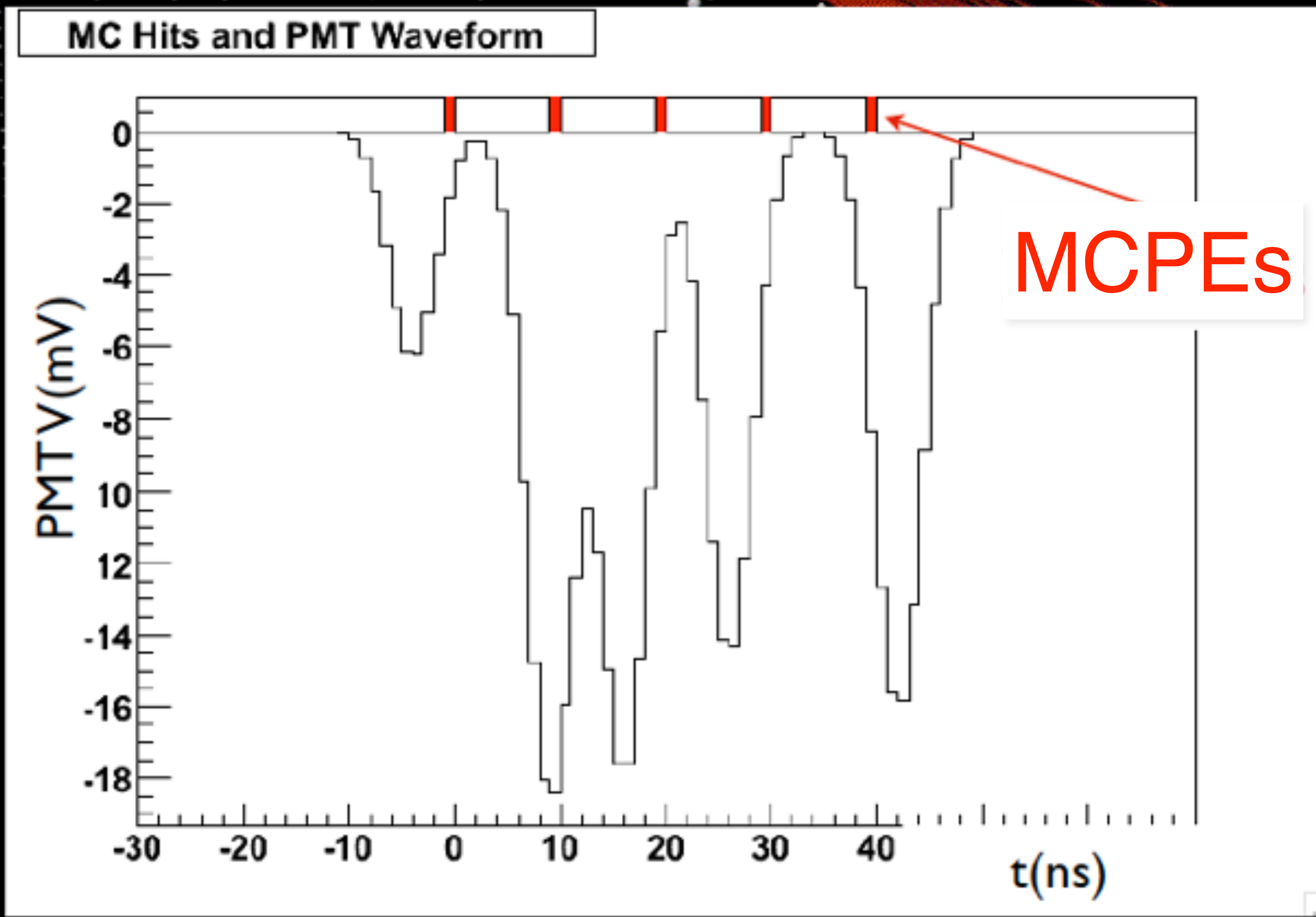
PMT



Generates PMT Waveform

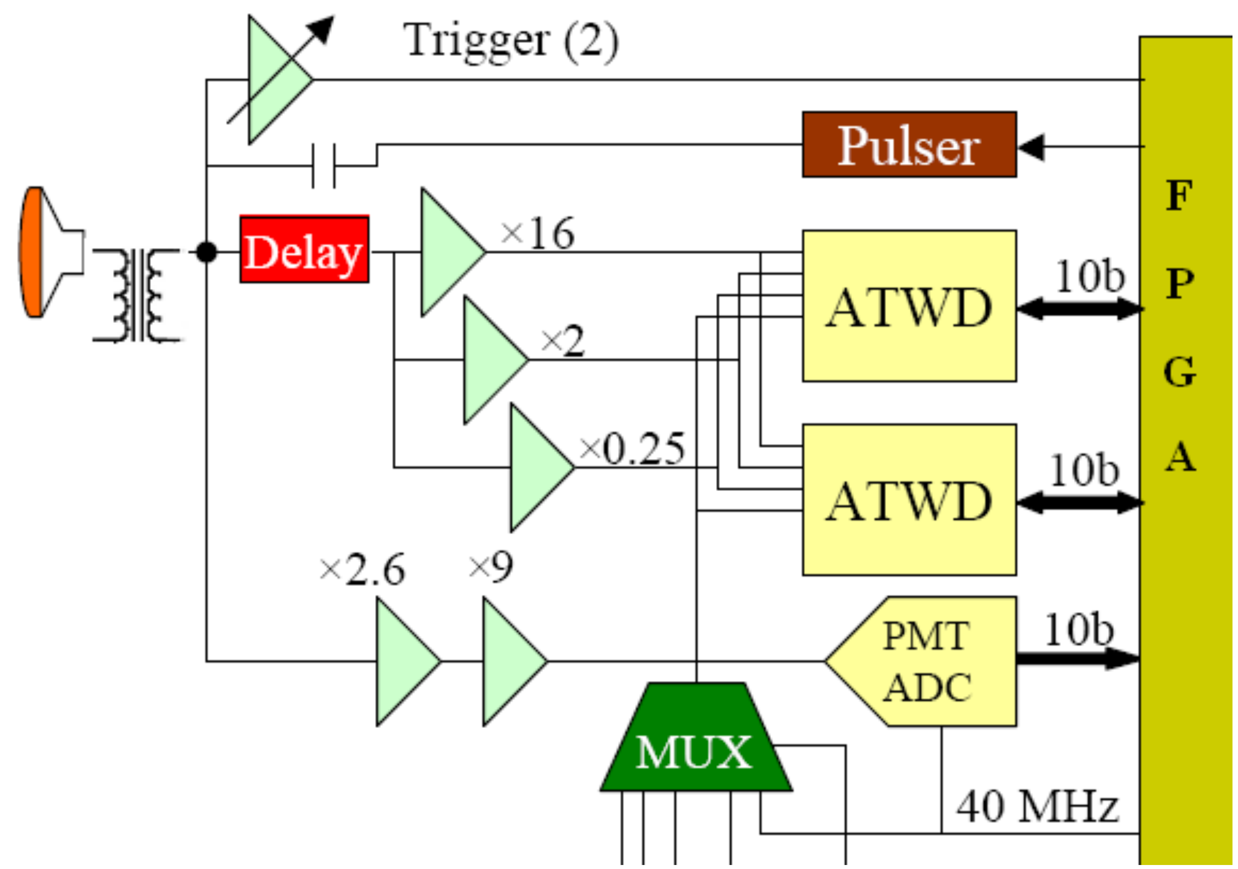
From distribution of (combined) MCPEs.

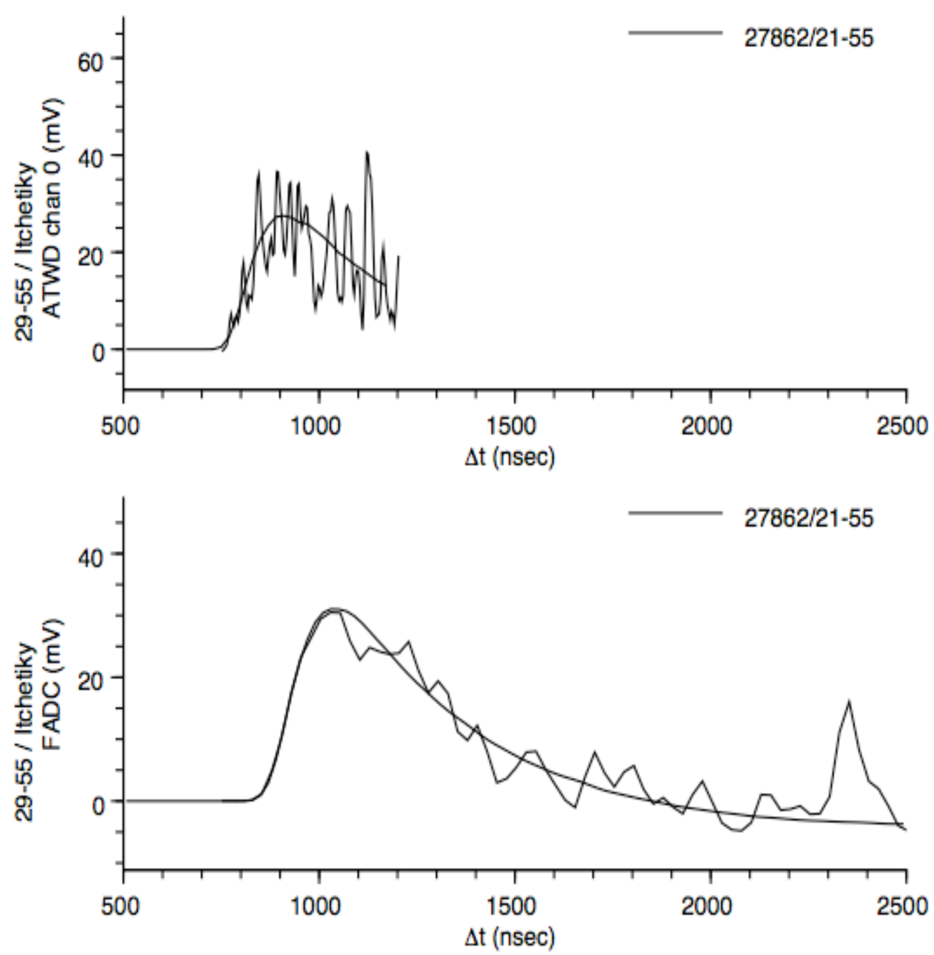
Outputs I3MCPulseSeries for each DOM.

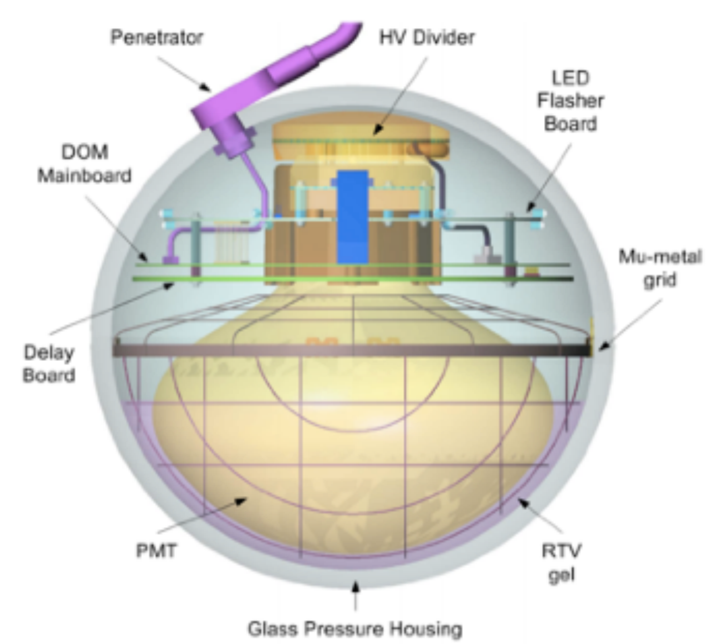


DOMLauncher: DOM electronics simulation

- Discriminator
- LC-logic
- Digitization
- Simulated effects
  - Electronic noise in the digitizers
  - Beacon launches (CPU triggered launches)
  - The FPGA Clock phase
  - RAPcal time uncertainty







Trigger Simulation

- **S**imple **M**ultiplicity **T**rigger (SMT)
  - *N* HLC hits or more in a time window
  - Example: InIce SMT8 with N_hits ≥ 8 in 5 μs
  - readout window around this captures early and late hits (-4 μs, +6 μs)

- **String** trigger (a.k.a. Cluster trigger in DAQ-land)
  - *N* HLC hits out of *M* DOMs on a string in a time window
  - Example: 5 hits from a run of 7 adjacent DOMs in a time window of 1500 ns

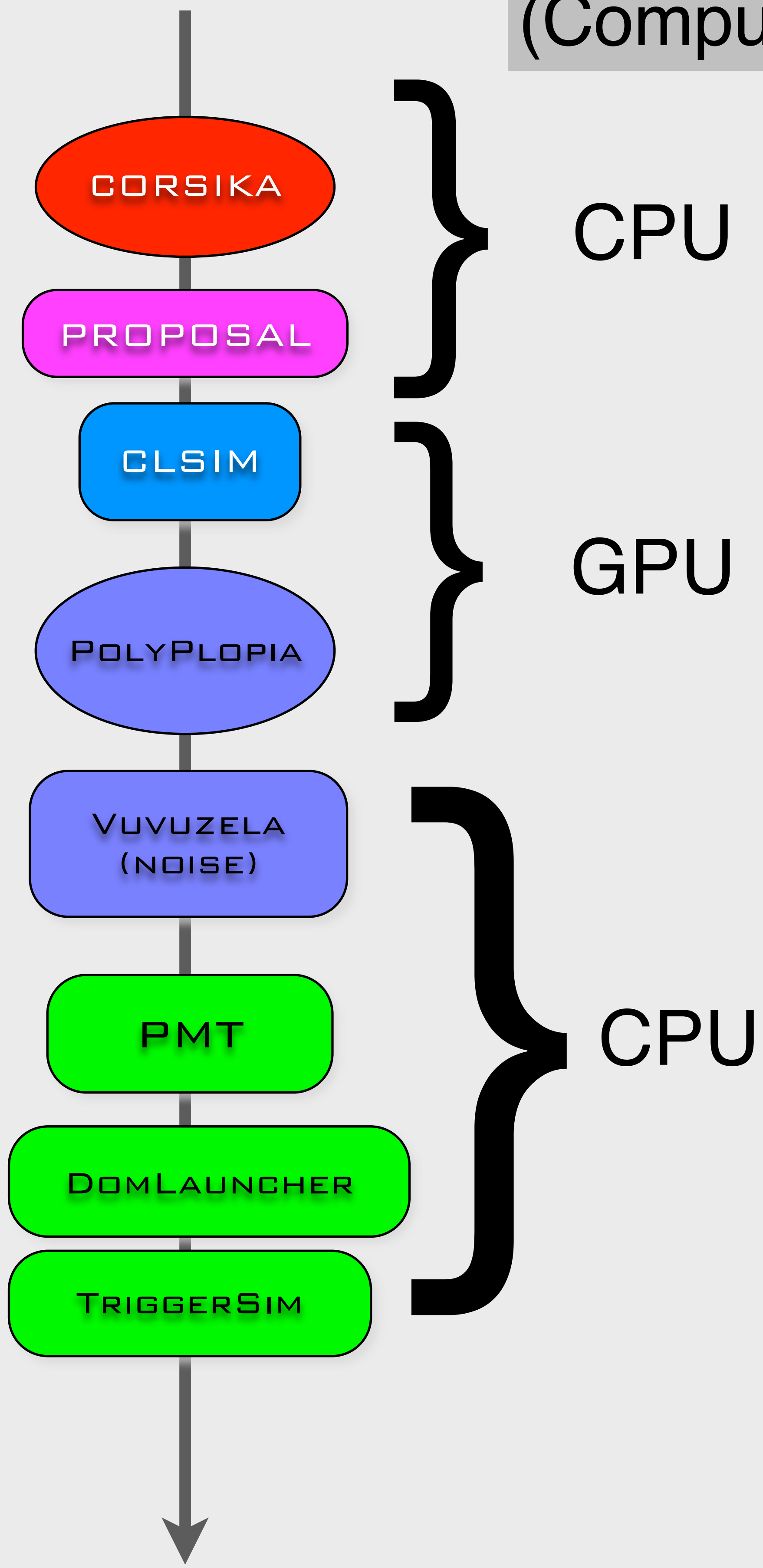
- **Volume** trigger (a.k.a Cylinder trigger in DAQ-land)
  - simple majority of HLC hits (SMT4) with volume element including one layer of strings around a center string
  - cylinder height is 5 DOM-layers (2 up and down from the selected DOM).

- **Slo**w **P**article trigger (SLOP)
  - slow-moving hits along a track
  - lengths of the order of 500μs and extending up to milliseconds

- ~~**F**ixed **R**ate trigger~~, **Min**imum **Bias** trigger, ~~**Calibration** trigger~~

J. Kelley - DAQ

The Shish Kabob
(Computing Resource Optimization)

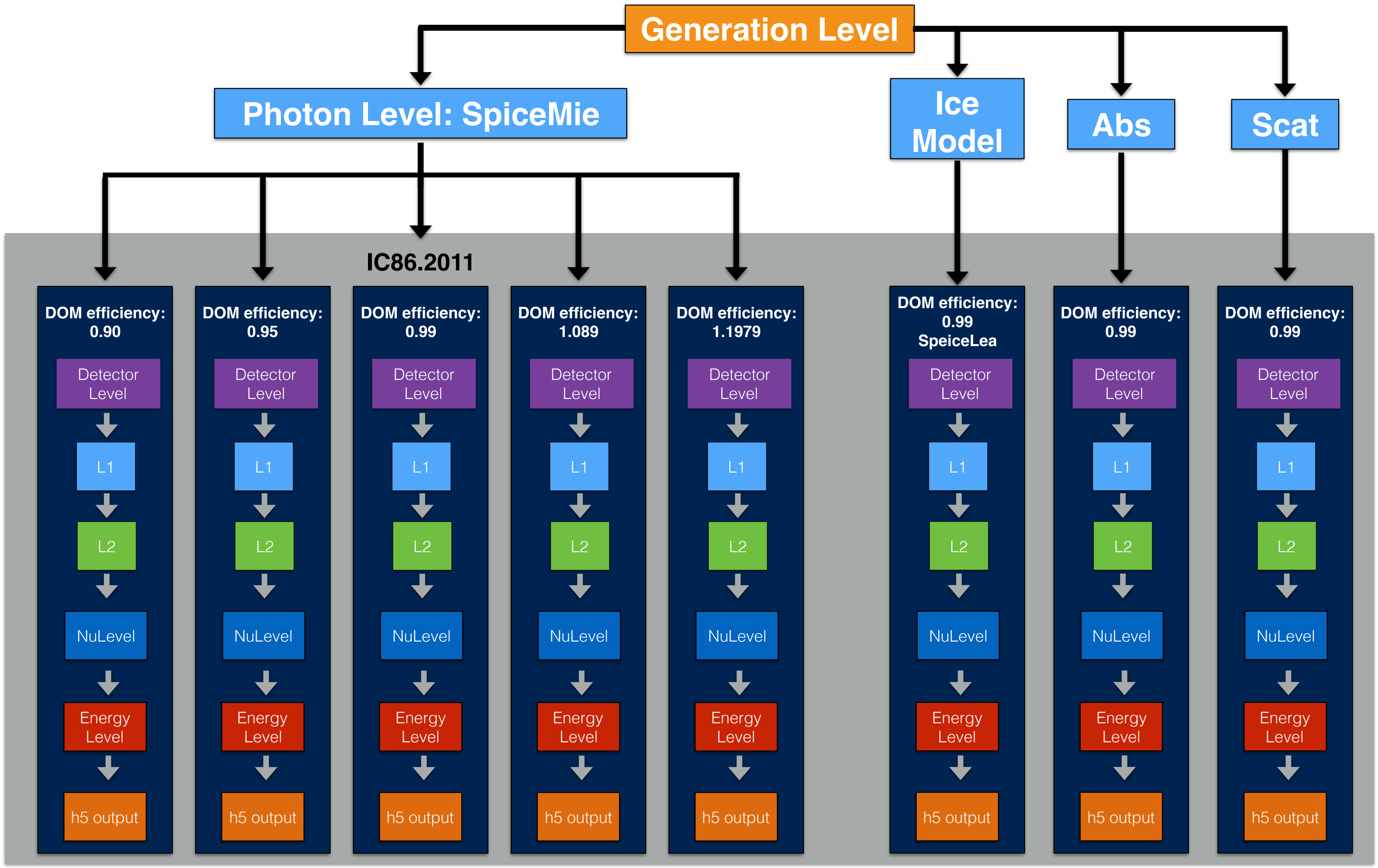


- Optimizing the shish kabob:
  - Different parts of the simulation chain have different resource requirements.
    - CORSIKA is CPU-intensive and requires little RAM
    - Photon propagation run almost exclusively on GPUs
    - Detector simulation is CPU bound and requires more memory.
- Things to keep in mind:
  - Running the whole chain on a GPU node will waste GPU resources and limit your throughput.
  - Intermediate storage:
    - breaking up chain requires transfering/storing intermediate files.
  - Reduce complexity in workflow

Simulating Systematic Uncertainties

Example: High-Energy Sterile Neutrino MC Generation

Spencer N. Axani



Generation level:

- MuonInjector
- Spectrum = E^{-2}
- Energy = 2E2 to 1E6 GeV
- NEvents = 1.2e9 events

Photon Level:

- DOM efficiency: 1.1979
- SpiceMie

It took almost a year to produce this MC for the IC86.2011 analysis.

We do not have the resources to do this for a 6 year analysis.

We need to find ways to optimize and cut back!

SnowStorm

https://events.icecube.wisc.edu/event/118/contributions/6499/attachments/5362/6082/DiffuseParallel_Brussels_SnowStormMCGlobalfit.pdf

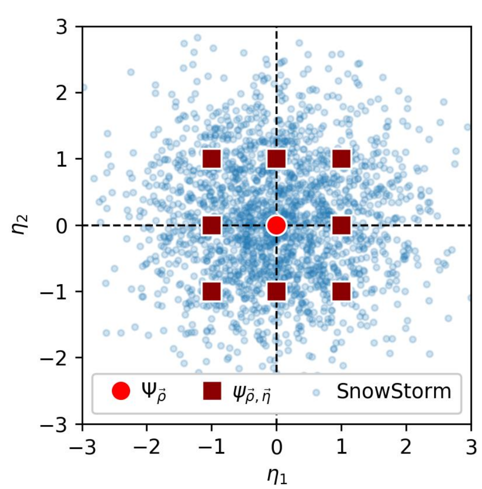
Erik Ganster

SnowStorm Simulation Chain – SnowStorm

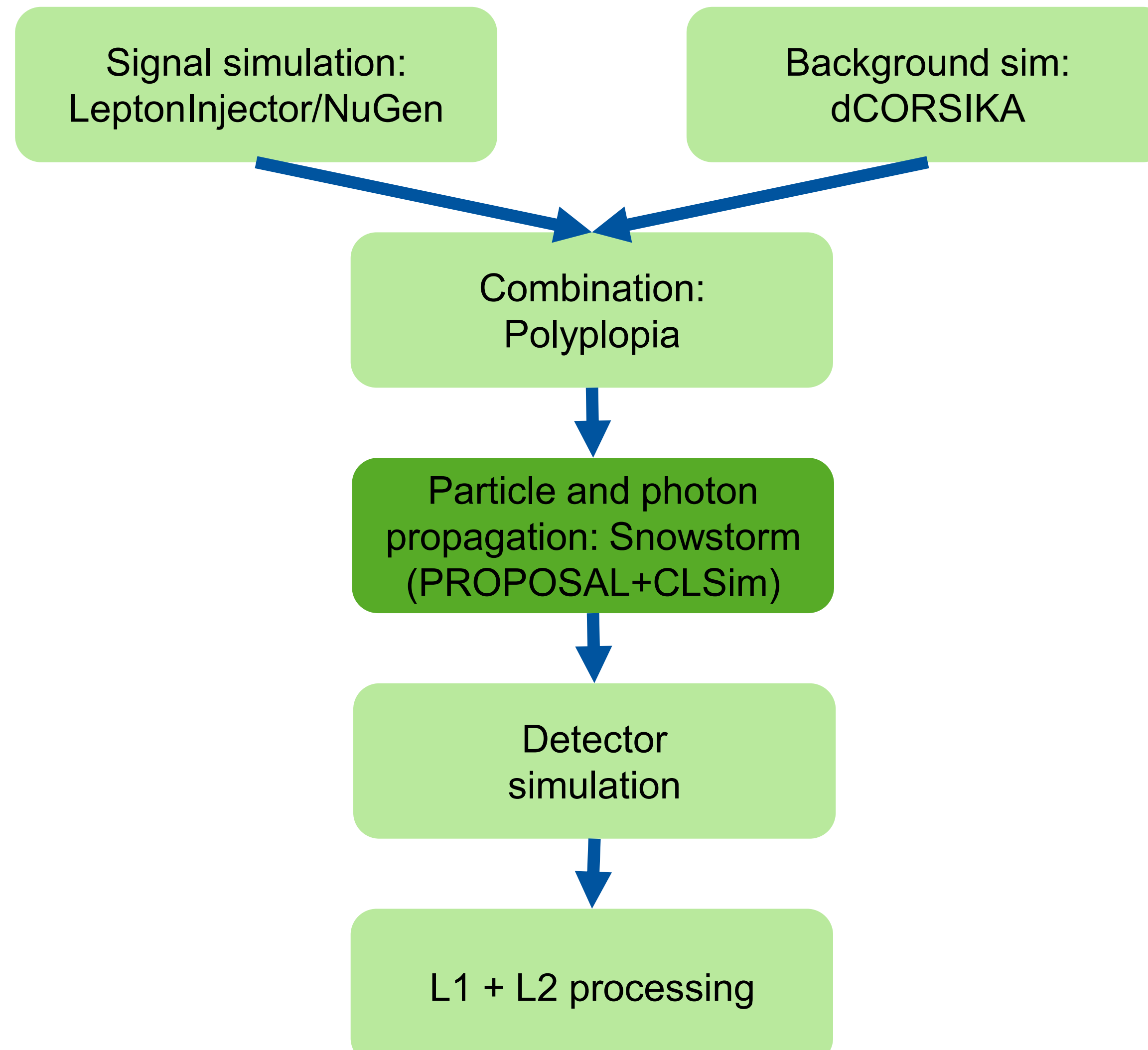
- Based on "standard" simulation chain
- Merge of signal+background I3MCTrees before any particle or photon propagation
  → Ensures that all particles get treated/propagated with the exact same parameters/settings further on

➢ Main SnowStorm simulation step:
  - ➢ Particle (muon) propagation with PROPOSAL
  - ➢ Photon propagation using CLSim

➢ Perturbing the ice model properties for chunks of frames using the *SnowStorm perturber*



SnowStorm short: Continuos variation of nuisance parameters (detector systematics) (blue) instead of discrete sets for specific values (red)



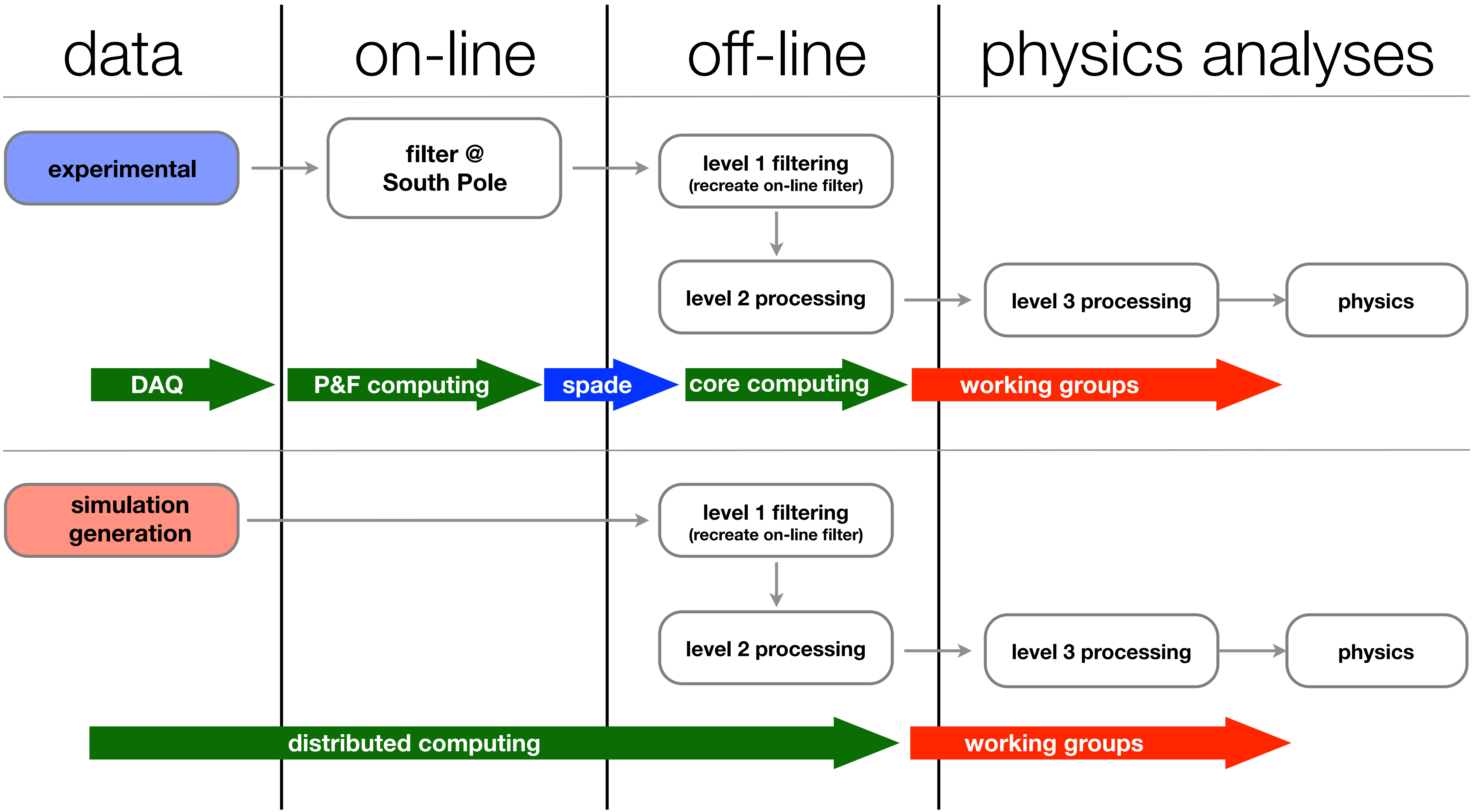
Simprod-Scripts

http://software.icecube.wisc.edu/documentation/projects/simprod_scripts/index.html

This project is a collection of scripts, tray segments and IceProd modules used in simulation production. The aim is to provide a central place with standard segments for running simulation in both production and privately.

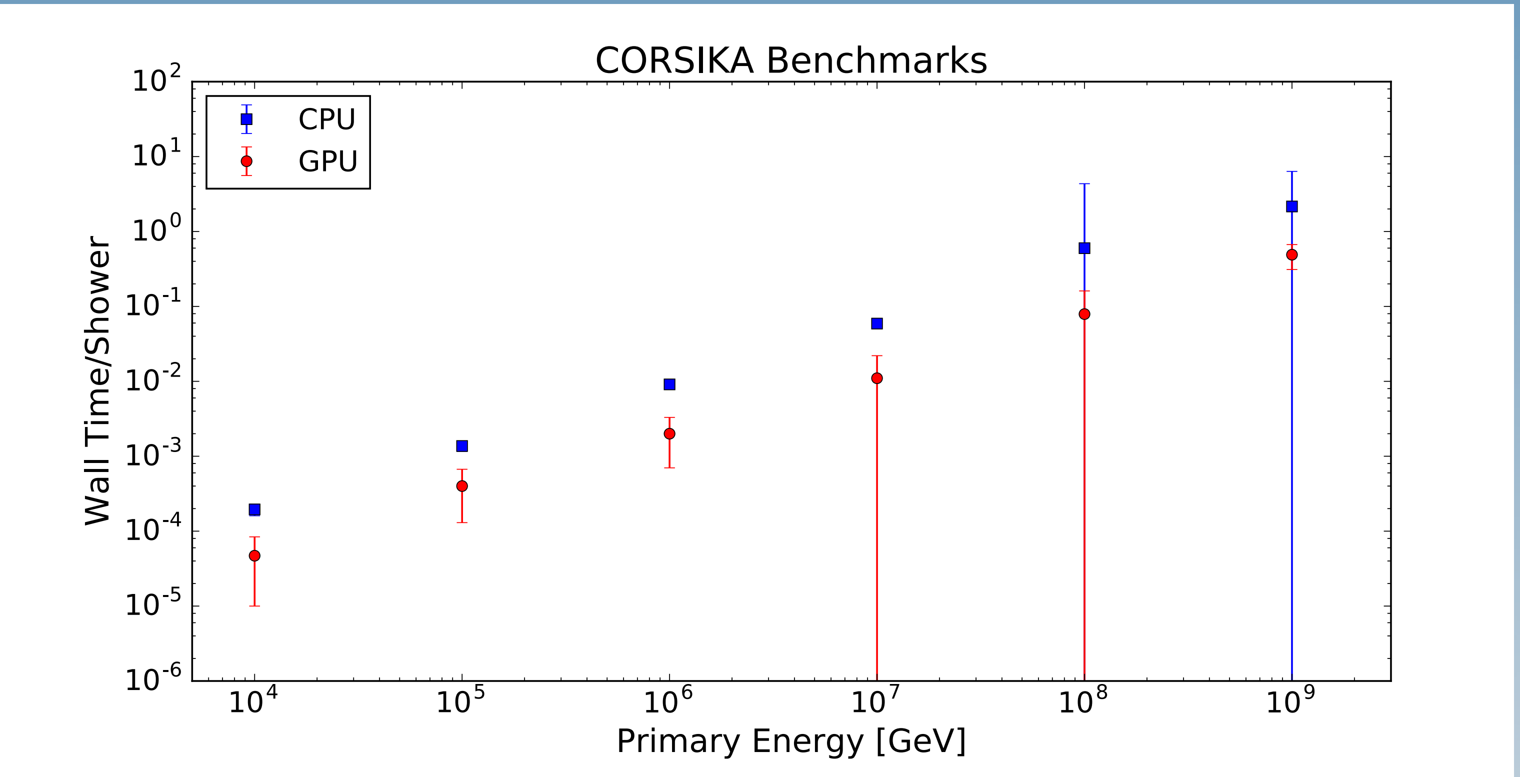
- **Tray Segments**: IceTray meta-modules that contain several I3Modules with default parameters.
- ~~**IceProd modules**: basic wrappers around tray segments that provide an interface for IceProd.~~
- **Scripts**: collection of python scripts used in simulation production
- **Examples**: The directory simprod-scripts/resources/examples contains a collection of example scripts for running IPModules
- **Tests:** are run on the build-bots to check that the different parts of the simulation are not broken with each commit to the software repository.

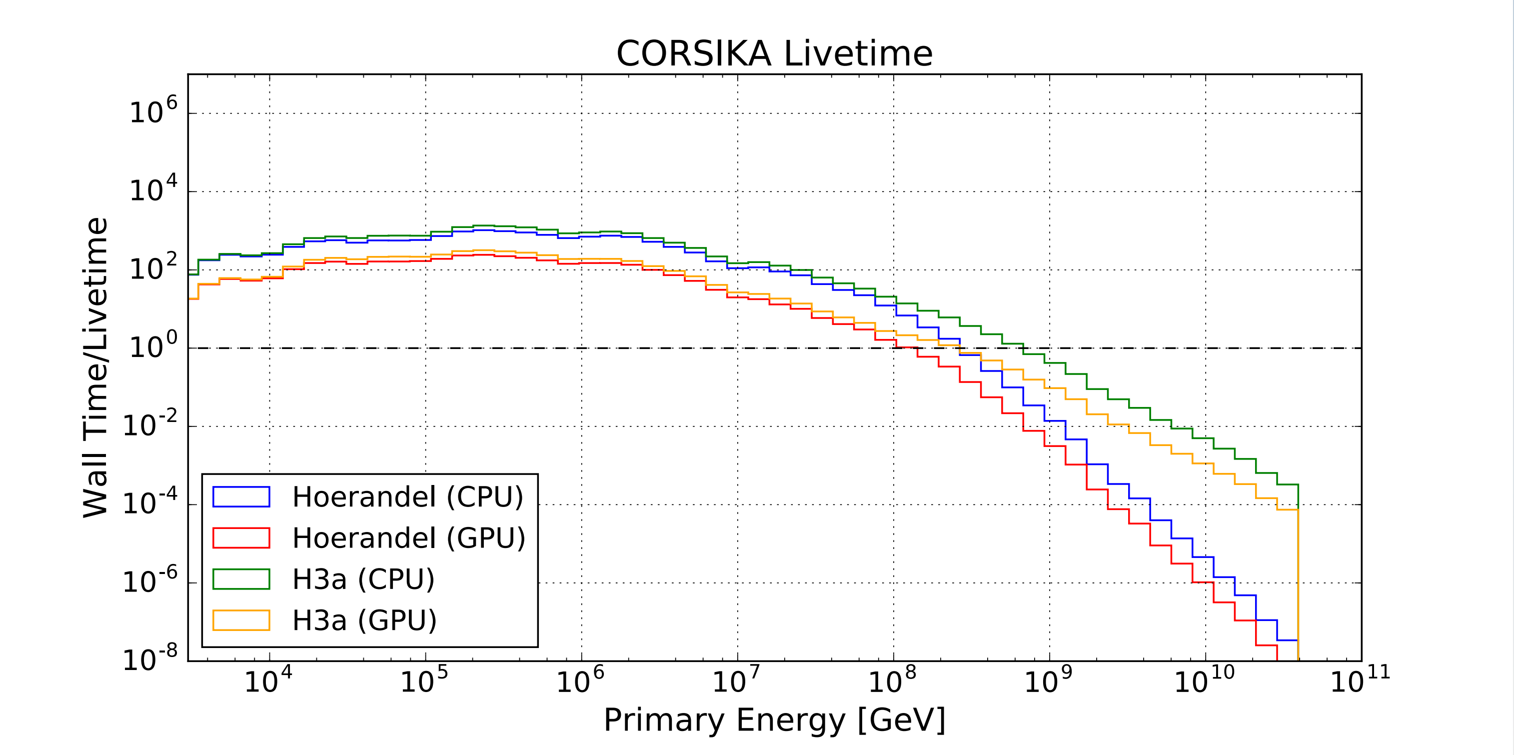
flow of experimental and simulation data



Computing Requirements

- Benchmark individual energy bins (E^-1 slope between bins).
- Differential CPU/GPU requirements per energy bin.
- Required number of CPU/GPU cores/units is determined by the area under the curve for a given energy interval





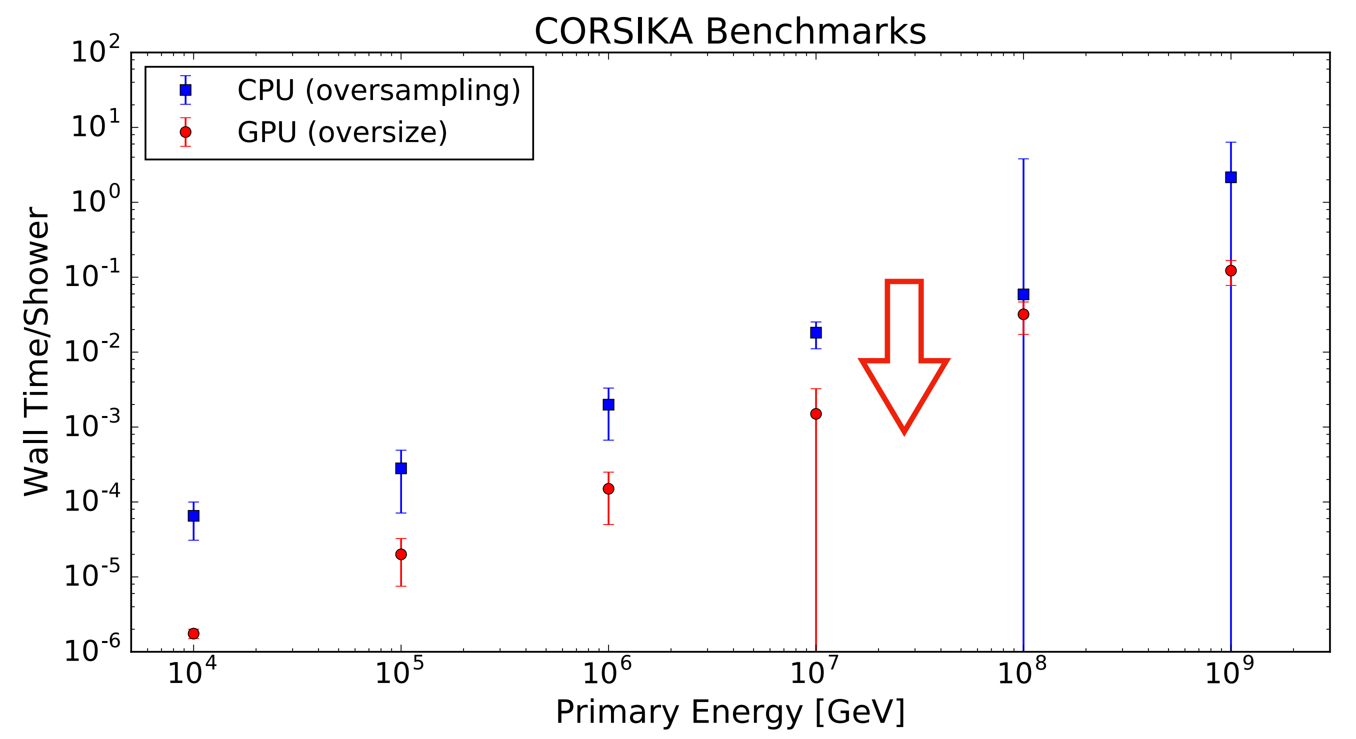
Computing Requirements

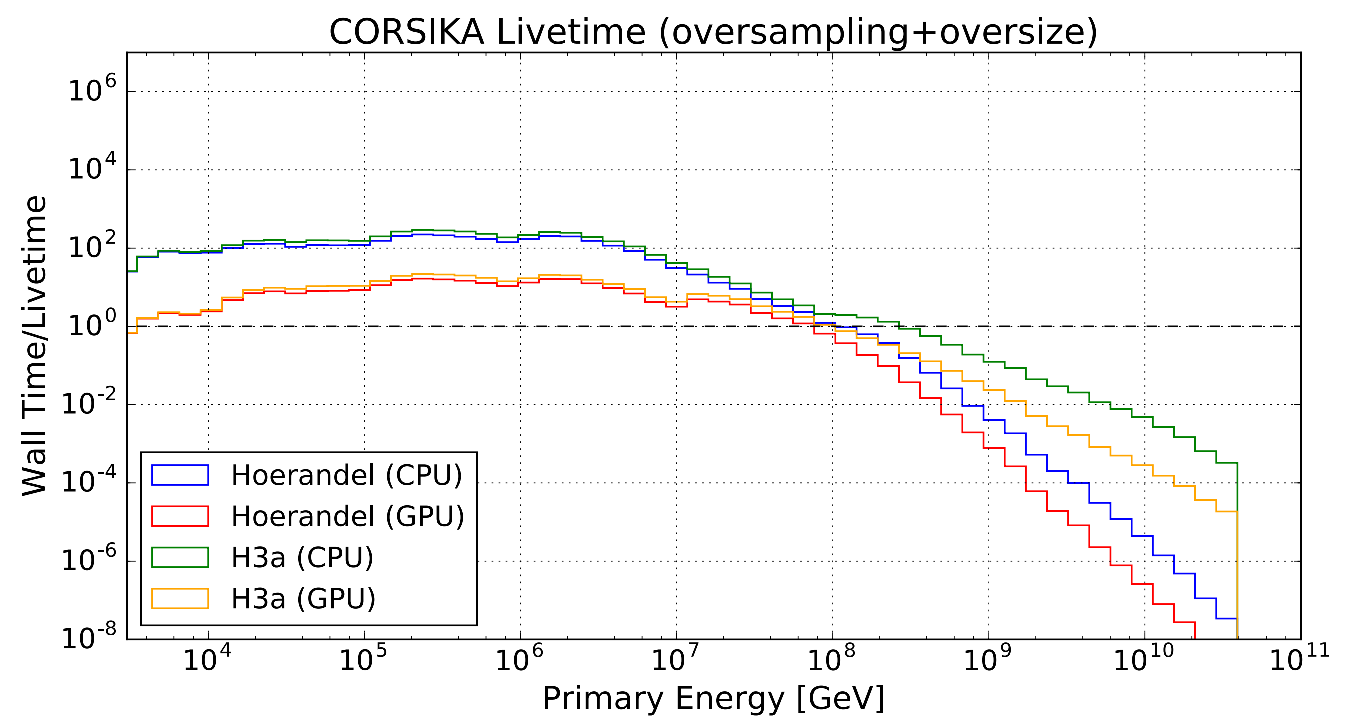
- DOM oversize 2->5
- Improvements in GPU efficiency have allowed for better utilization with larger DOM oversizing.

Fix in ClSim significantly improves GPU utilization in particular for low-energy CORSIKA.

Double buffering mode that reset number of parallel events to 1 after a flush and caused every other batch to send only 2 events to the GPU, leading to a dramatic drop in efficiency.

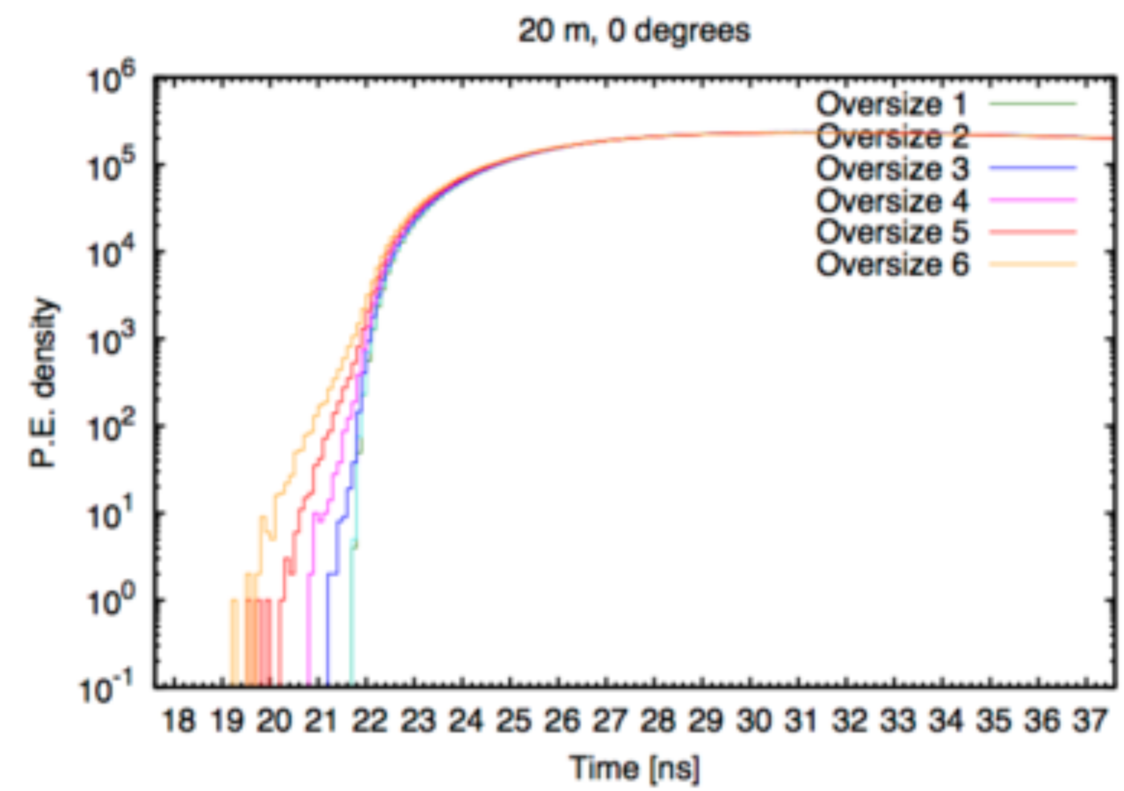
	Original	Optimization	Improvement Factor
CPU	14714	3516	4.18
GPU	4451	336	13.25

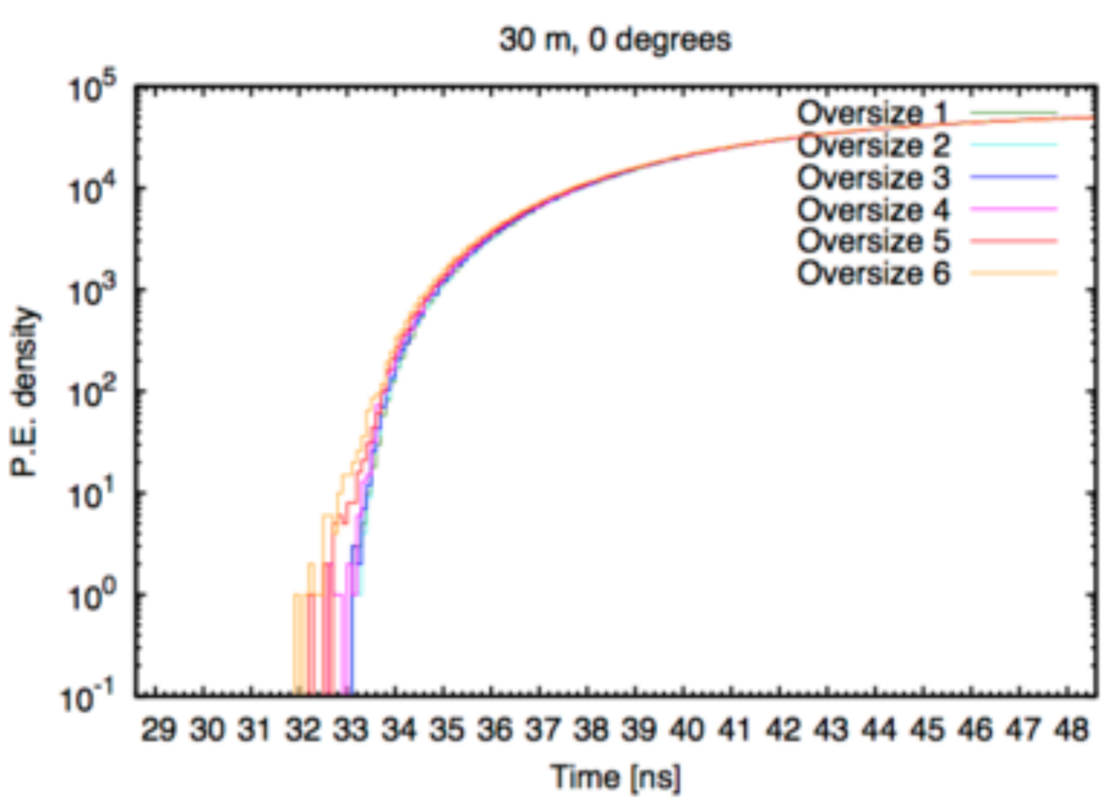


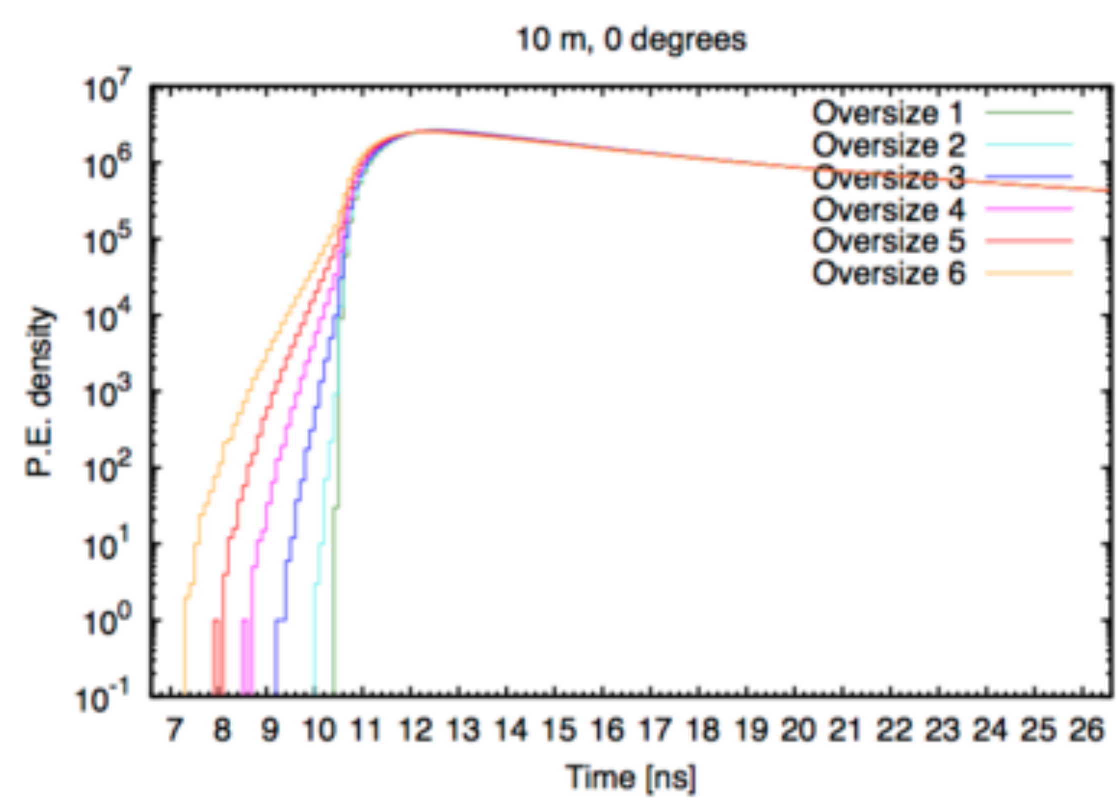


Energy dependence/Distance of source to DOM:

- In general, it's been determined that the problems derived from DOM oversizing are not dependent on energy but do get worst at smaller distances.
- Chris Weaver presented a study of the PE timing distributions for various DOM oversize factors at different distances.



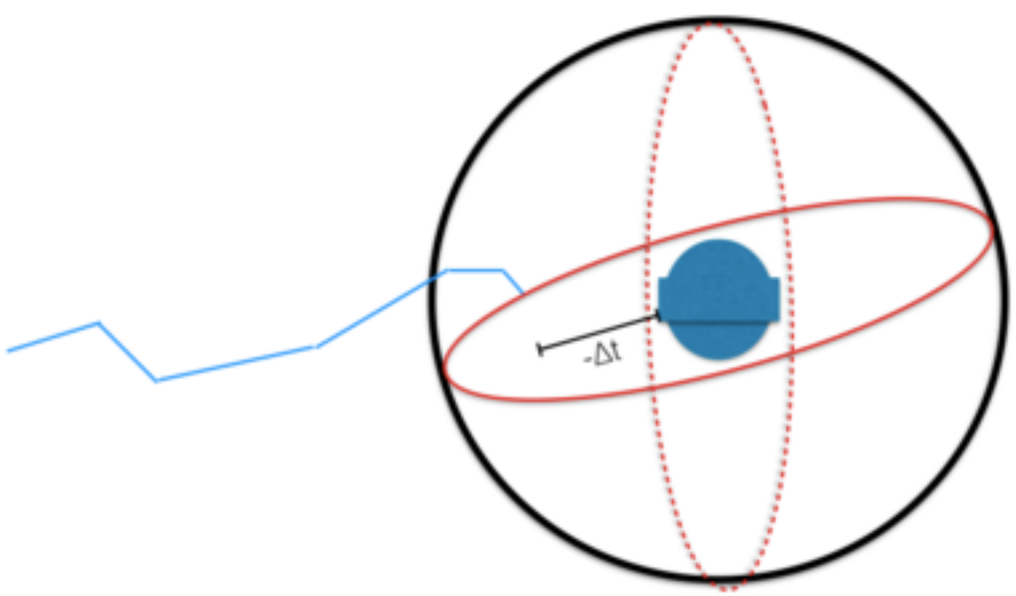




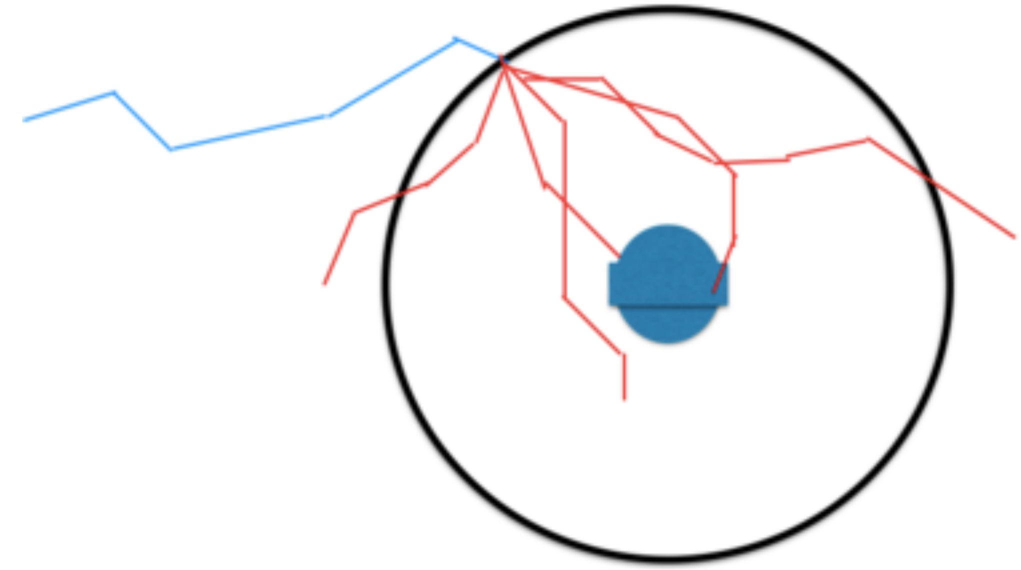
Photon Propagation

DOM oversizing in PPC/CISim

- An alternative approach was discussed during the workshop:
- Propagate photon "bundles" until they reach an oversize sphere at which point you branch and propagate each photon individually.
- This is in essence the opposite of the current oversizing but should also save on GPU computations since you only need to propagate a fraction of photons until they reach a distance R away from a DOM.



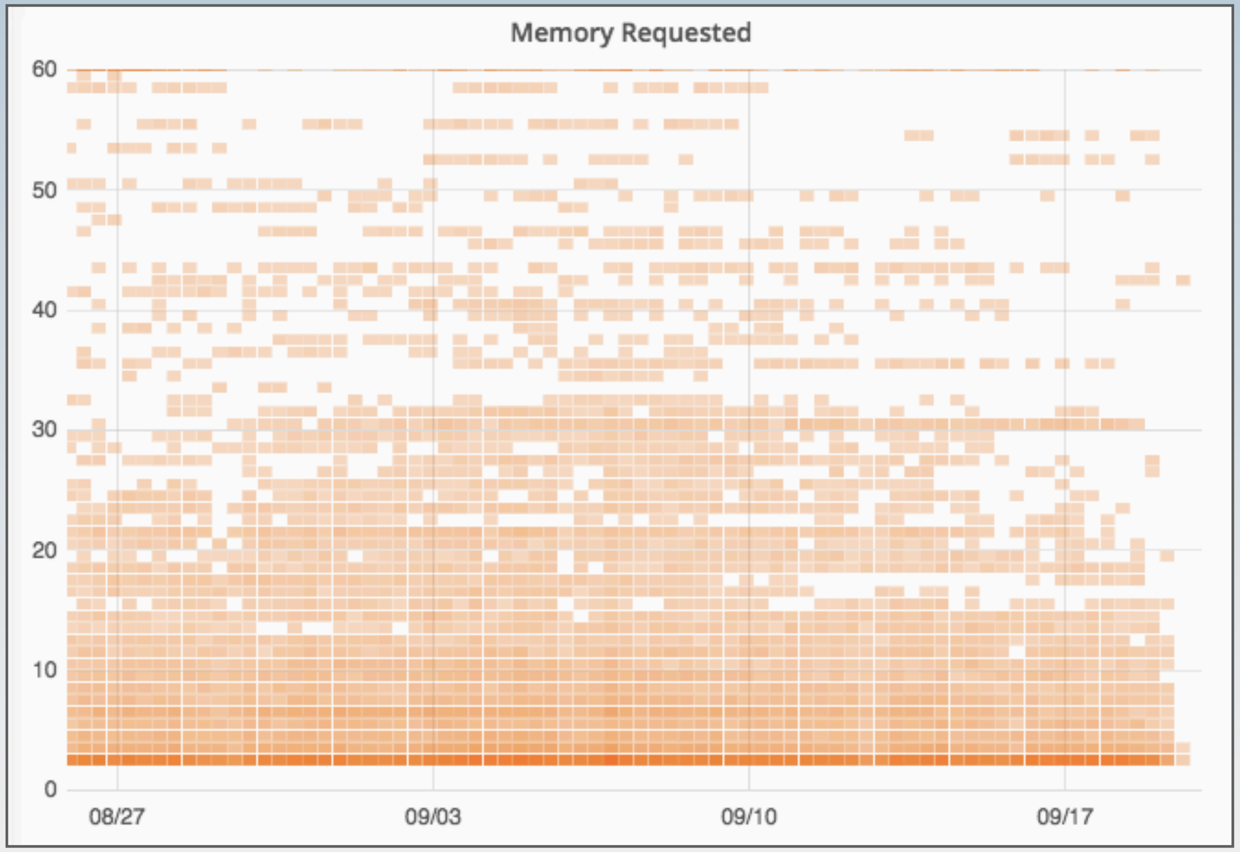
Current “pancake” DOM oversizing

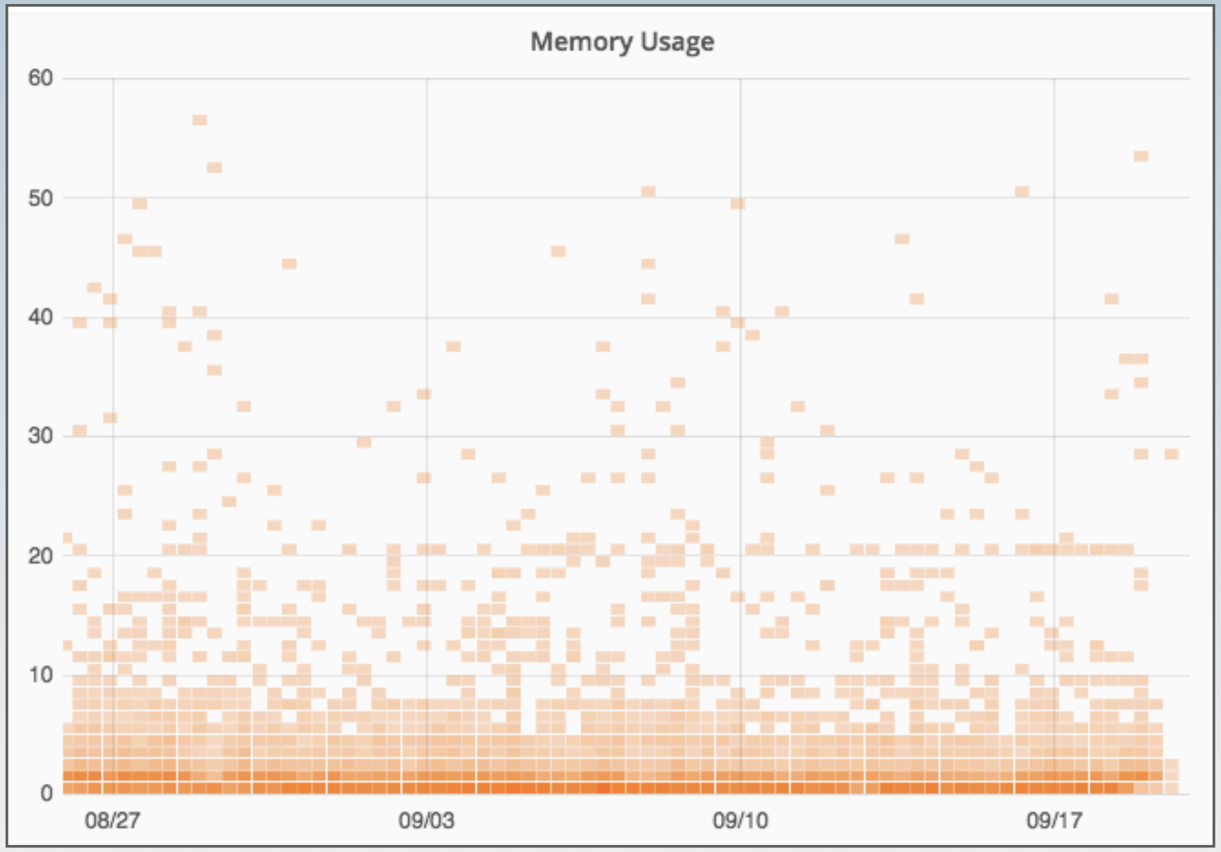


Proposed “bundle” oversizing

Resource Utilization

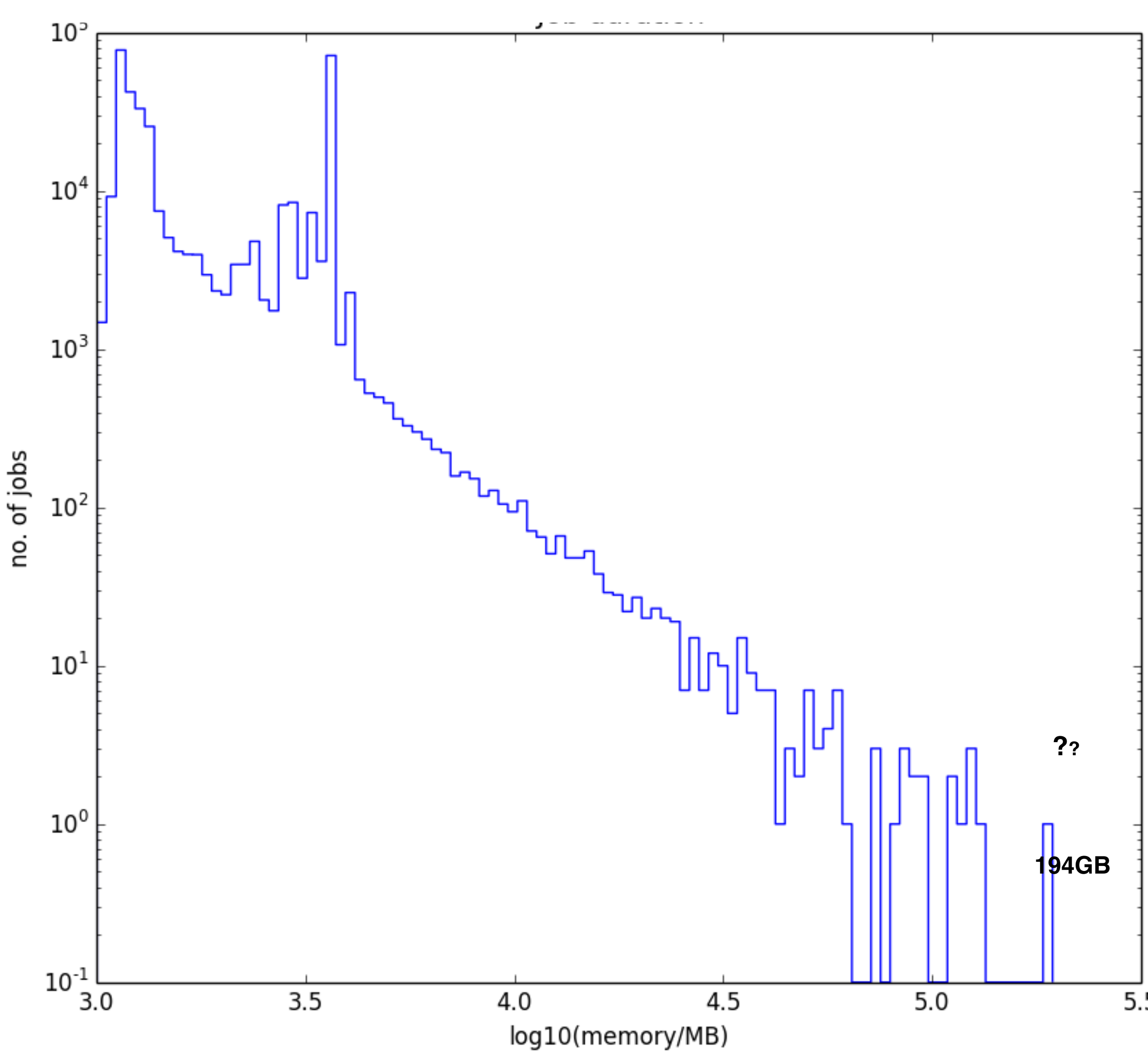
- Memory foot print of simulations has a large variance
- Some jobs require 60+ GB of RAM
- Typically, one or two high energy events (out of thousands) will be responsible for the large memory.



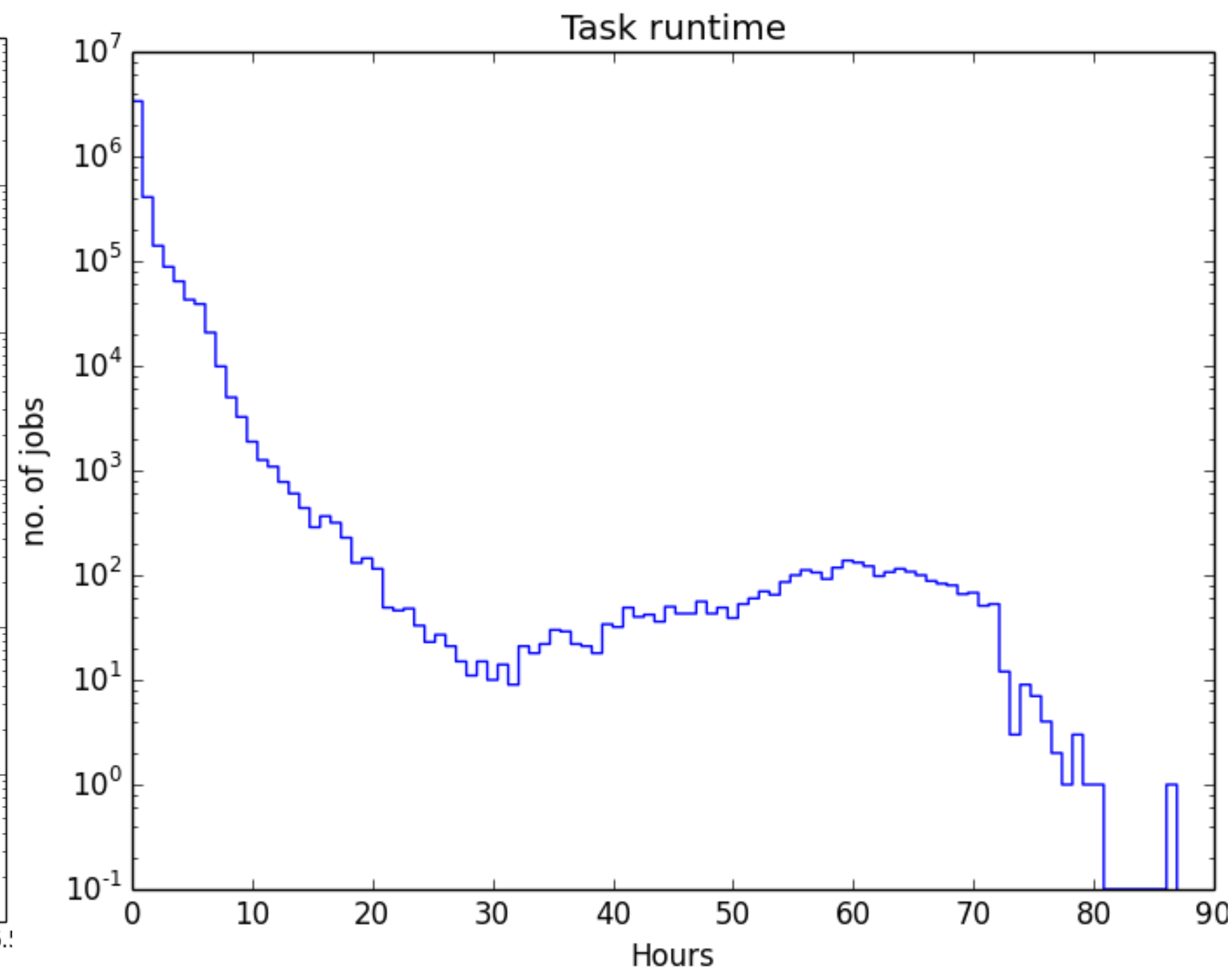


Resource Prediction/Optimization

Memory Usage



CPU



Simulation Production

- You will typically not be generating you own simulation.
- Simulating IceCube takes many computing cycles
- The collaboration utilizes distributed computing resources from around the world
- You can find information on generated datasets in
  - https://grid.icecube.wisc.edu/simulation/DashBoard/
- Simulations are stored in “Data Warehouse”
  - In-ince: /data/sim/IceCube/[YEAR]
  - IceTop: /data/sim/IceTop/[YEAR]