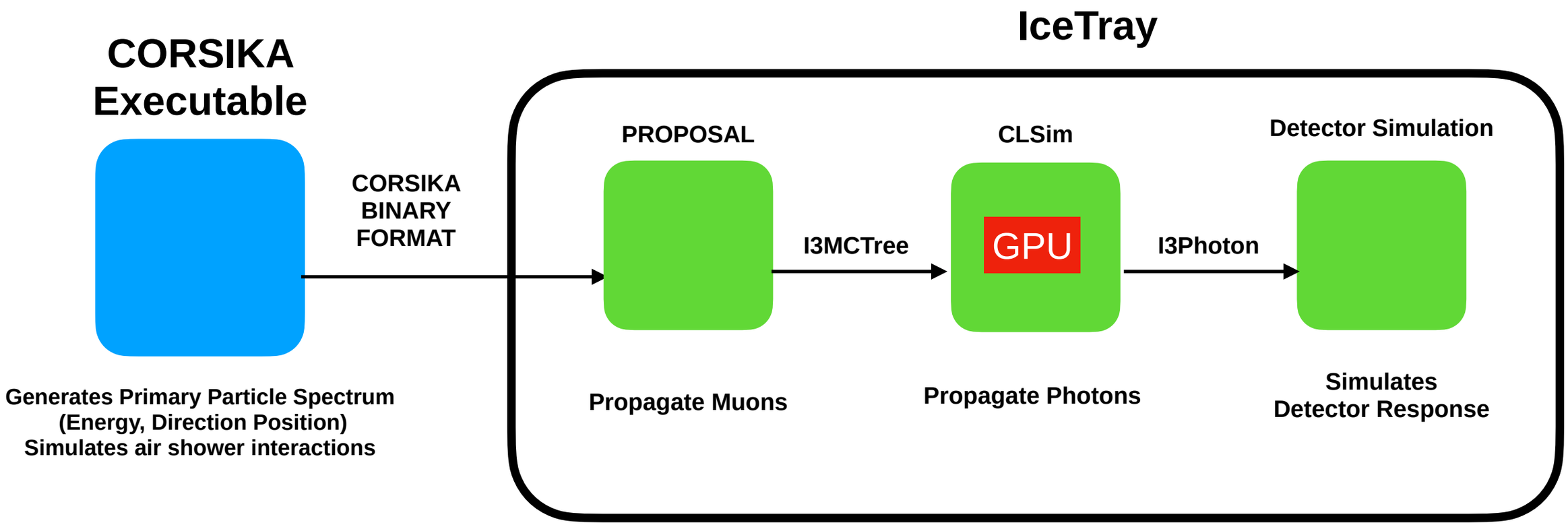
Triggered CORSIKA
And Multiprocess Server

Kevin Meagher
IceCube Simulation Workshop
October 18, 2021

How CORSIKA used to work



I3Photon = Lots of Memory
I3MCPE = Uses less memory

CORSIKA files are generated by a separate CORSIKA binary
IceTray then processes in a linear fashion by each module
First Muons are propagated by by PROPOSAL,
Then Photons are processed by CLSim using the GPU
The entire I3Photon sequence is stored in memory before being converted to the binned I3MCPE format

Why is CORSIKA so hard to produce?

Low Energy:

- CORSIKA shower files are created separately and transferred at the start of the job which saturates IO

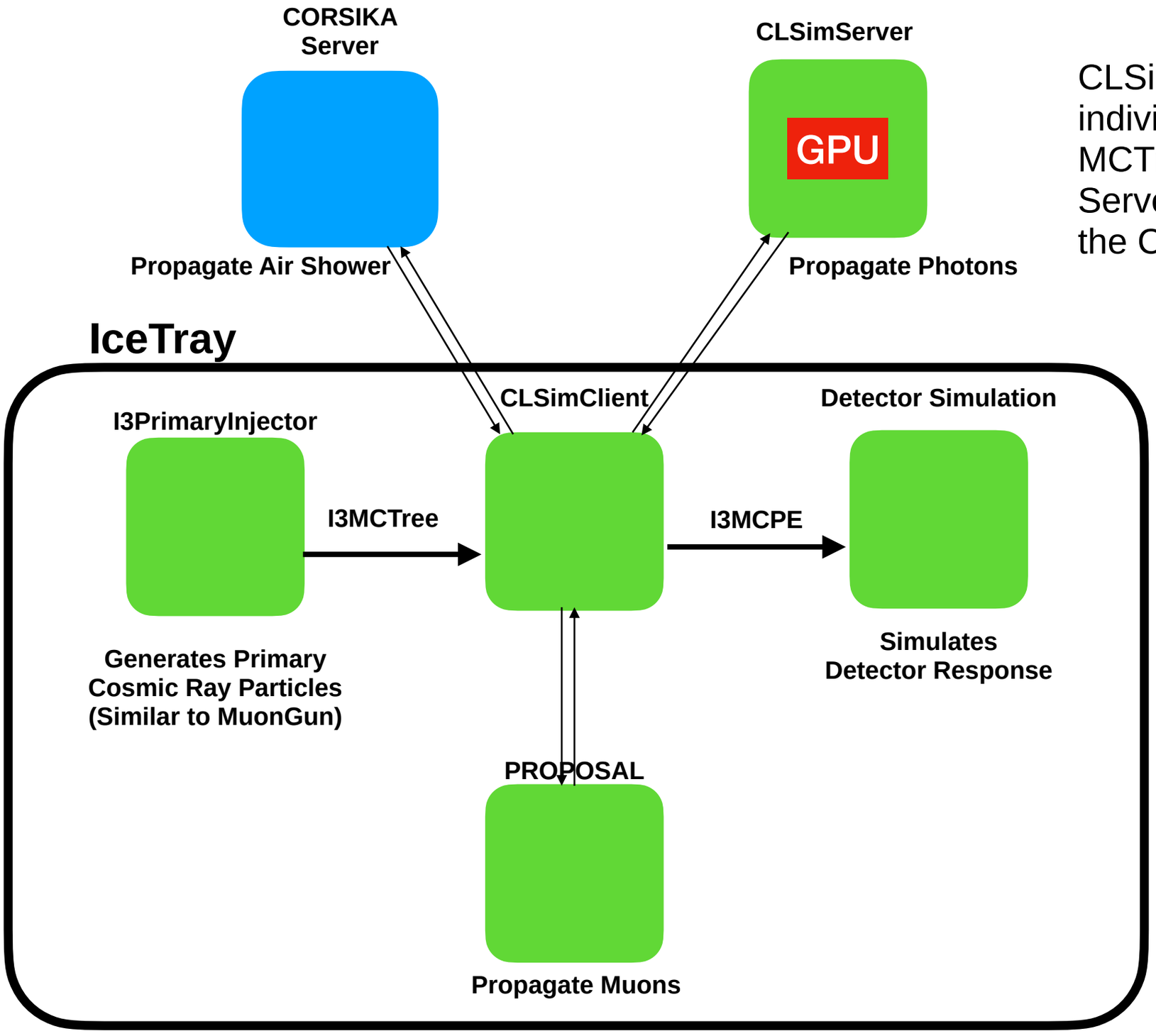
Medium Energy:

- CORSKIA showers need more CPU than GPU which is an inefficient use of our cluster resources (CPUs sit idle while waiting for GPUs to finish)

High Energy:

- Large showers push the memory limits on nodes. CLSim needs to store the entire event (both the MCTree and I3Photons) in memory. Power-law statistics require that we have to allocate memory for very rare events.

Triggered CORSIKA and Multiprocess Server CLSim



CLSimClient passes individual particles from the MCTree to the CORSIKA Server, to PROPOSAL to the CLSimServer

I3MCPE are created directly from the output of each individual CLSim propagation Saving memory

What do we gain from this?

No Need to break up CORSIKA jobs by energy:

- Low Energy:
  - No need to generate CORSIKA files separately (prevents IO bottleneck)
- Medium Energy:
  - Multiple instances of IceTray can share a CLSimServer resulting in better CPU utilization
- High Energy:
  - Individual particles are passed from CORSIKA to PROPOSAL to CLSim and binned I3MCPE are made for each particle from CLSim rather than the entire event. This significantly reduces the memory footprint
  - Multiple instance of IceTray can run in the same cluster job

Event More Benefits: Oversampling and Other Tricks

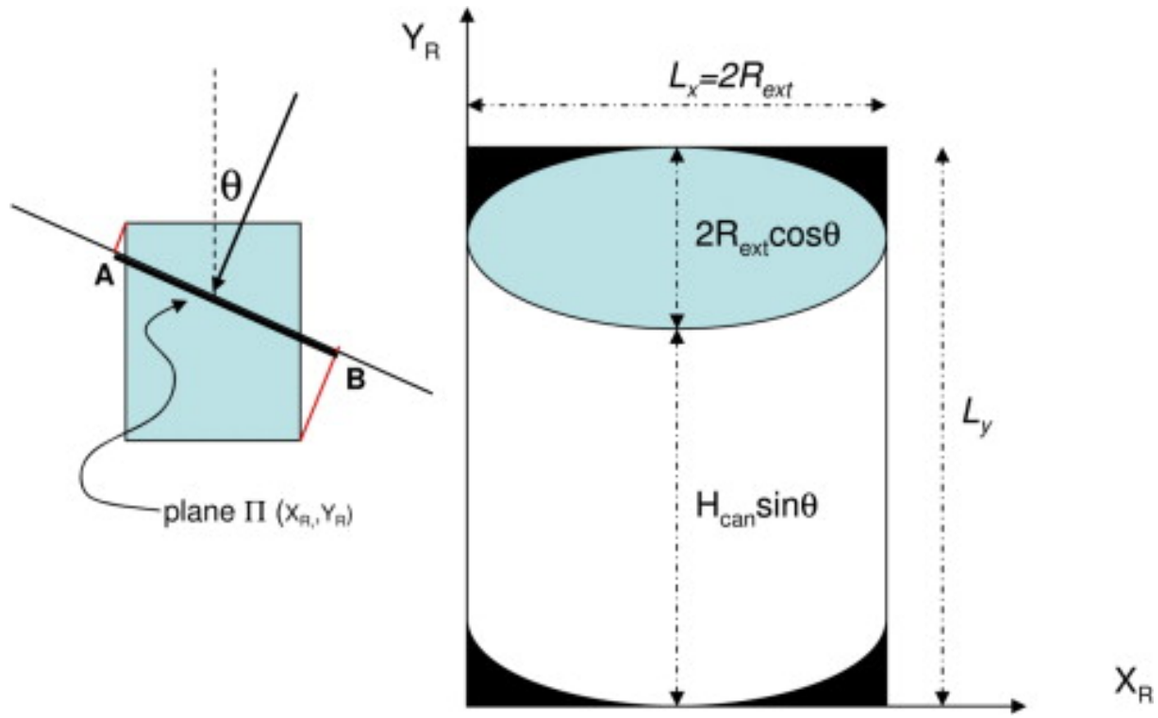
- Cosmic ray air shower primary can be generated according to arbitrary spectral and spatial distributions
  - Generate primaries directly on the detector cylinder before cosmic ray propagation (similar to MuonGun)
- Different CORSIKA configuration cards can be sent to different events
  - Set muon energy threshold based on the inclination of the shower
- CORSIKA propagation of a shower can be under-sampled based on shower development
  - Kill events with low leading energy muon

Other ideas which have been proposed to bias simulation

- Oversample coincident events: eg 1 for coincident showers, 10^{-4} for single showers
- Only populate EM component if the shower hits IceTop.
- Oversample IceCube muon “Lanes”
- Tau analysis: oversample charmed mesons
- ESTES: oversample veto hotspots
- MESE/cascades: oversample based on expected charge in the veto region
- Cosmic rays: oversample high P_T muons
- Force neutrino interaction in air shower

I3PrimaryInjector Module

- Utilizes S-Frame object to keep track of generation surface
- Uses `SampleImpactRay()` to sample on the surface of the cylinder
- Samples the energy from a different power-law for each primary type
- Creates I3MCTree with primary
- Creates an I3ShowerBiasMap and puts it in the frame (so CORSIKA knows how to bias showers)



server_sim.py

One Script to run the entire shish kabob is in simprod-scripts

- Creates I3CLSimServer
- Runs tray with:
  - I3PrimaryInjector
  - PolyplopiaSegment
  - I3CLSimClientModule with the following propagators:
    - CorsikaService as a CosmicEventGenerator
    - PROPOSAL and I3CMC as propagators
    - I3CLSimLightSourceToStepConverterAsync

Facilities to run multiple trays which connect to the same I3CLSimServer existed in the past and will be reenabled soon

IceProd

Datasets Profile Logout

Submit New Dataset

Submit Dataset like Current

Dataset 21889 Details

View Config Edit Config

Settings

description:	ME IC86.2016 Triggered CORSIKA-in-ice 5-component model Sibyll2.3c (CORSIKA 77401) with weighted spectrum of E^-2.6, using Spice3.2 ClSim. Angular range of 0deg < theta < 89.99deg and energy range of 3e4GeV < Eprim < 1e6GeV. DOM oversize = 5
jobs_submitted:	100000
tasks_submitted:	300000
tasks_per_job:	3
group:	simprod
dataset_id:	5deb19300c1411eca9f2141877284d92
dataset:	21889
status:	processing
start_date:	2021-09-02T17:36:55.621073
username:	kmeagher
priority:	0.25 Update
debug:	False
jobs_immutable:	False

Reset Hard Reset Suspend Truncate Delete Logs

Jobs

processing 1201
errors 378
complete 69159

Tasks

waiting 1346
processing 774
reset 11
suspended 633
complete 209450

Task Status by Task Name

Name	Type	Waiting	Queued	Running	Complete	Error
server	GPU	0	0	324	70288	0
filtering	CPU	530	0	75	70003	1
L1L2	CPU	816	0	375	69159	10

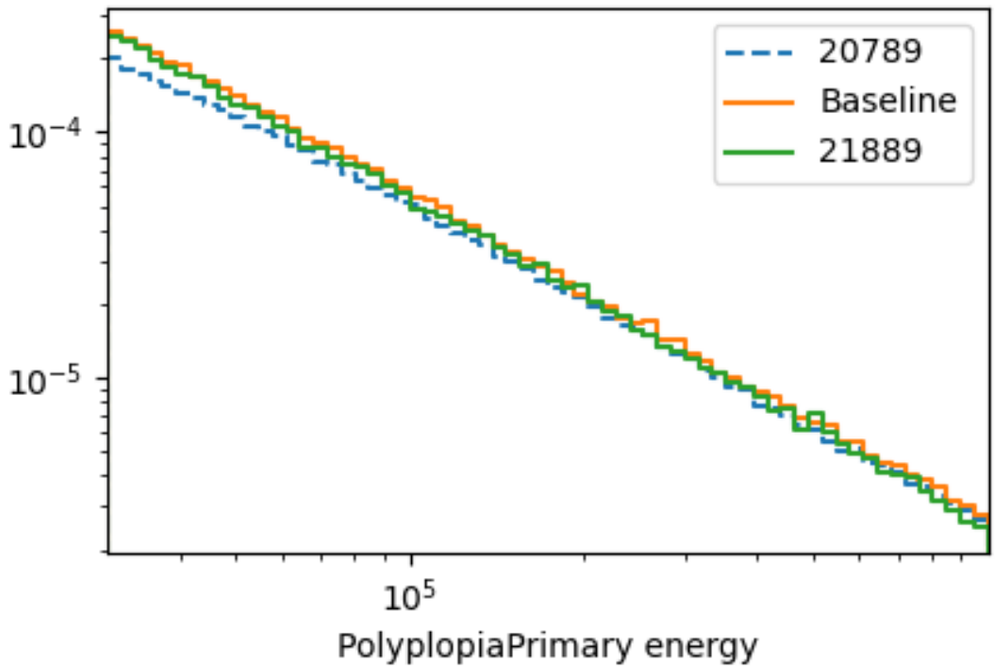
Completion Statistics

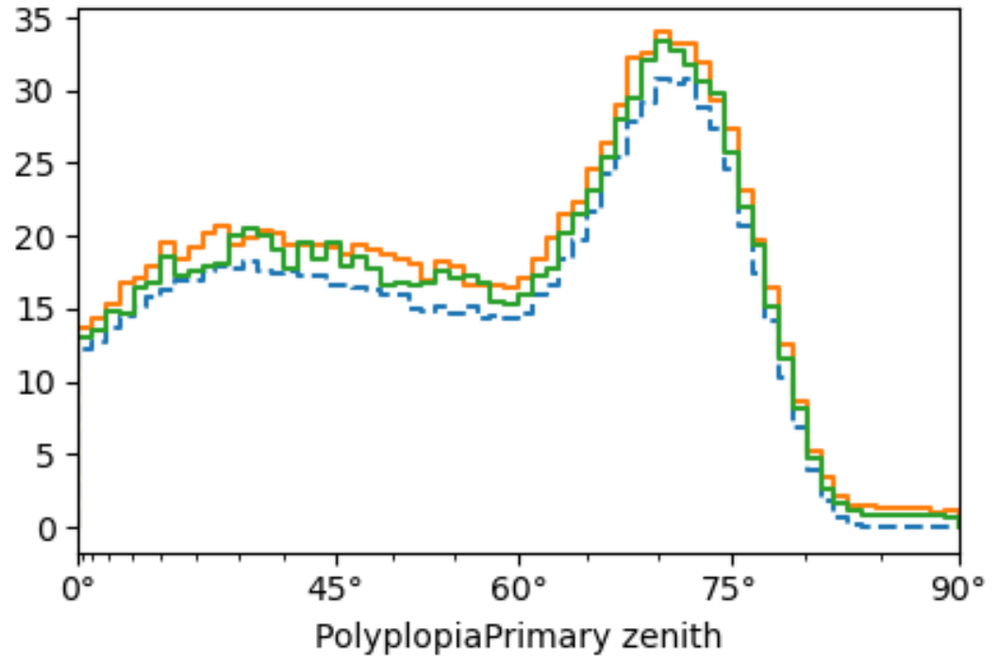
Name	Avg/stddev (hours)	Max/min (hours)	Eff
server	0.29 / 0.07	1.83 / 0.17	89%
filtering	3.81 / 1.42	45.14 / 1.58	99%
L1L2	2.99 / 0.97	29.51 / 1.65	99%

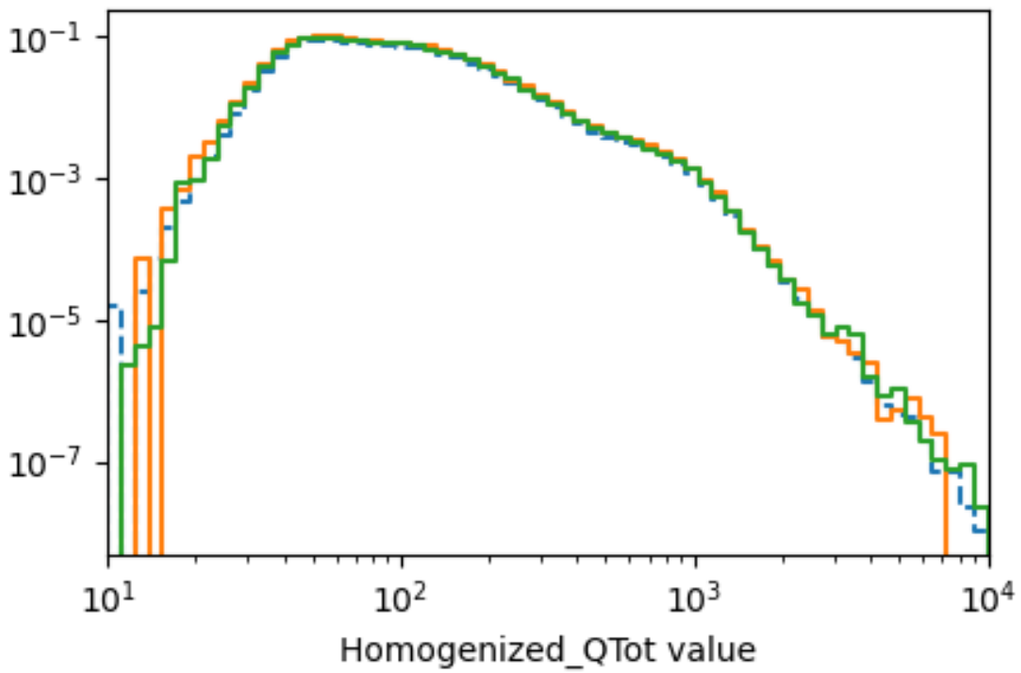
IceProd 2.5.7

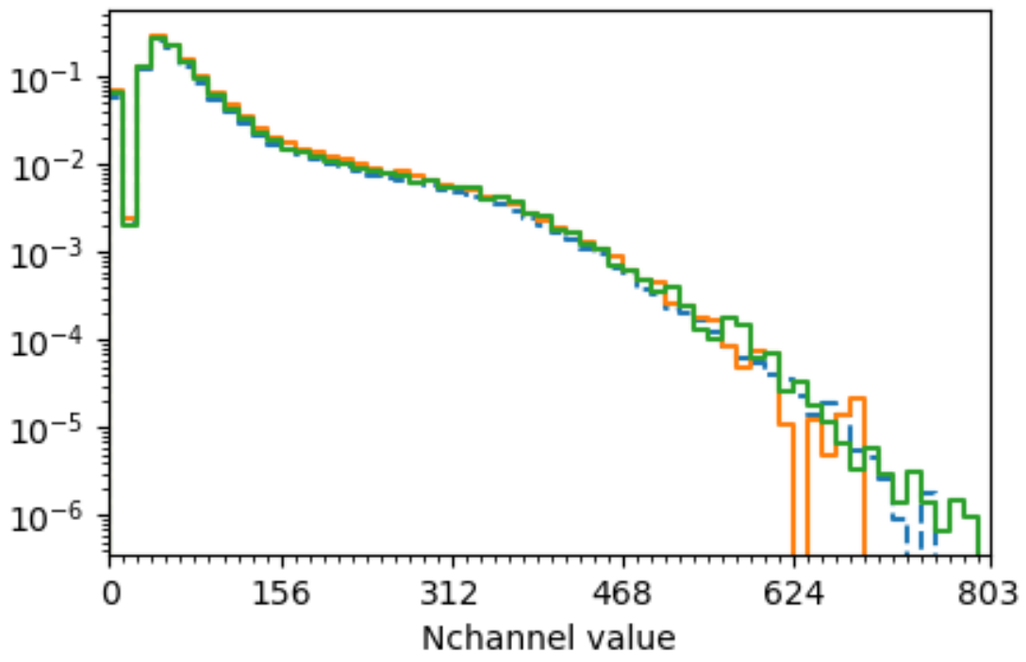
Dataset 21889

- First large scale dataset using Triggered CORSIKA
- Primary energy: 30 TeV to 1 EeV
- 100,000 jobs each with 300,000 showers
- Muon bias of 10^{-3}
- ~70% complete

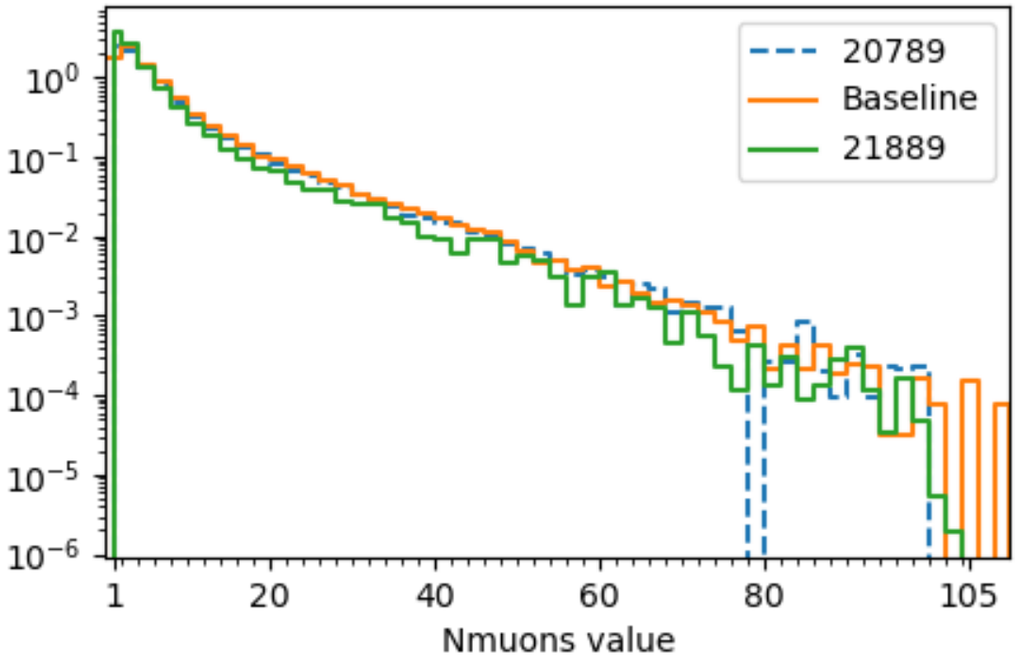


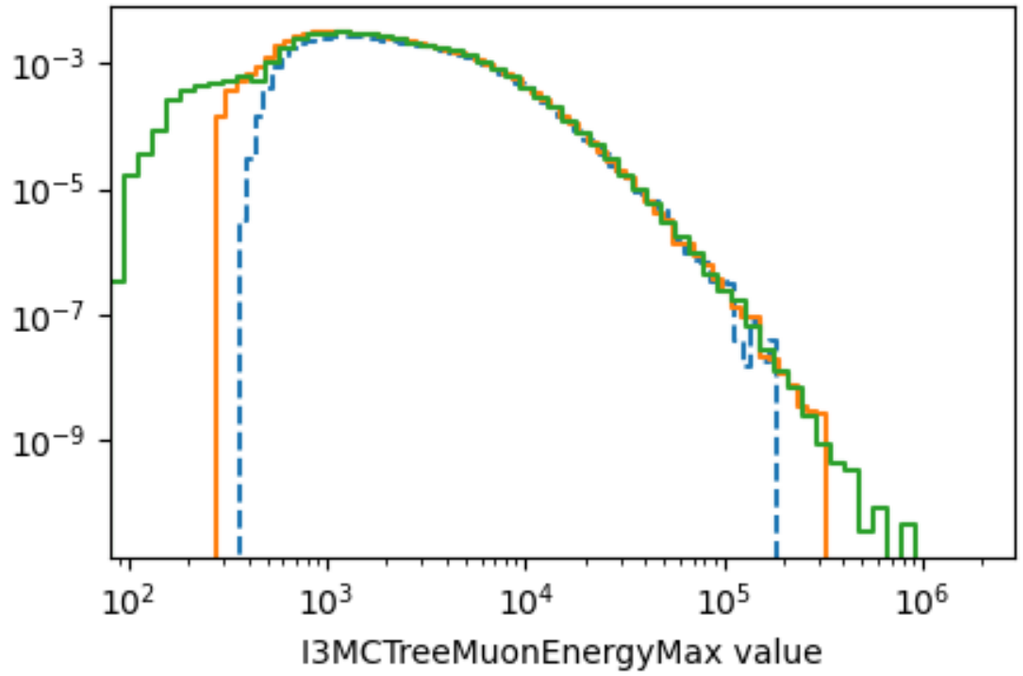


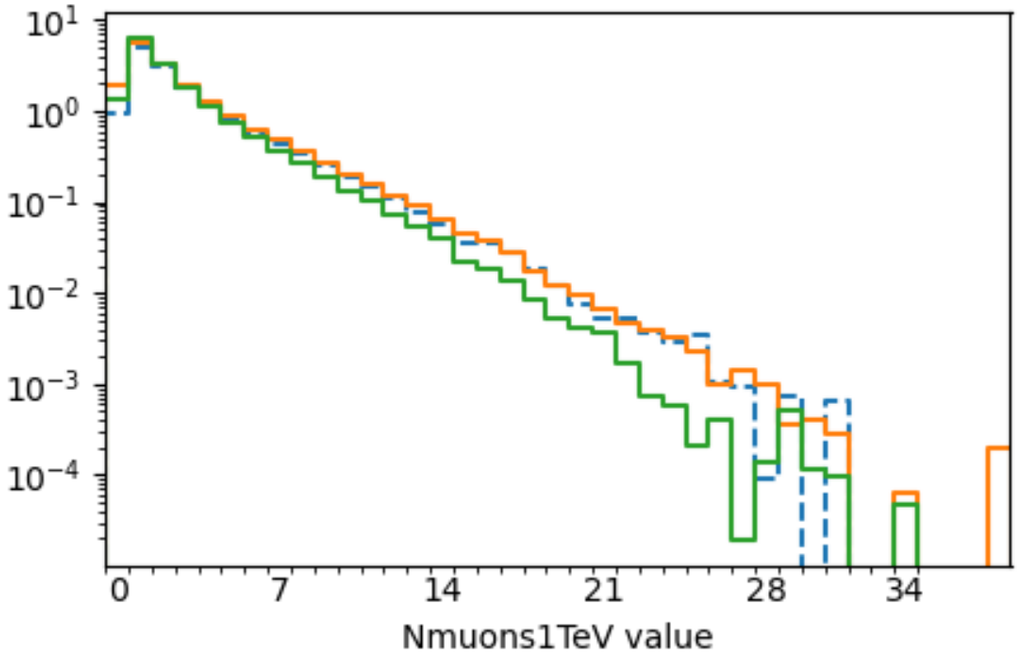


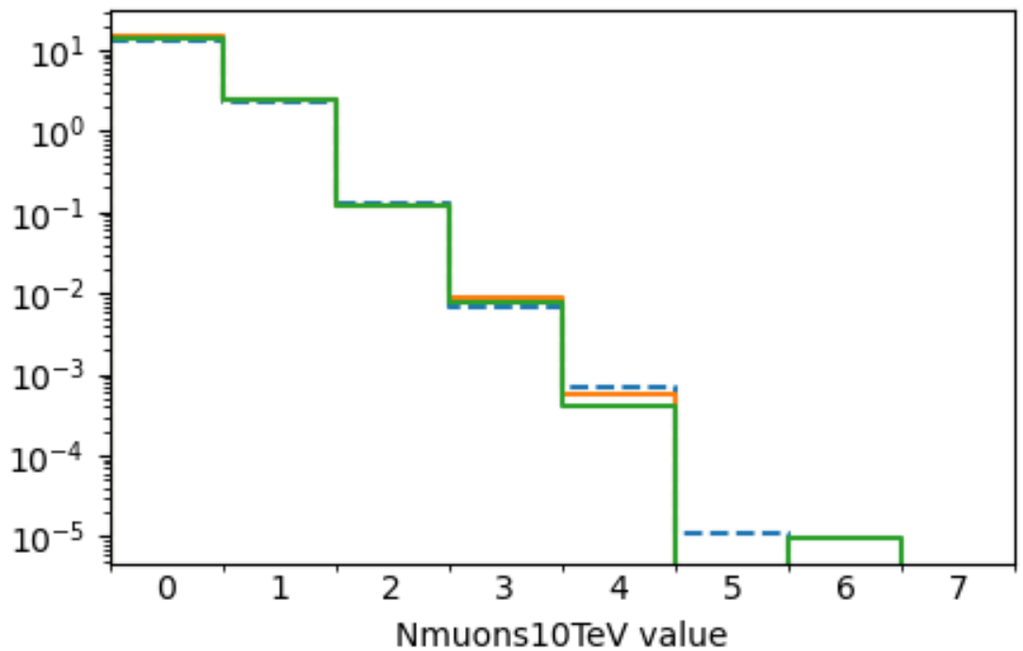


- 20789: Conventional CORSIKA dataset from 2016
- Baseline: Latest release run with conventional CORSIKA script
- 21889: New triggered CORSIKA

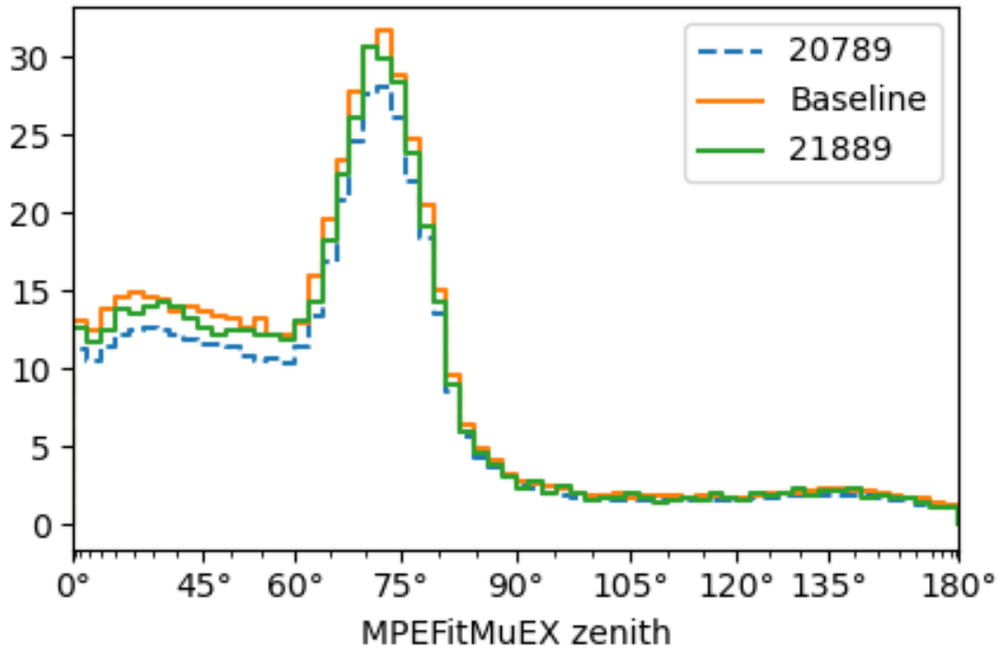


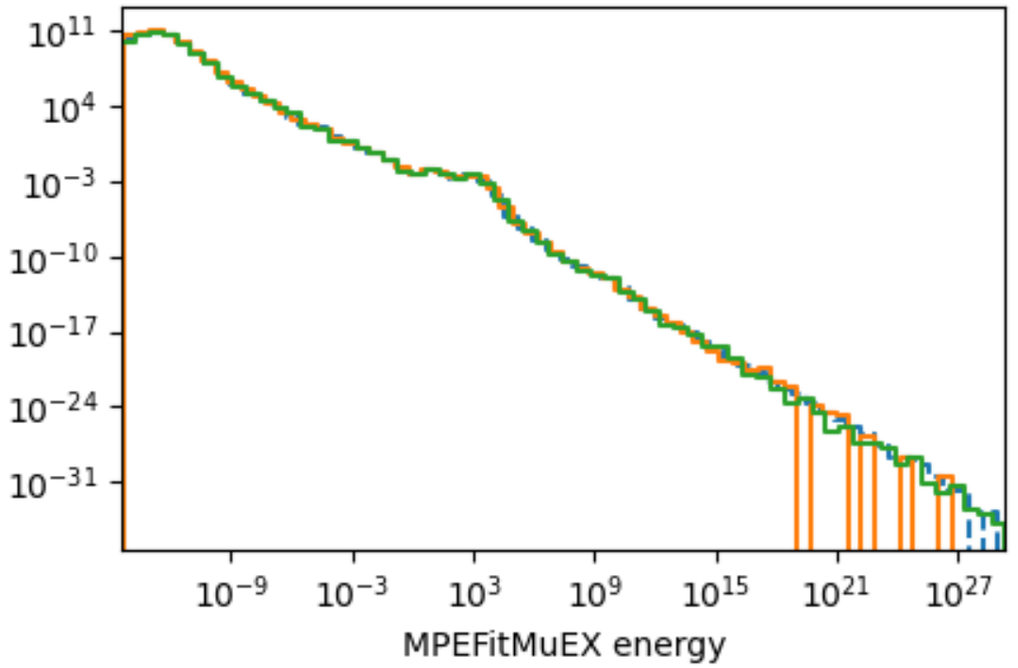


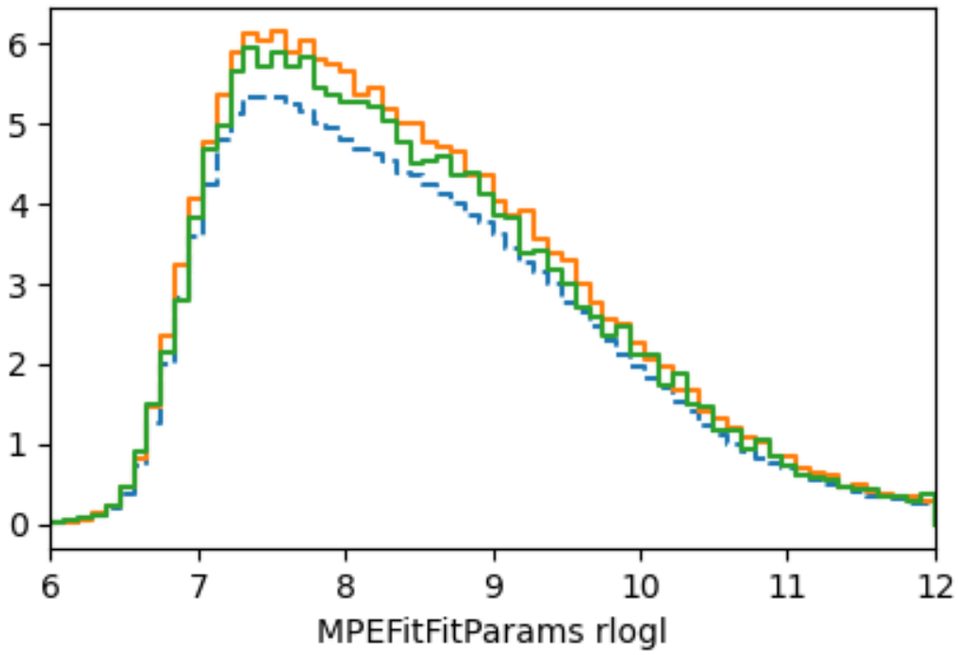


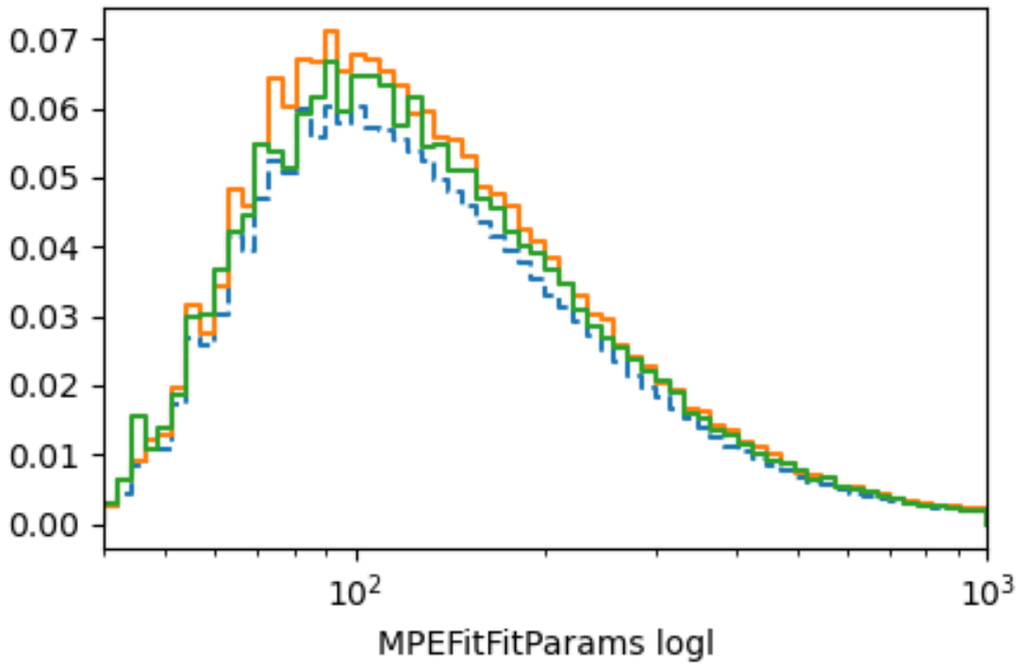


- 20789: Conventional CORSIKA dataset from 2016
- Baseline: Latest release run with conventional CORSIKA script
- 21889: New triggered CORSIKA

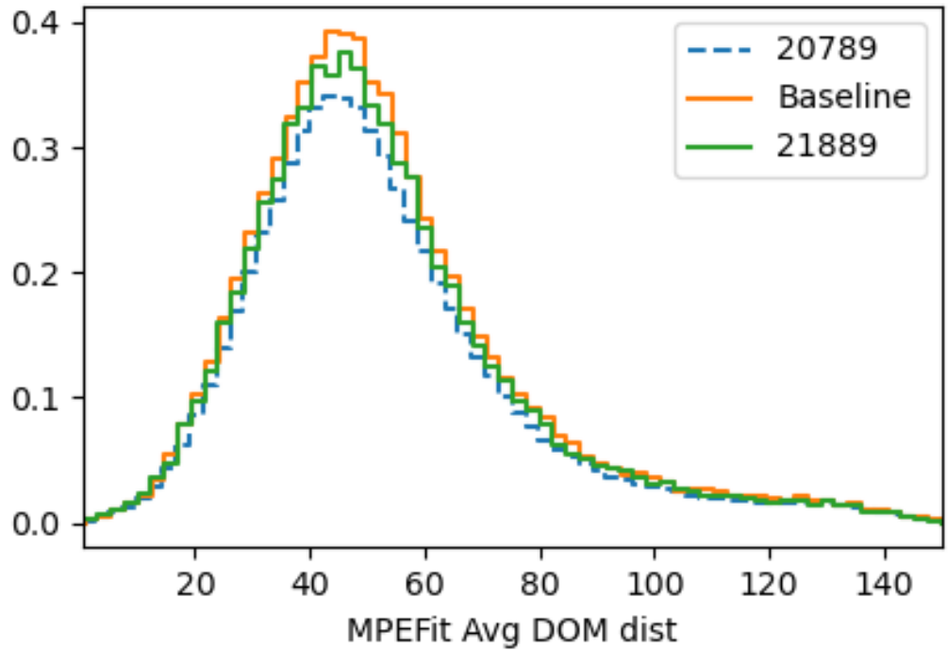


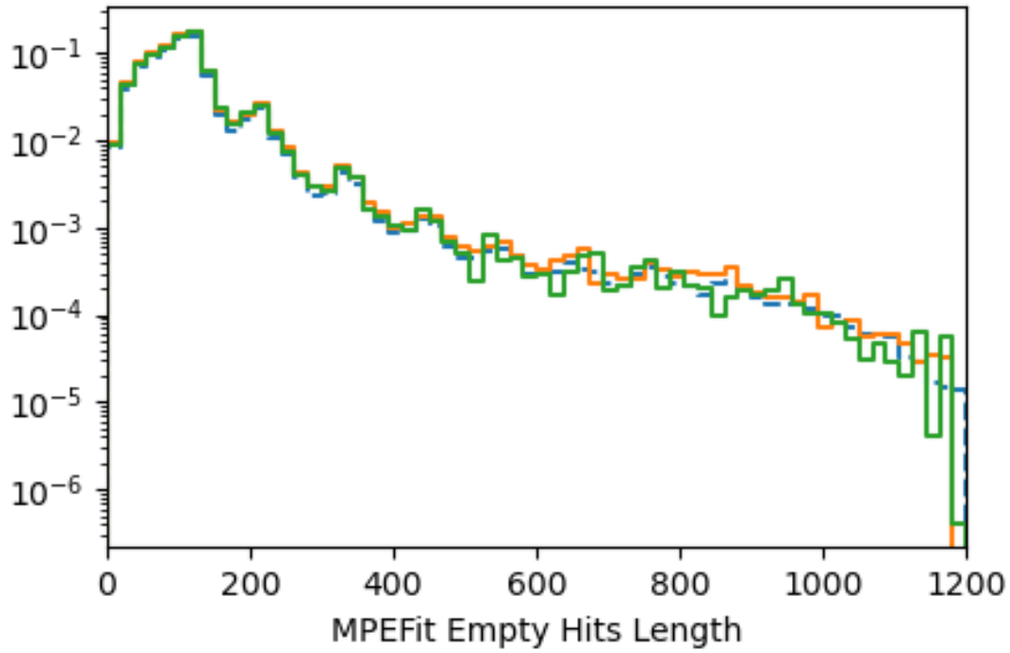


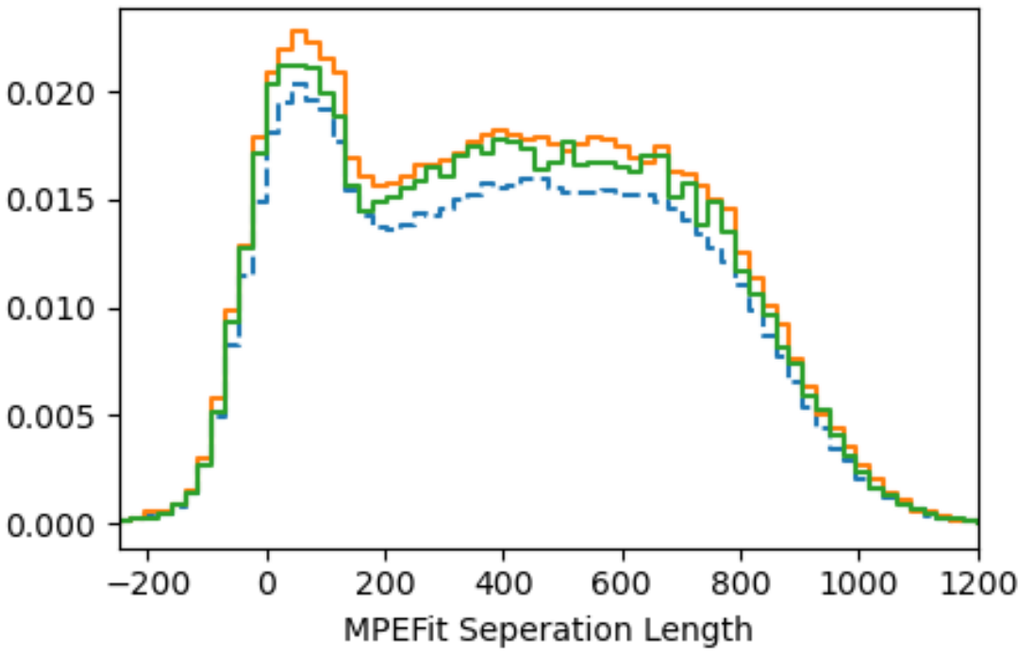


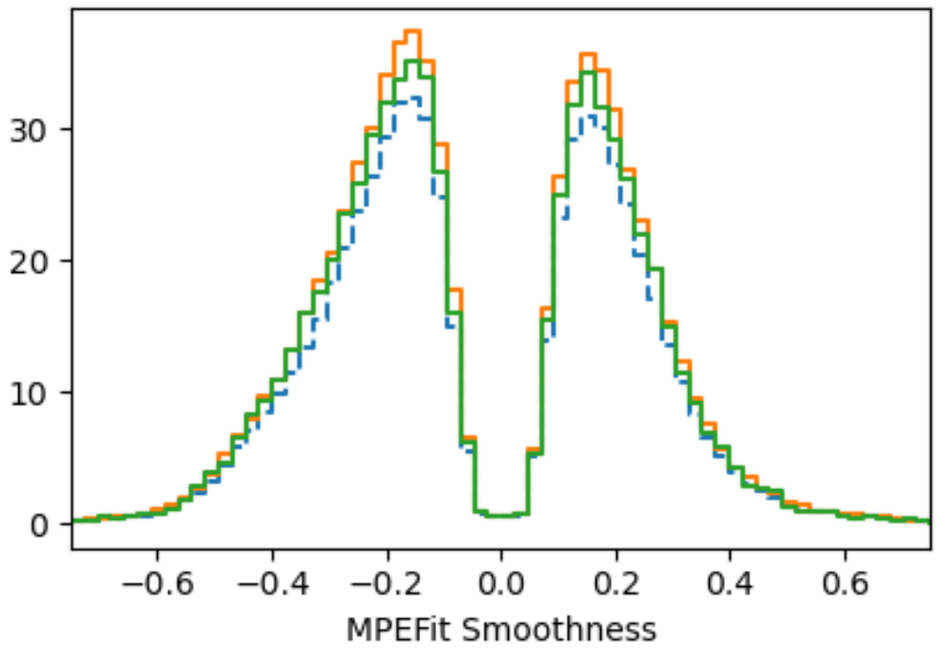


- 20789: Conventional CORSIKA dataset from 2016
- Baseline: Latest release run with conventional CORSIKA script
- 21889: New triggered CORSIKA





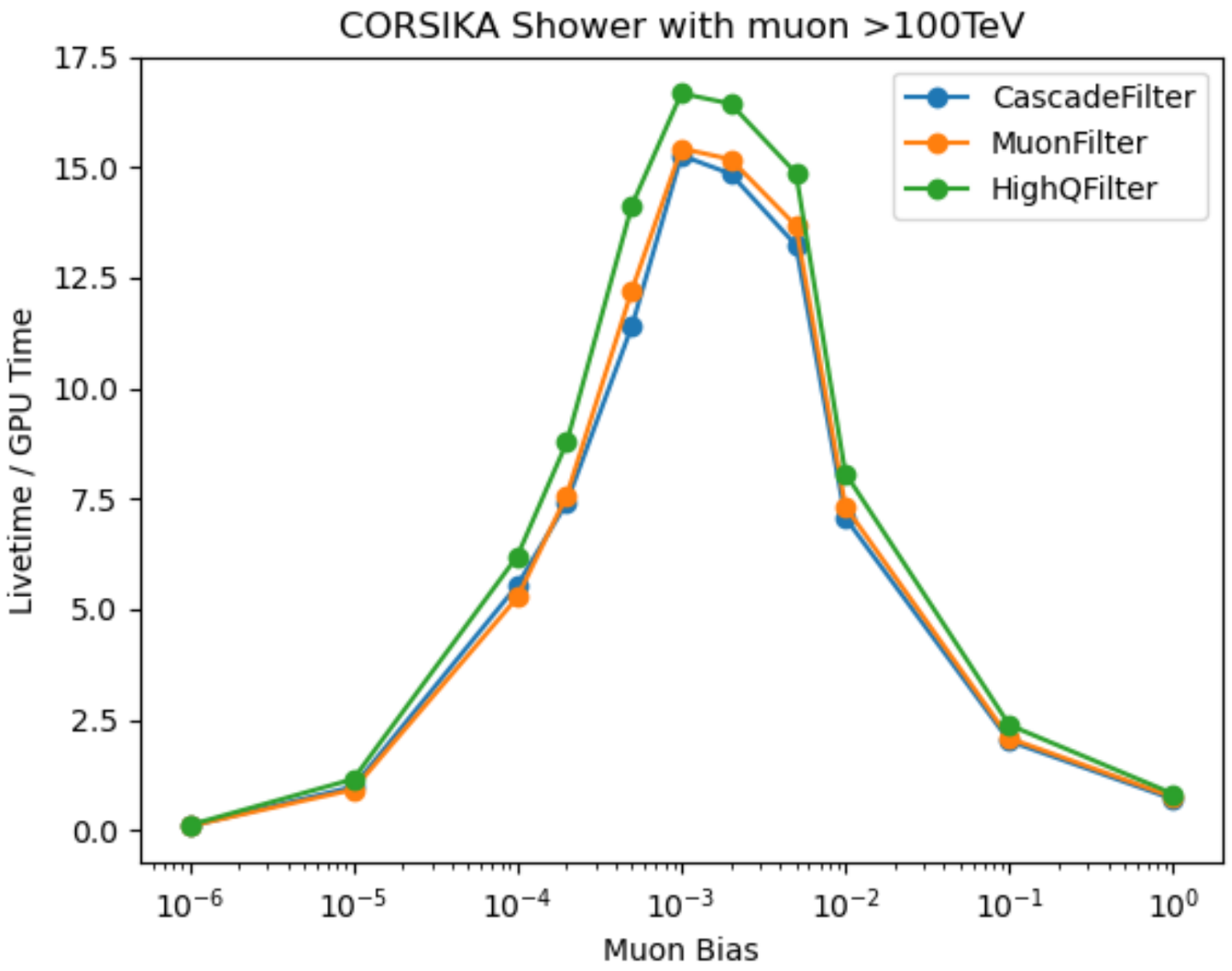




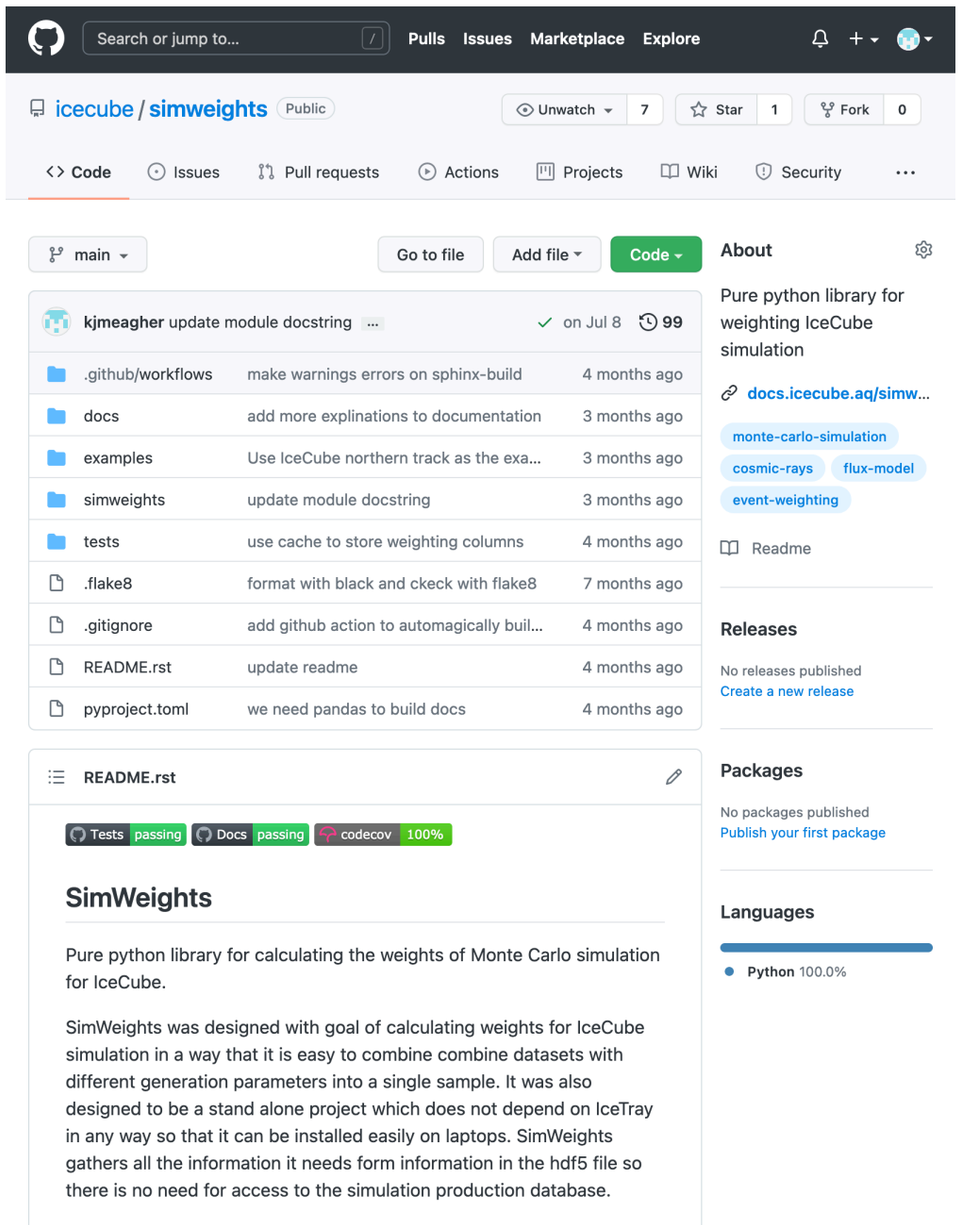
- 20789: Conventional CORSIKA dataset from 2016
- Baseline: Latest release run with conventional CORSIKA script
- 21889: New triggered CORSIKA

Performance

Biasing on the energy of the leading muon in the air shower results in a 20x increase in the simulate lifetime per GPU time



SimWeights for Weighting



- Available on GitHub as a small pure python module
- Does not access database
- Will work with any combination of generation surfaces
- Uses S-Frames to determine the number of files in sample
- Also works with older CORSIKA and NuGen

Proposal for PPC and CLSim

- The current terminology is confusing: when people say CLSim it can be very confusing.
- A lot of what is in the clsim project has nothing to do with OpenCL code
- Propose we move I3CLSimServer, I3CLSimCLient, etc to sim-services (or a new project) and drop the “CL” from the name.
- The project clsim will only contain actual OpenCL code
- Put CUDASim in a new project
- Now we can write a ppc implementation of I3SimStepToPhotonConverter which only depends on sim-services

Current Status

- Triggered CORSIKA can solve a number of issues associated with simulation production
- Triggered CORSIKA produces results which are comparable with the reference dataset
- A 20x increase in GPU utilization is achieved in real world benchmarks
- Weighting is handled by a new pure python module: SimWeights