Hardware Resources

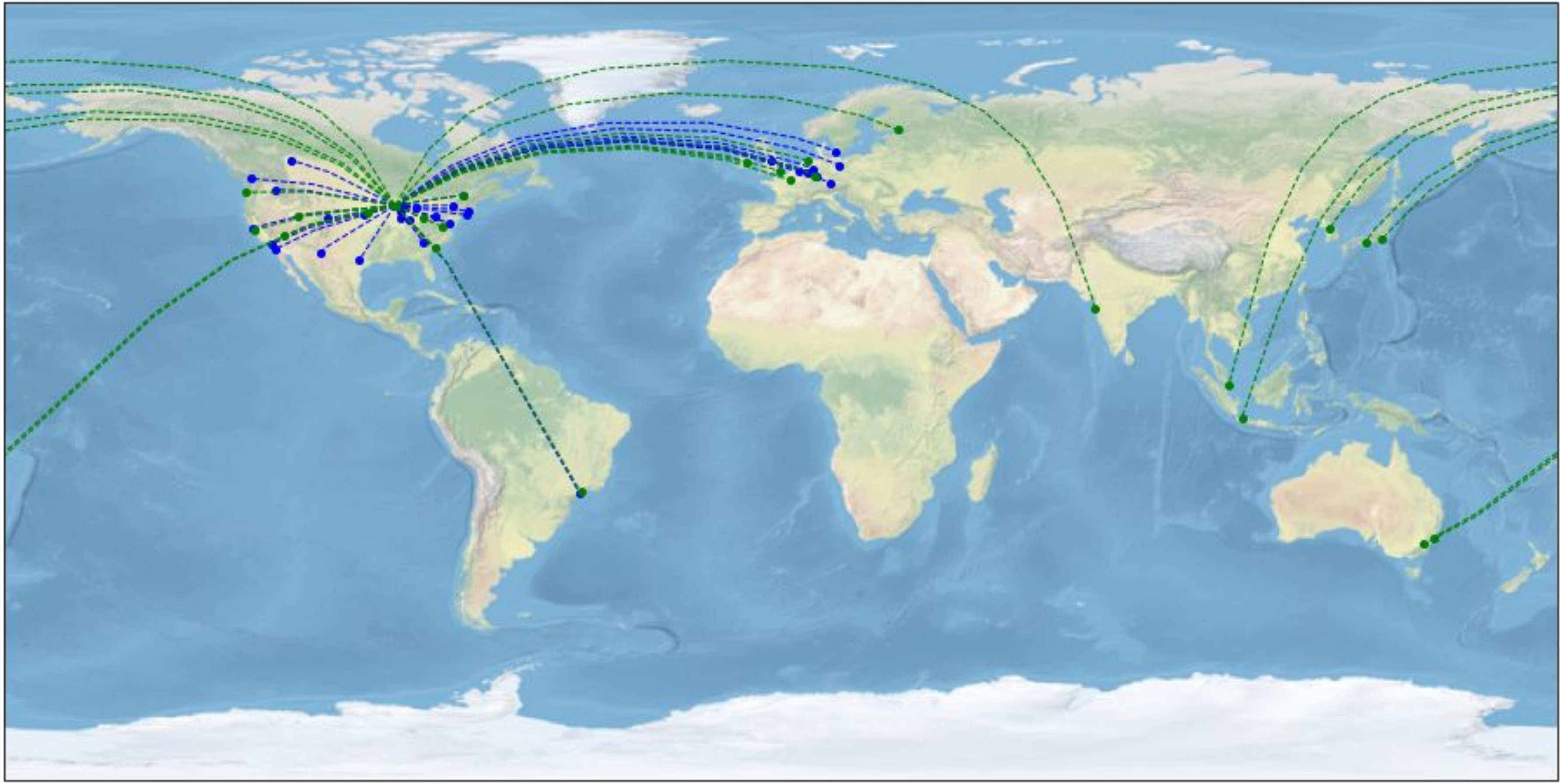
and why they matter

Sim Workshop 2020

David Schultz

Overview

- Current and future resources
- How we currently use resources
  - Memory
  - Disk / Network
  - GPU efficiency
- All the plots



Computing Resources

Years ago: mostly single cpu slots, 4 GB/core, HTC

Today: mostly 2-8 cpu slots, 2-3 GB/core, more large centers and HPC (especially for GPUs)

Future: >32 cpu slots, 1-2 GB/core, HPC and large centers, more accelerators and “special” pieces

Computing Resources

Major shifts:

- More centralized - this is the wish of NSF/DOE
  - Corollary: HPC centers will contain most resources
- More parallelized - multi-core is the only good way to increase chip performance now
- More specialized - see AI special data types, tensor cores
  - Expect more of this, with different machines requiring differently optimized code

Computing Resources

Notes on HPC:

- Generally network-challenged
  - Bad (or no) external connectivity from workers
  - Need to transfer through “data gateways”
- Gives us whole nodes
  - Common to get 64+ cores, multiple GPUs

Computing Resources

Notes on GPUs:

- Roughly a doubling of speed every ~2 years
- More die area to special hardware (AI, tensor cores)
- To get best performance, must use native tools
  - CUDA: many versions across grid

Computing Resources

NPX:

- Moving to be an analysis cluster
- Increasingly not for simulation
  - Backfill only

Current Resource Usage

Resource Goals

- Short jobs: between 20 minutes and 3 hours
- Low memory usage
- Low disk usage - especially for short runtimes
- High GPU utilization

Accurate resource predictions!

Resource Usage Plots

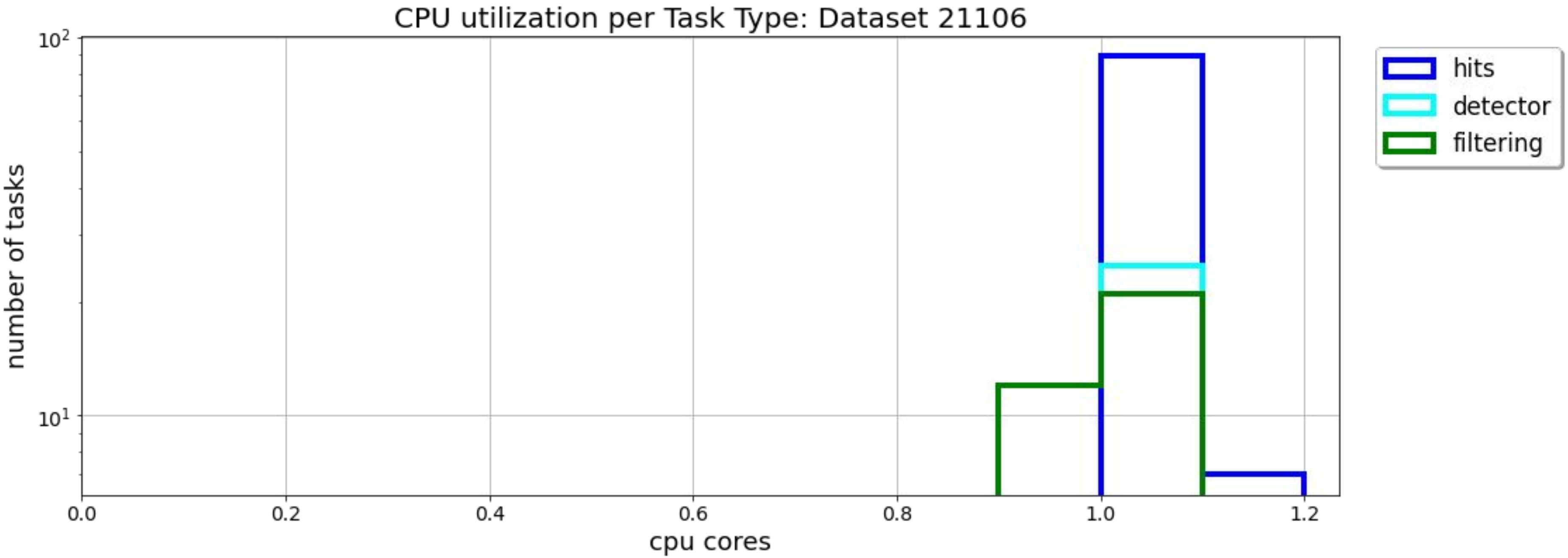
Guide for reading graphs:

- Stats from IceProd
- Disk equates to network i/o
- Measures peak memory usage (sustained over a minute of time)
- Failures/evictions usually mean going over requested resources

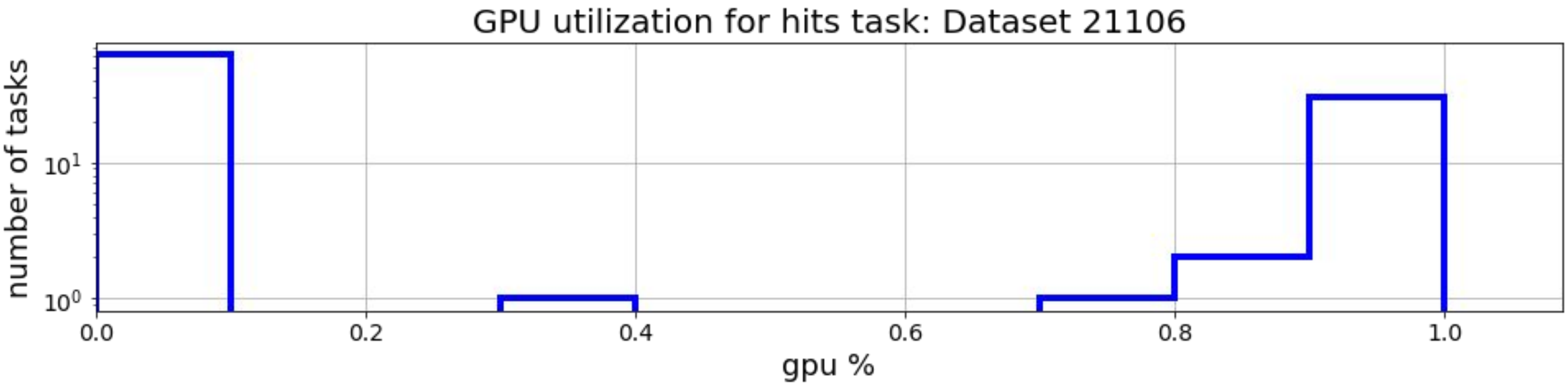
Usage - MuonGun - 21106

simulation/V06-01-02

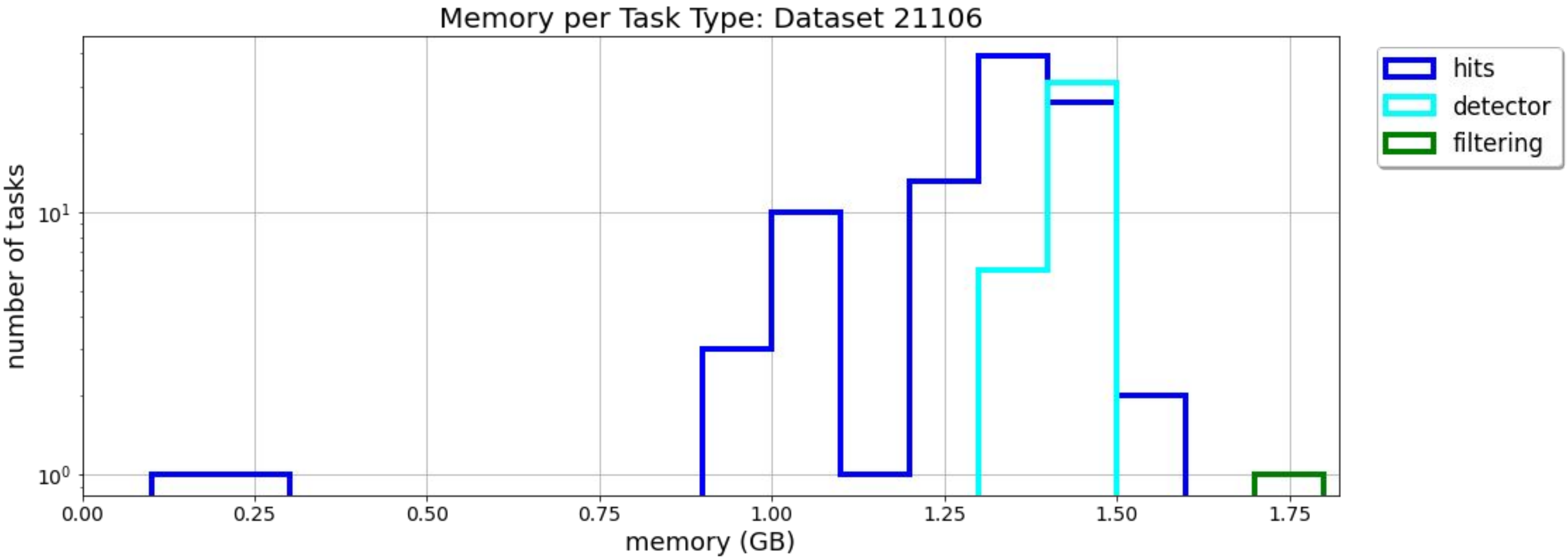
Usage - MuonGun



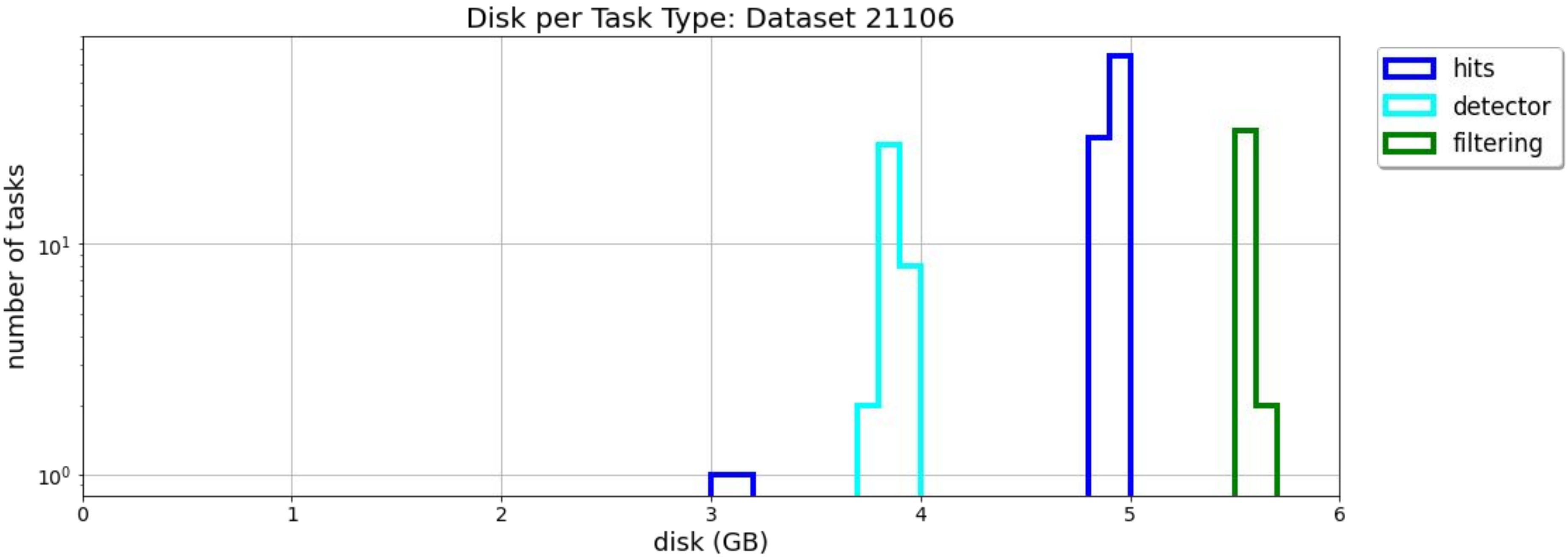
Usage - MuonGun



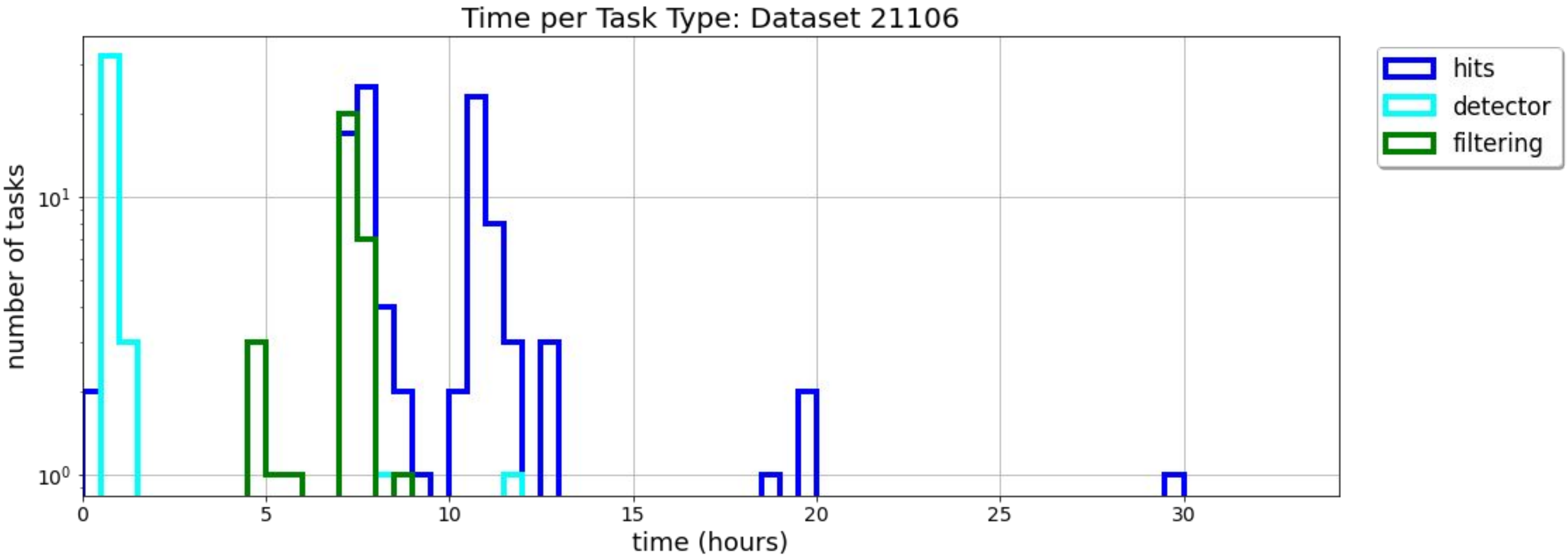
Usage - MuonGun



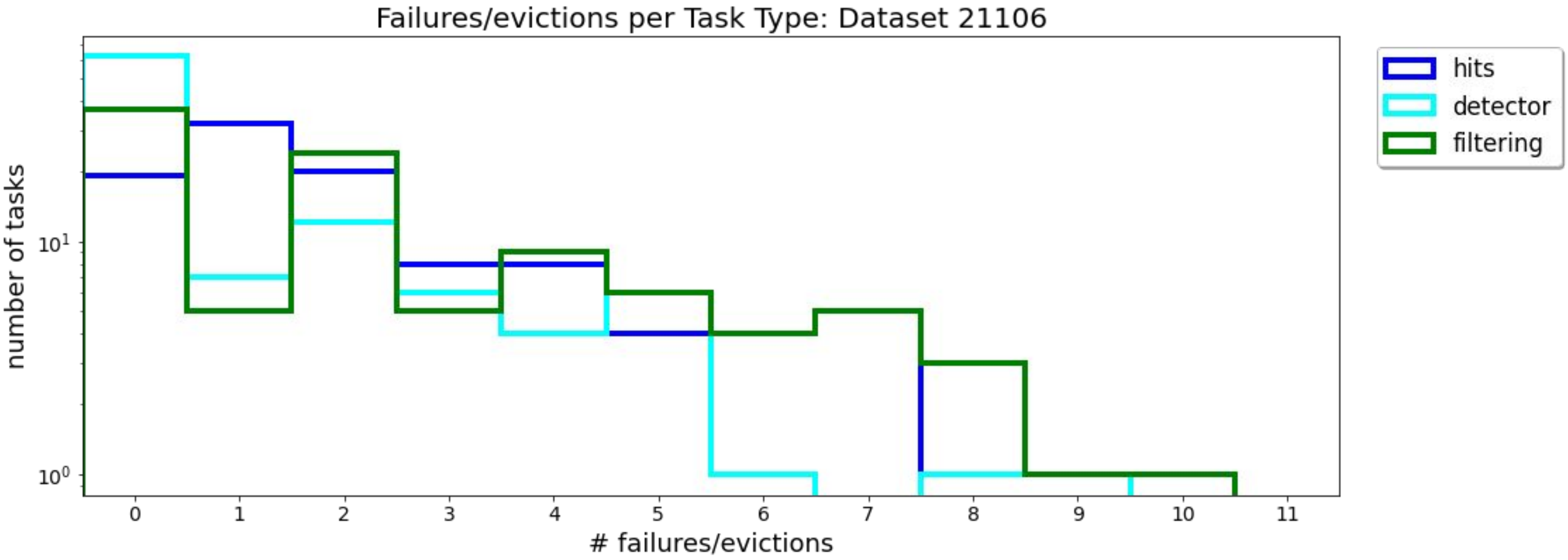
Usage - MuonGun



Usage - MuonGun



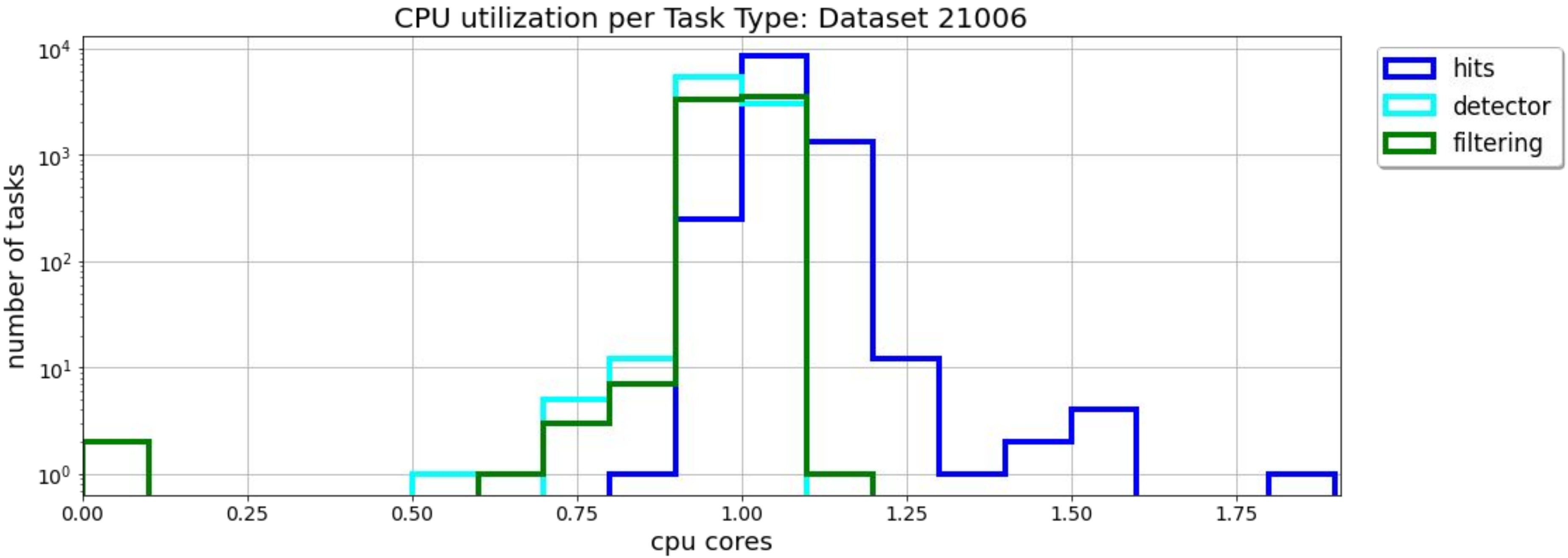
Usage - MuonGun



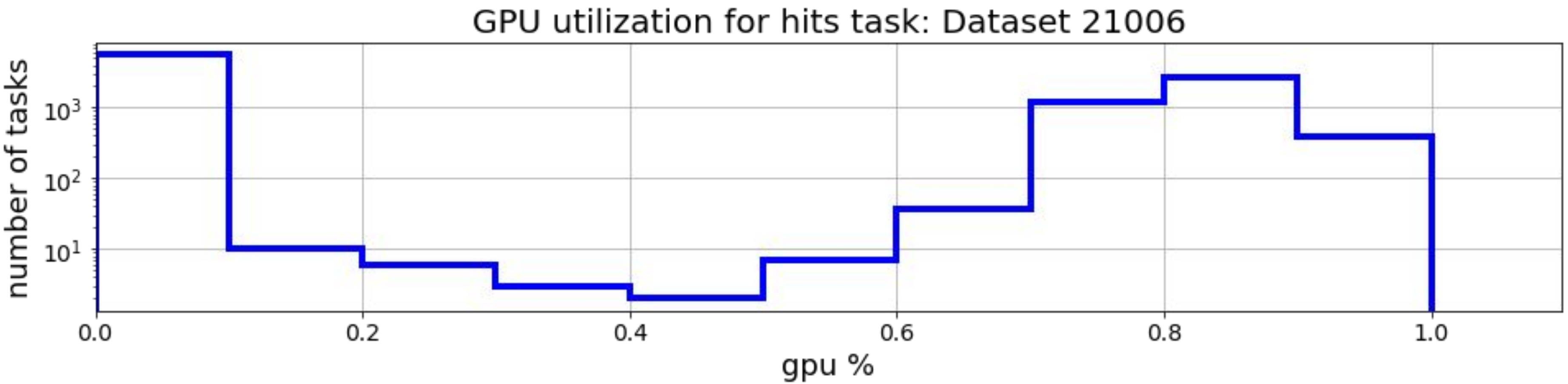
Usage - NuMu - 21006

combo/V00-00-03

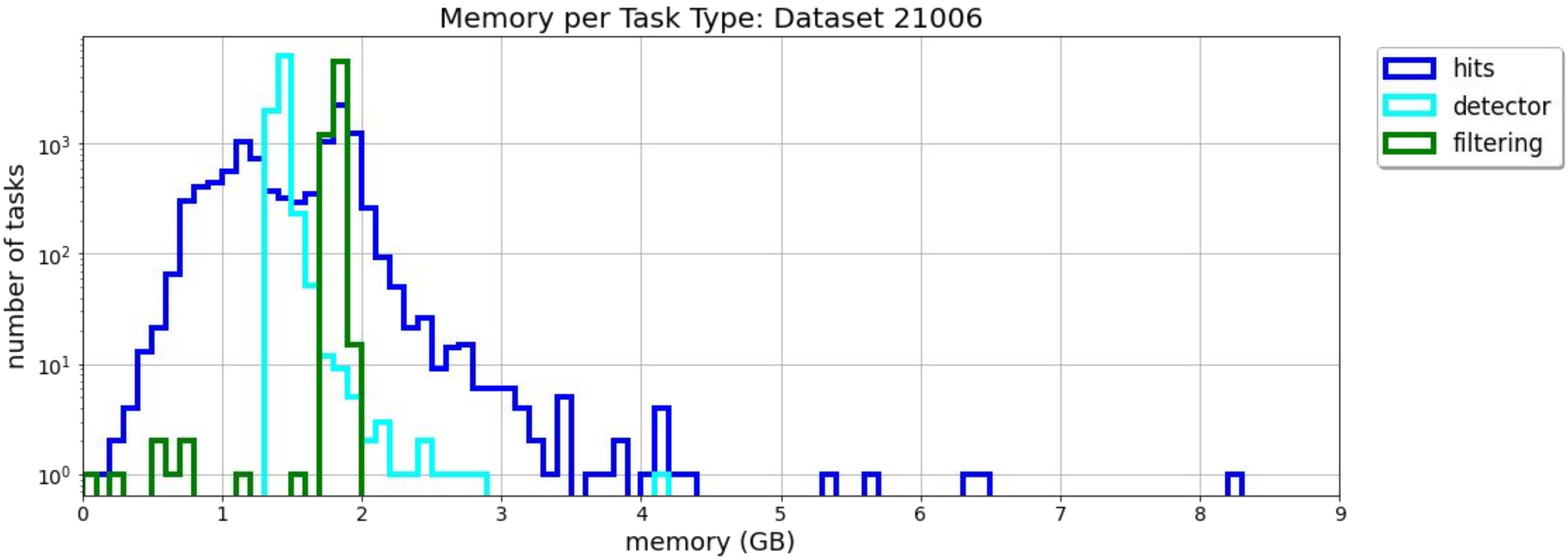
Usage - NuMu


CPU utilization per Task Type: Dataset 21006
number of tasks
10^0
10^1
10^2
10^3
10^4
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
cpu cores
hits
detector
filtering

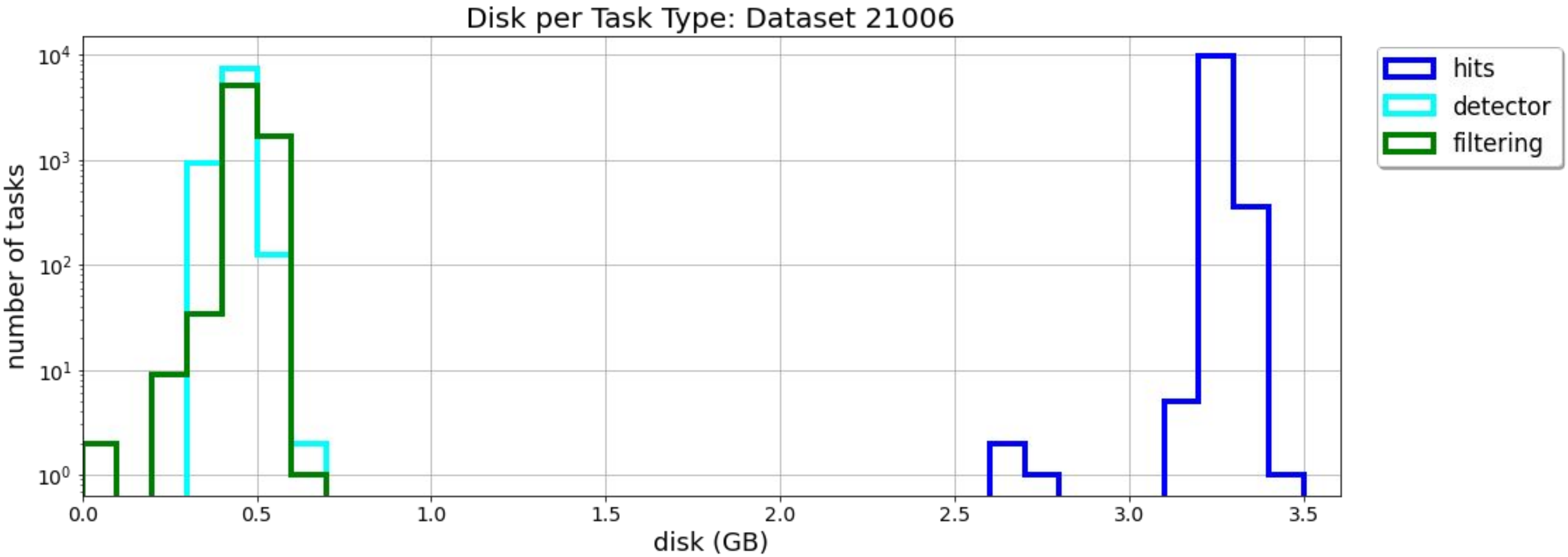
Usage - NuMu


GPU utilization for hits task: Dataset 21006
number of tasks
10^3
10^2
10^1
0.0
0.2
0.4
0.6
0.8
1.0
gpu %

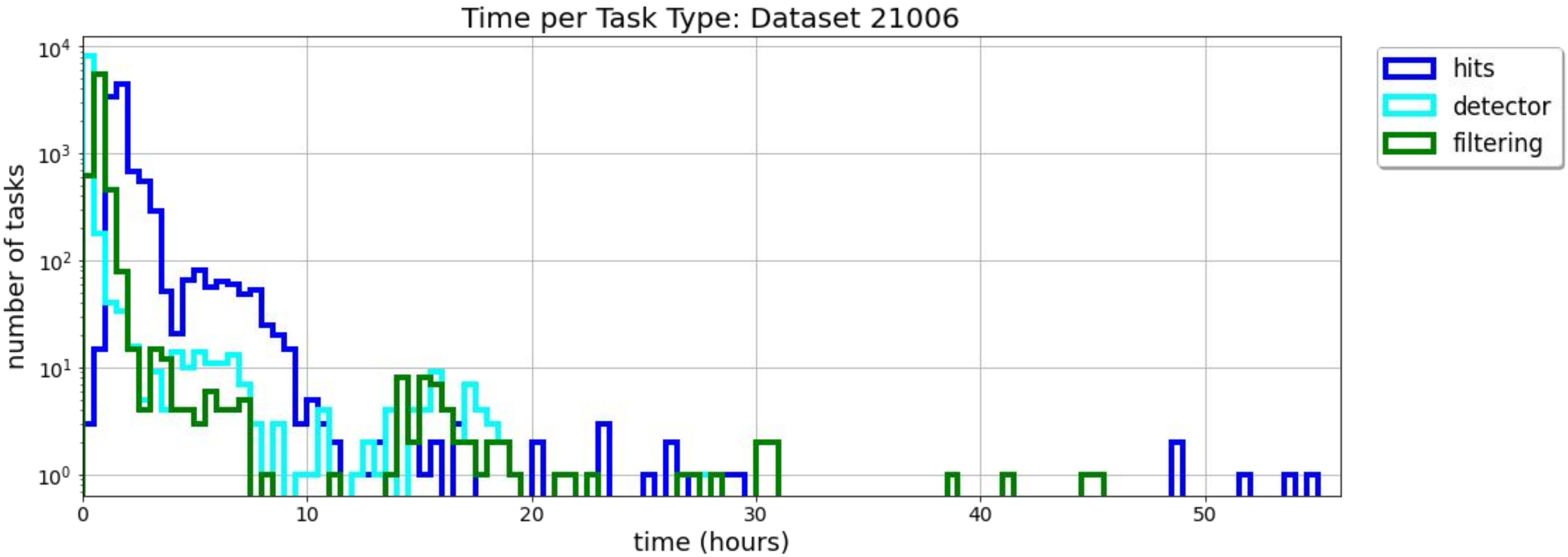
Usage - NuMu


Memory per Task Type: Dataset 21006
number of tasks
10^0
10^1
10^2
10^3
0
1
2
3
4
5
6
7
8
9
memory (GB)
hits
detector
filtering

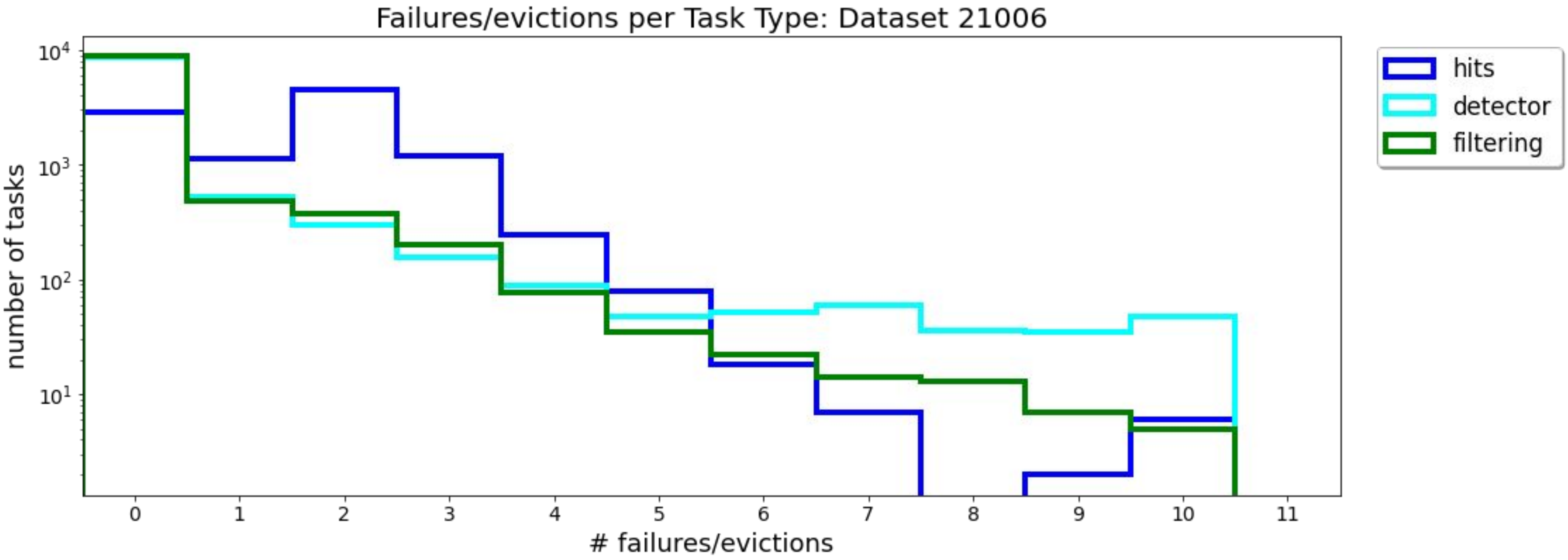
Usage - NuMu


Disk per Task Type: Dataset 21006
number of tasks
10^0
10^1
10^2
10^3
10^4
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
disk (GB)
hits
detector
filtering

Usage - NuMu


Time per Task Type: Dataset 21006
number of tasks
10^0
10^1
10^2
10^3
10^4
time (hours)
0
10
20
30
40
50
hits
detector
filtering

Usage - NuMu

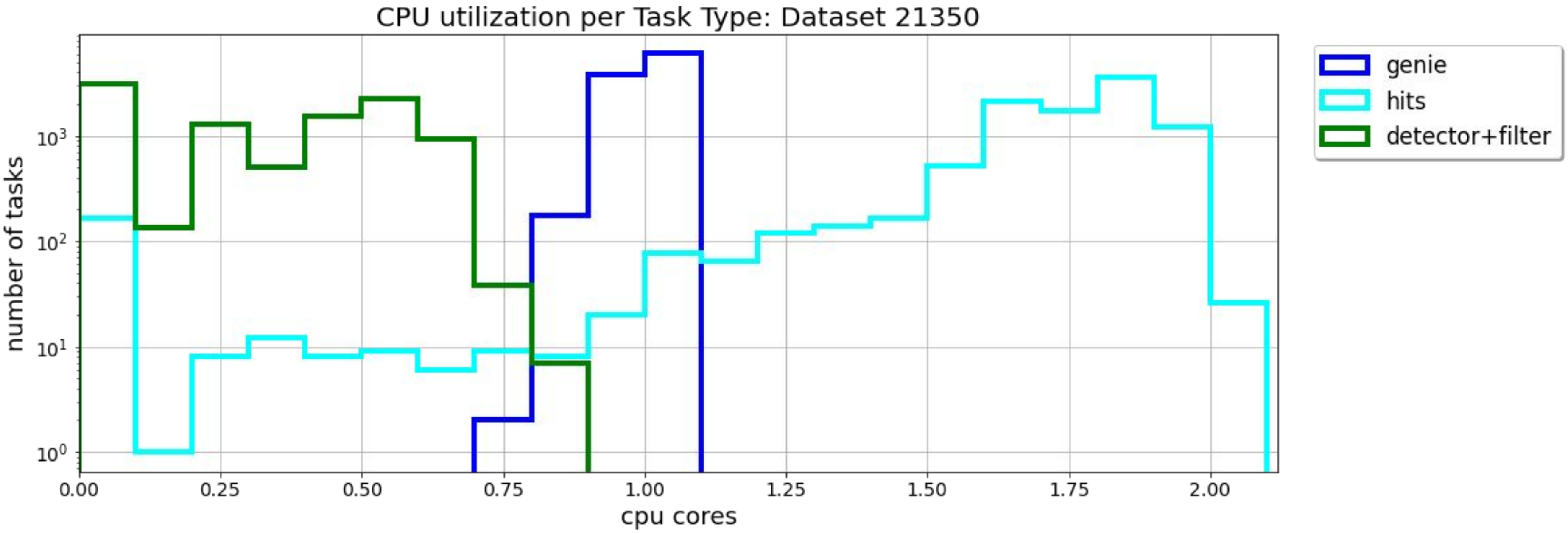


Failures/evictions per Task Type: Dataset 21006
number of tasks
10^4
10^3
10^2
10^1
0
1
2
3
4
5
6
7
8
9
10
11
# failures/evictions
hits
detector
filtering

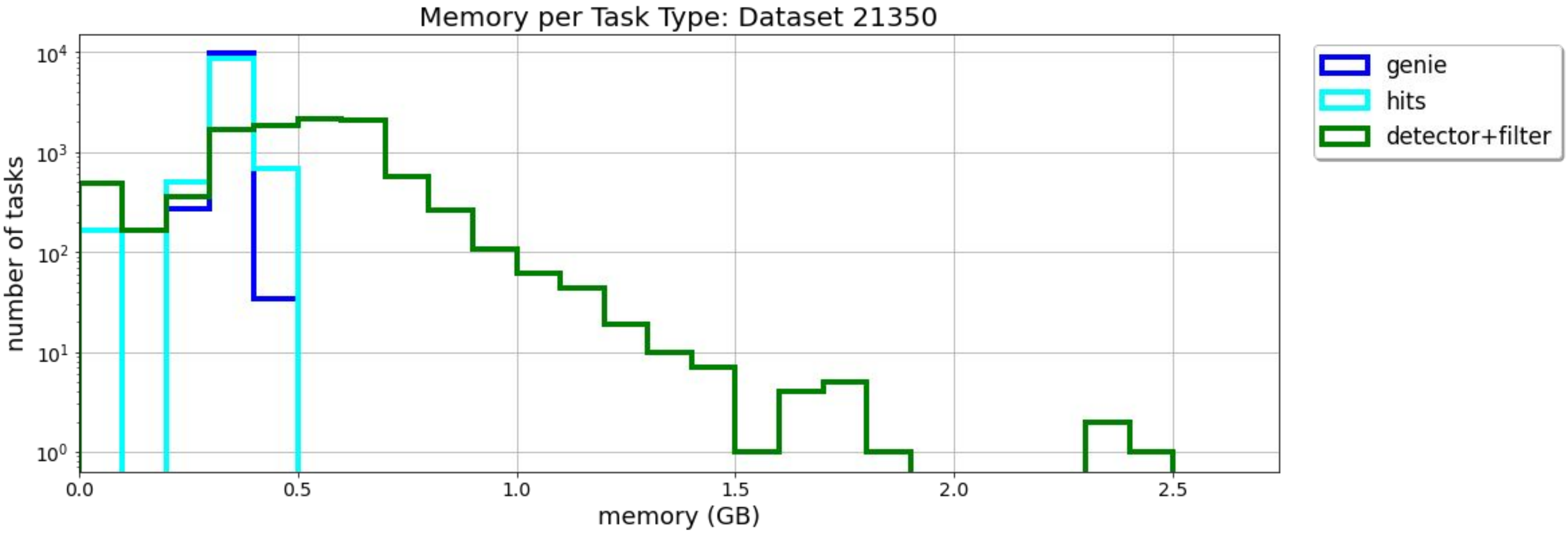
Usage - GENIE - 21350

simulation/V06-01-01

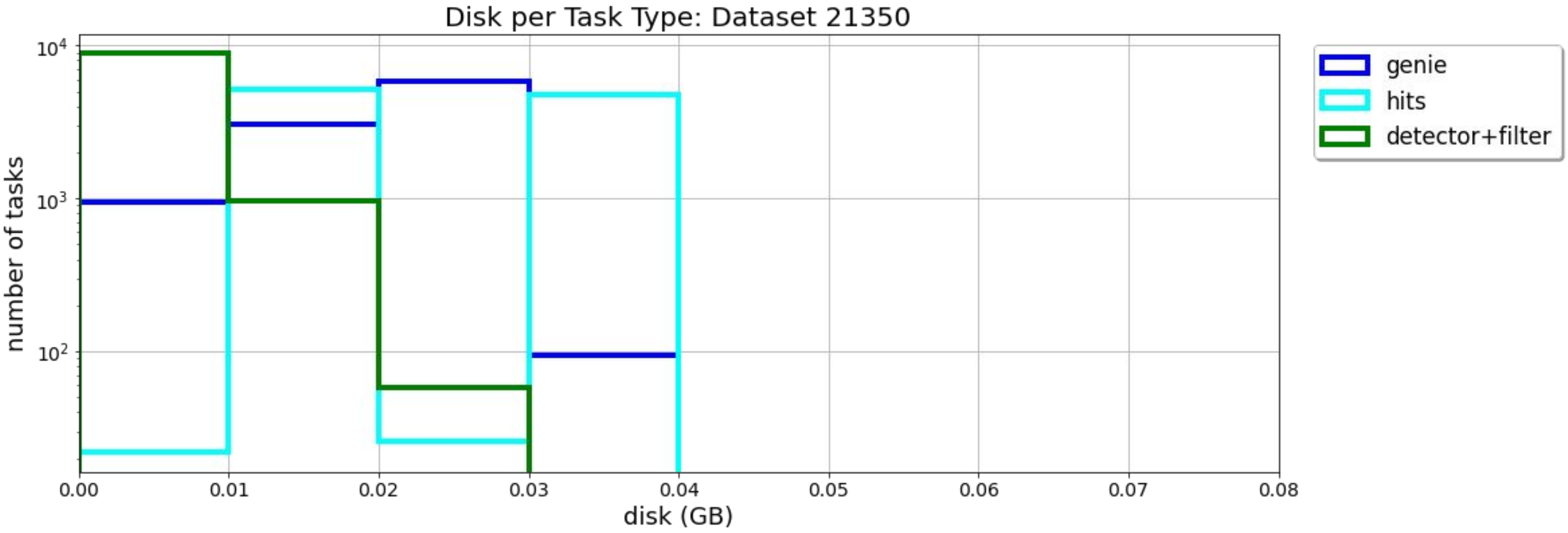
Usage - GENIE


CPU utilization per Task Type: Dataset 21350
number of tasks
10^0
10^1
10^2
10^3
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
cpu cores
genie
hits
detector+filter

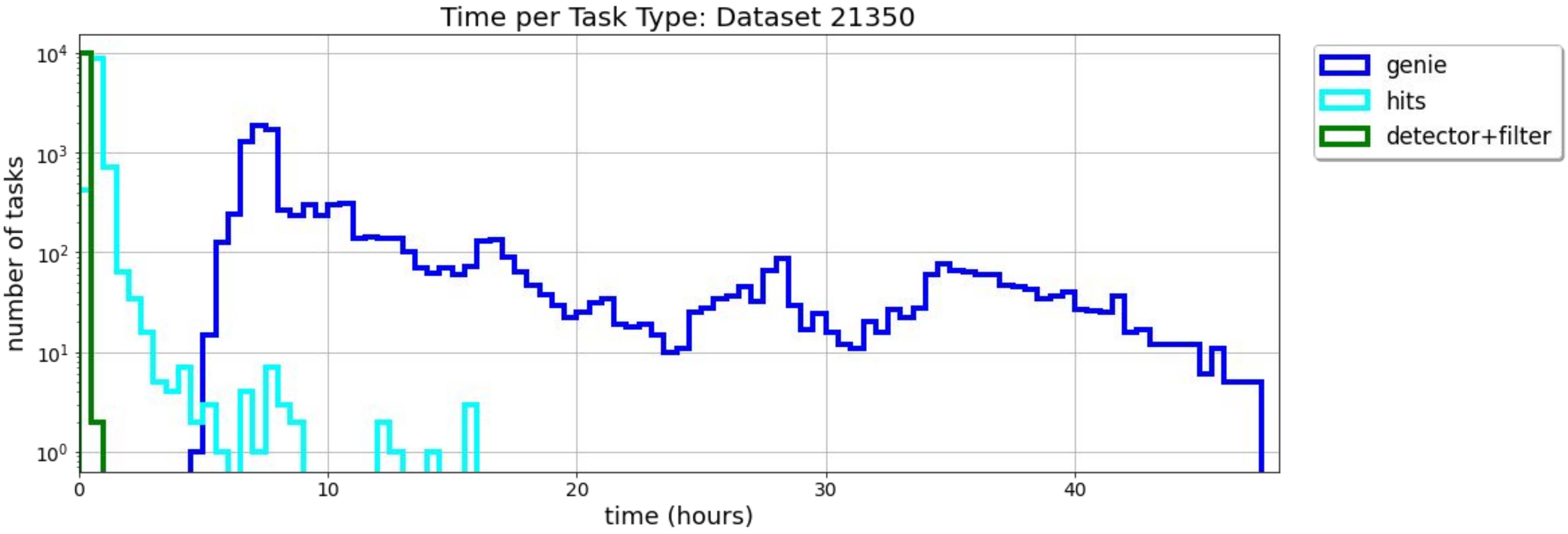
Usage - GENIE


Memory per Task Type: Dataset 21350
number of tasks
10^0
10^1
10^2
10^3
10^4
0.0
0.5
1.0
1.5
2.0
2.5
memory (GB)
genie
hits
detector+filter

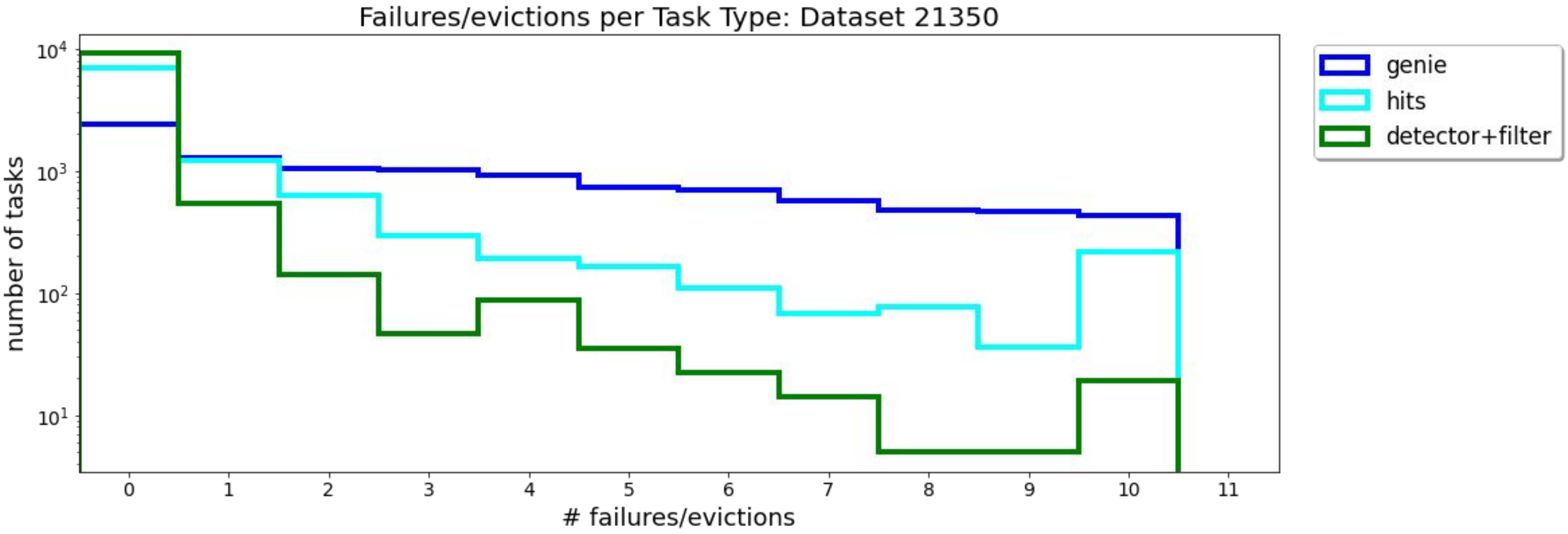
Usage - GENIE


Disk per Task Type: Dataset 21350
number of tasks
10^4
10^3
10^2
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
disk (GB)
genie
hits
detector+filter

Usage - GENIE


Time per Task Type: Dataset 21350
number of tasks
10^0
10^1
10^2
10^3
10^4
0
10
20
30
40
time (hours)
genie
hits
detector+filter

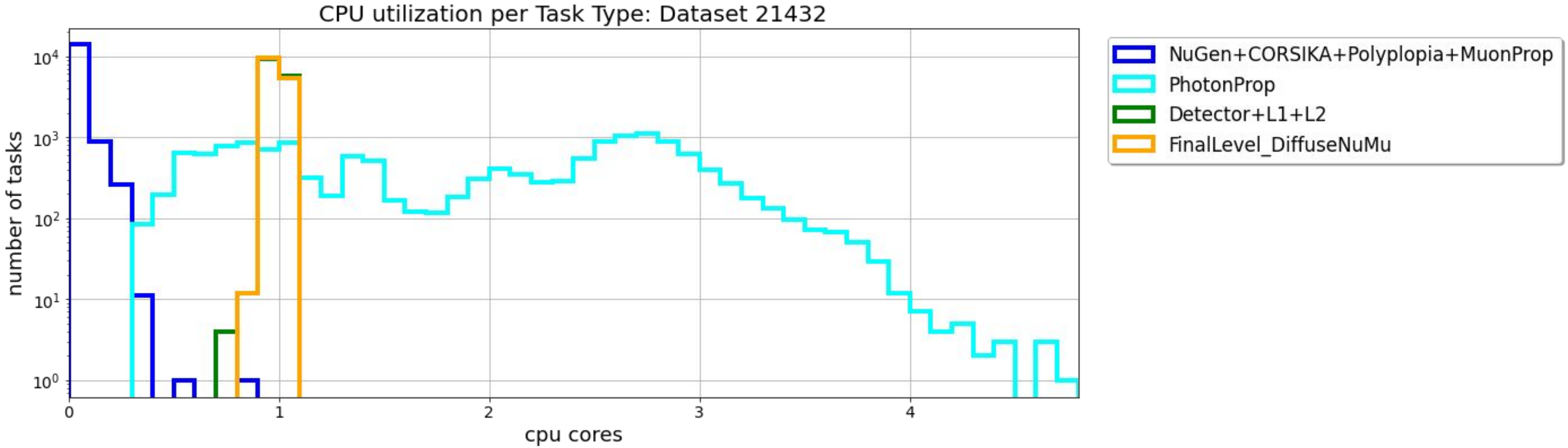
Usage - GENIE


Failures/evictions per Task Type: Dataset 21350
number of tasks
10^4
10^3
10^2
10^1
0
1
2
3
4
5
6
7
8
9
10
11
# failures/evictions
genie
hits
detector+filter

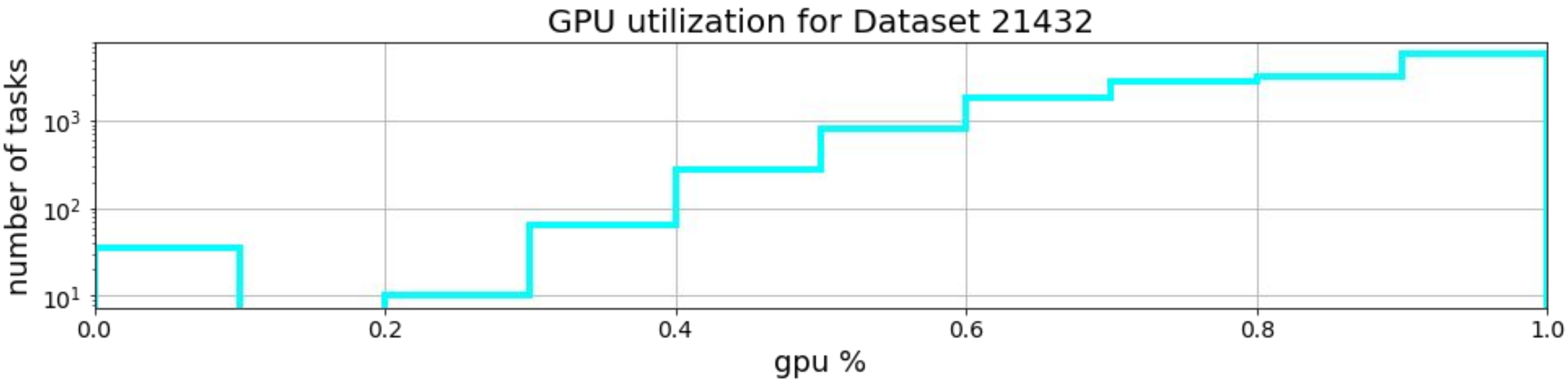
Usage - GlobalFit Snowstorm - 21432

combo/V01-00-02

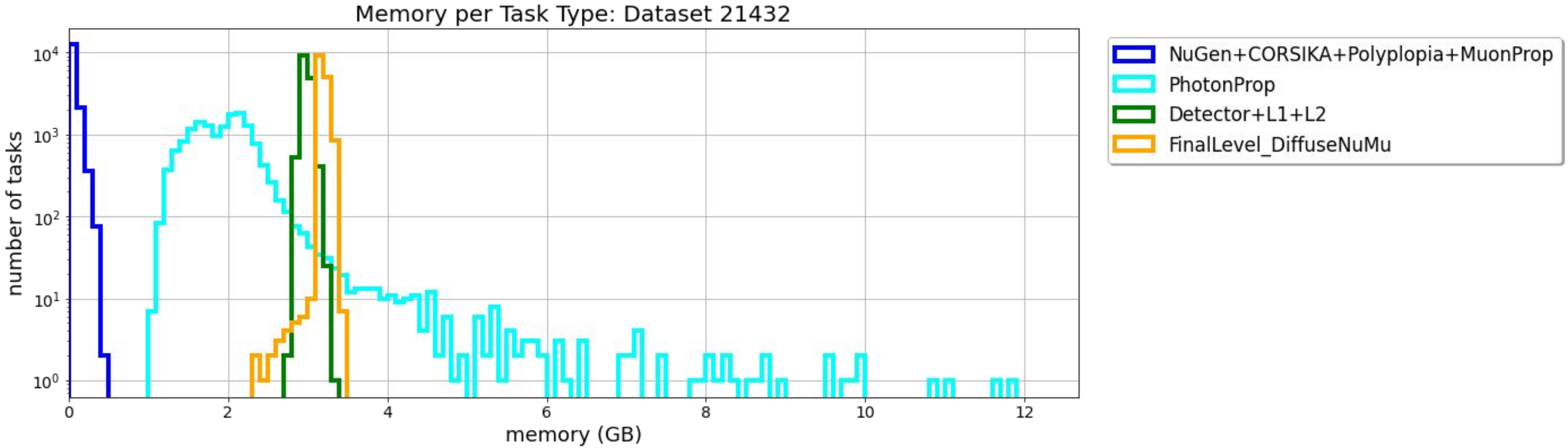
Usage - Snowstorm



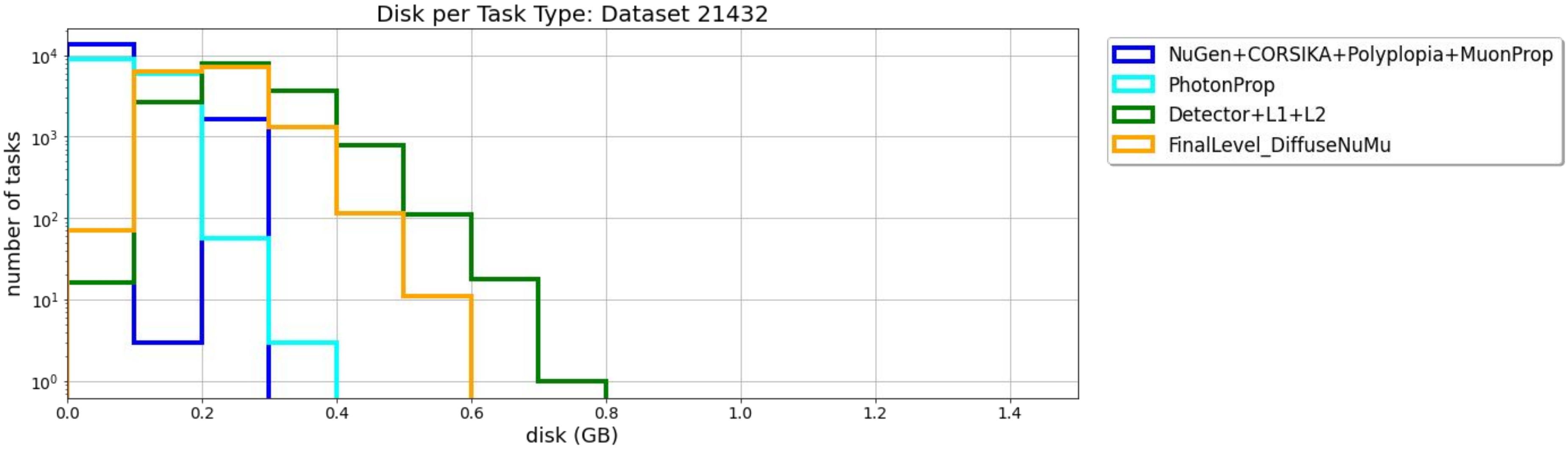
Usage - Snowstorm



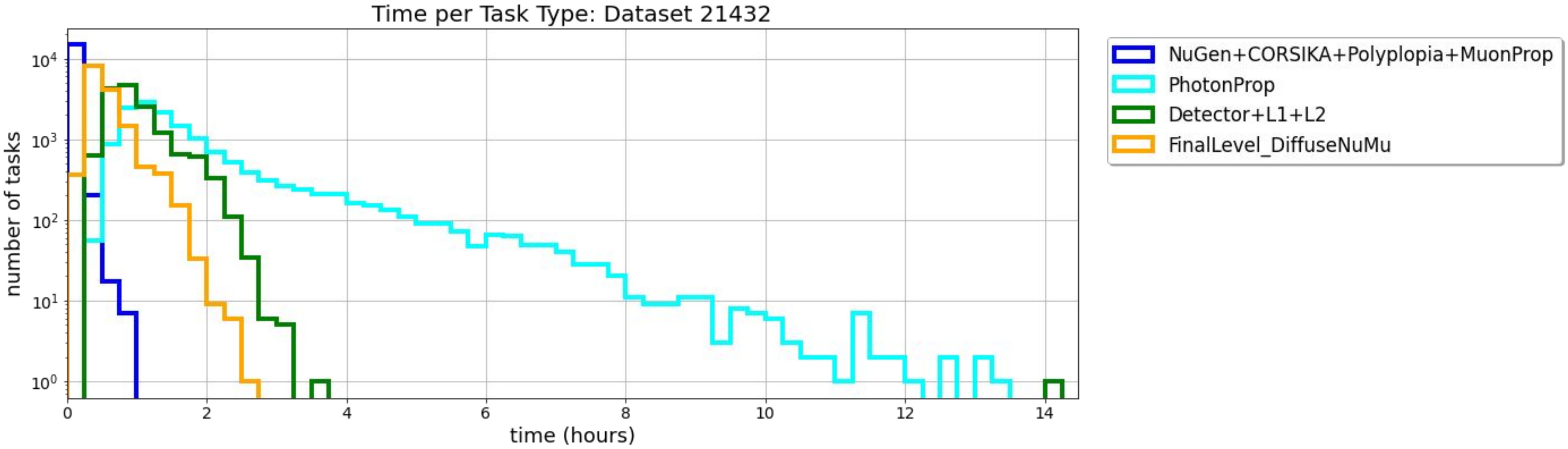
Usage - Snowstorm



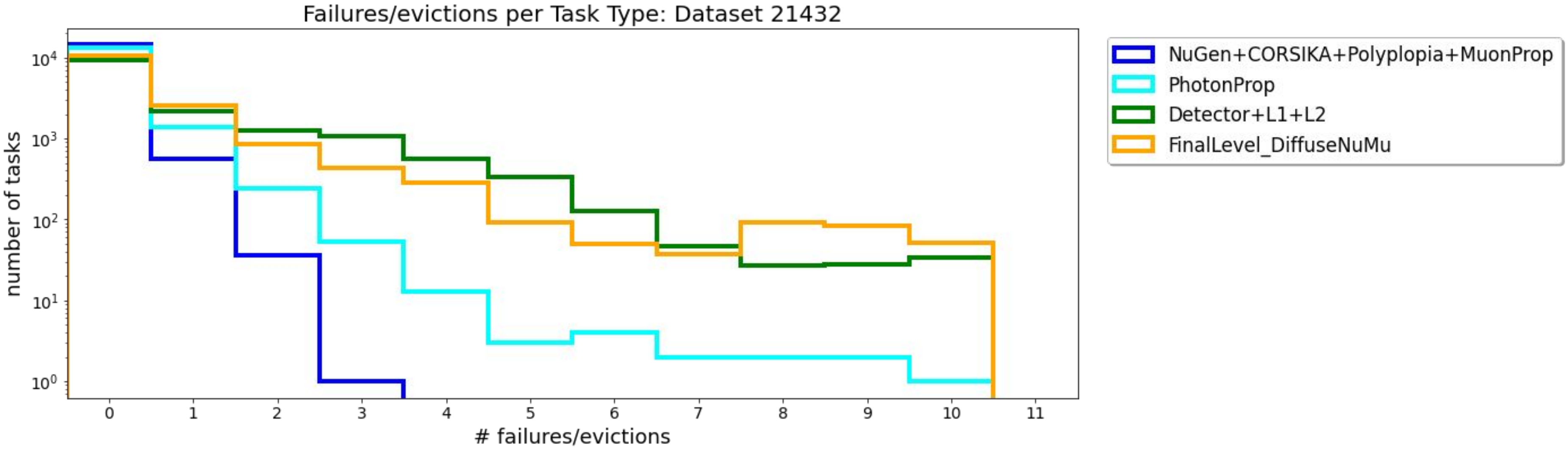
Usage - Snowstorm



Usage - Snowstorm



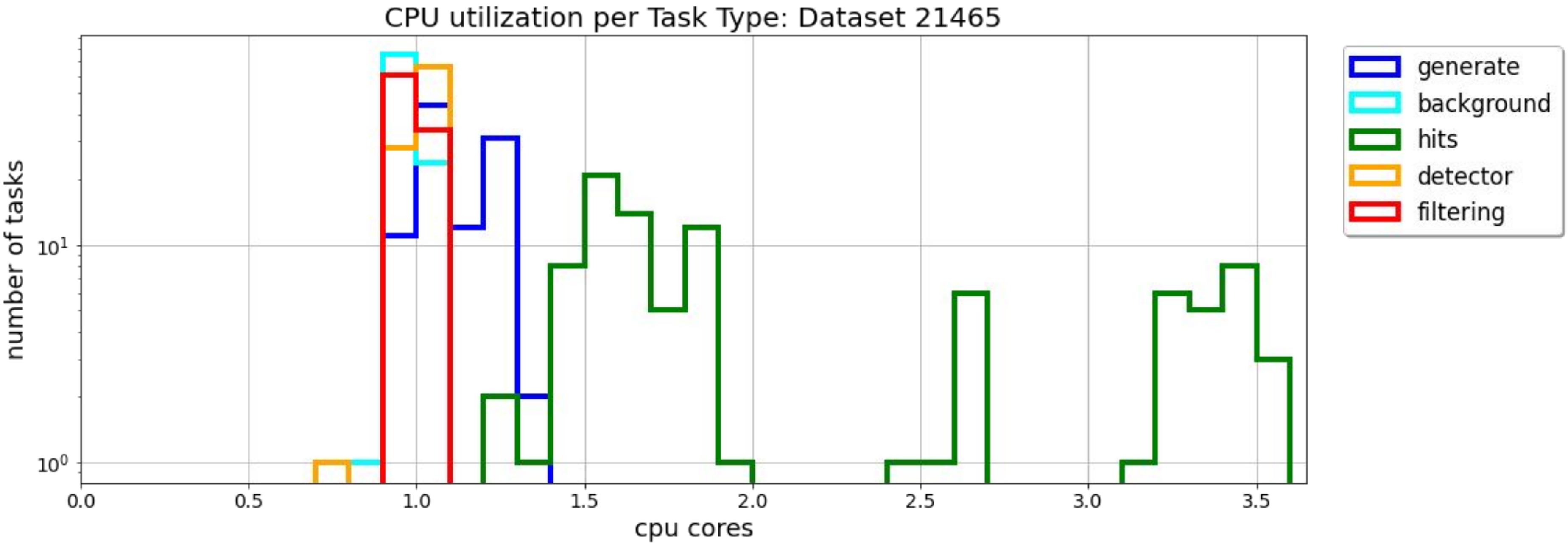
Usage - Snowstorm



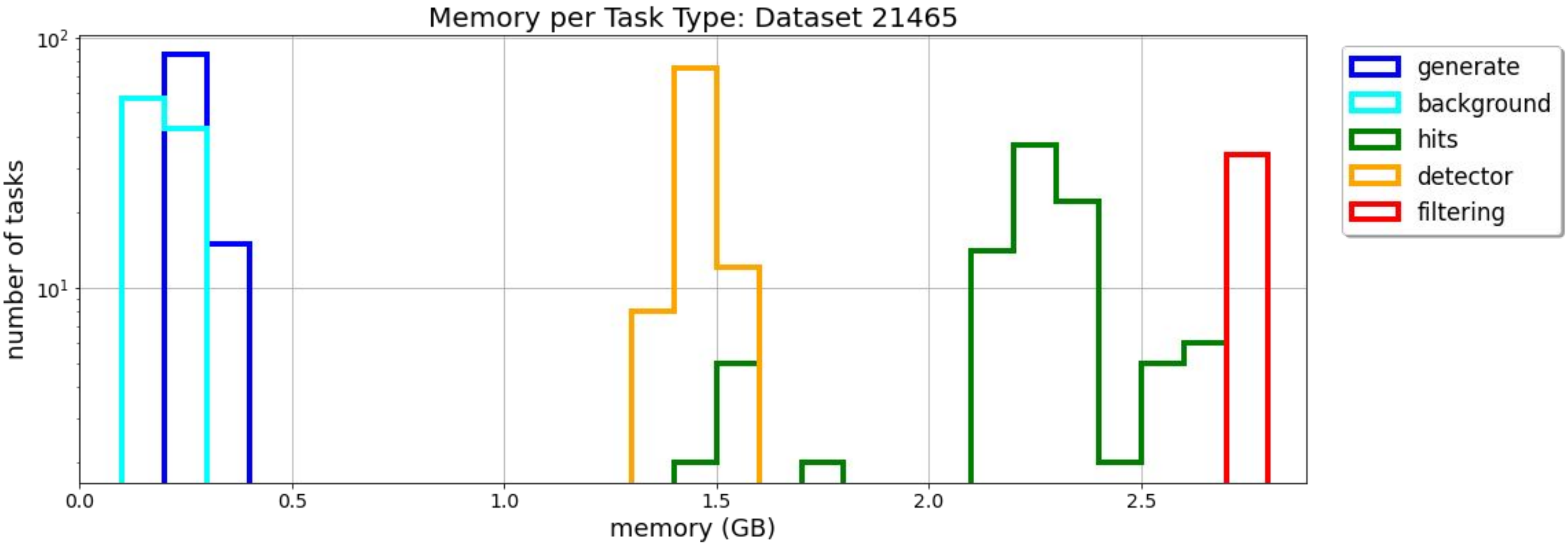
Usage - AWS Demo 3 - CORSIKA 1e3-1e8 - 21465

combo/V00-00-03

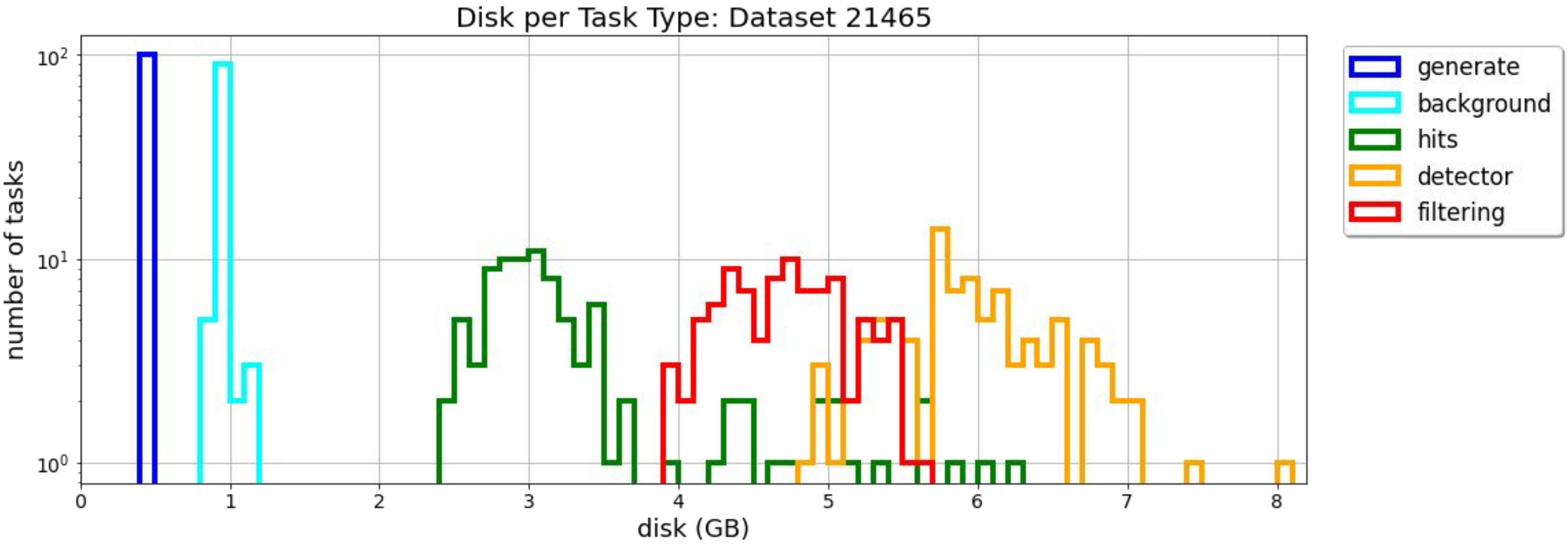
Usage - AWS Demo 3



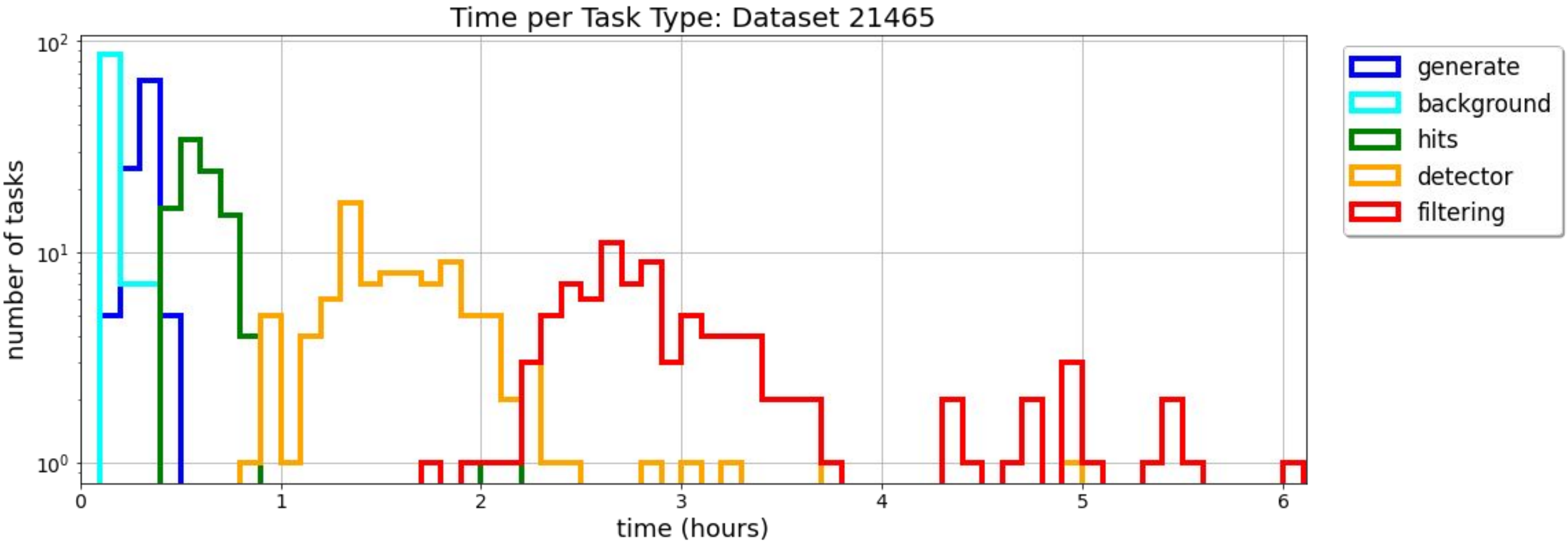
Usage - AWS Demo 3



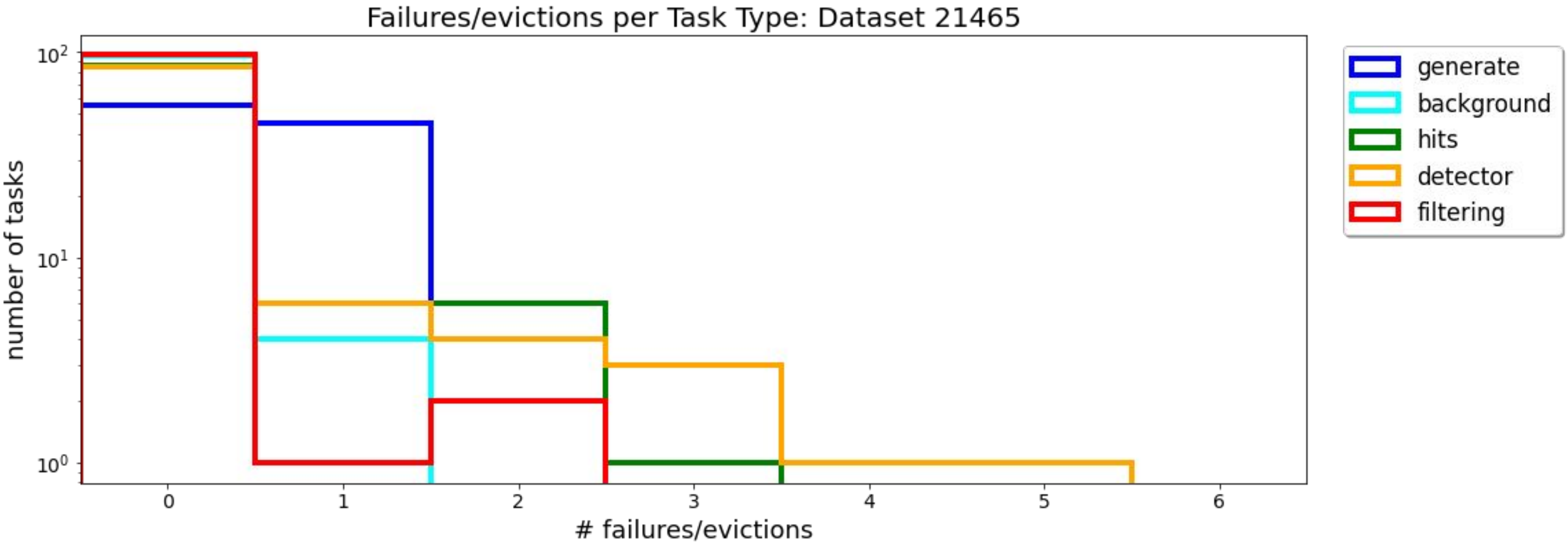
Usage - AWS Demo 3



Usage - AWS Demo 3



Usage - AWS Demo 3



GPU Usage / Efficiency

GPU Usage / Efficiency

We last discussed this several years ago

- Fixed issue with buffer sizing then

But, GPUs have gotten faster: 4x improvement

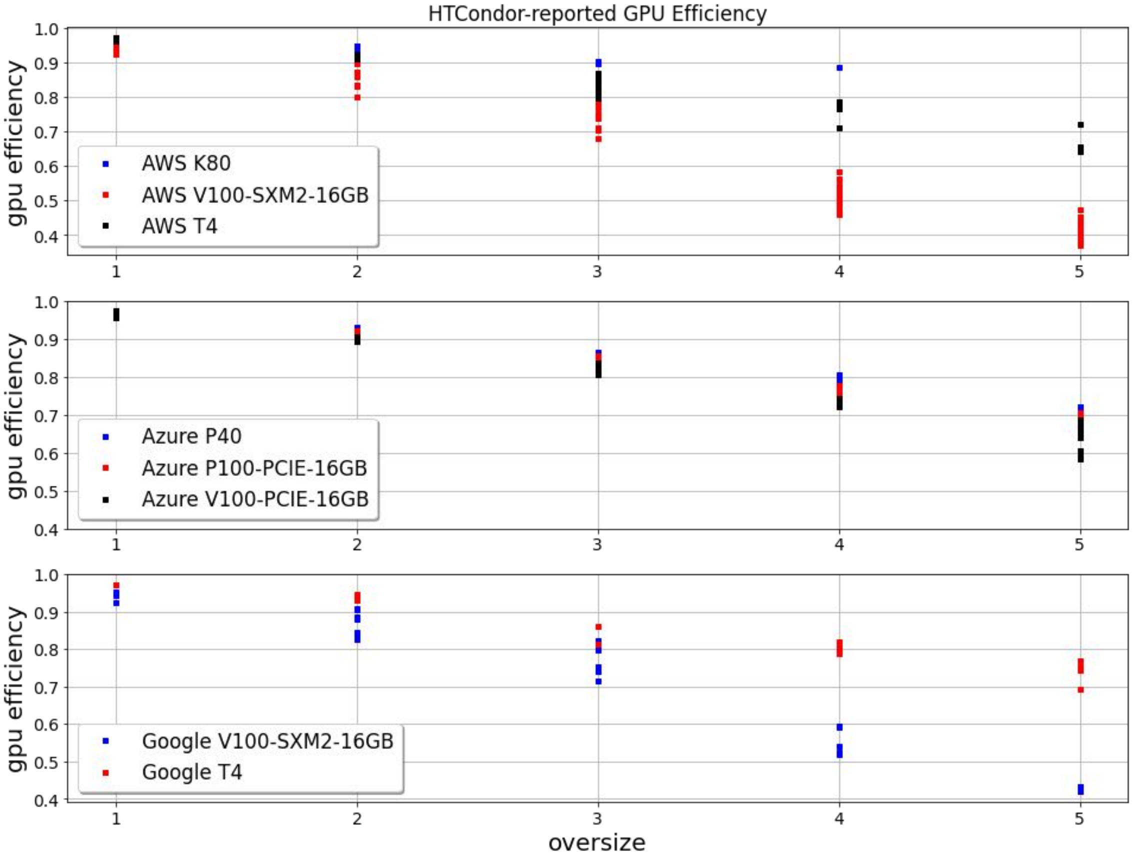
Results from cloud testing indicate new problems

GPU Usage - AWS Demo 3

GPU efficiency as a factor of dom oversize

Azure nodes have faster CPU cores

- likely bottleneck: a CPU thread



GPU Usage / Efficiency

So, good news for JvS - we have extra headroom to implement the expensive random solution

Bad news for clsim maintainer - there is a clear bottleneck on the CPU

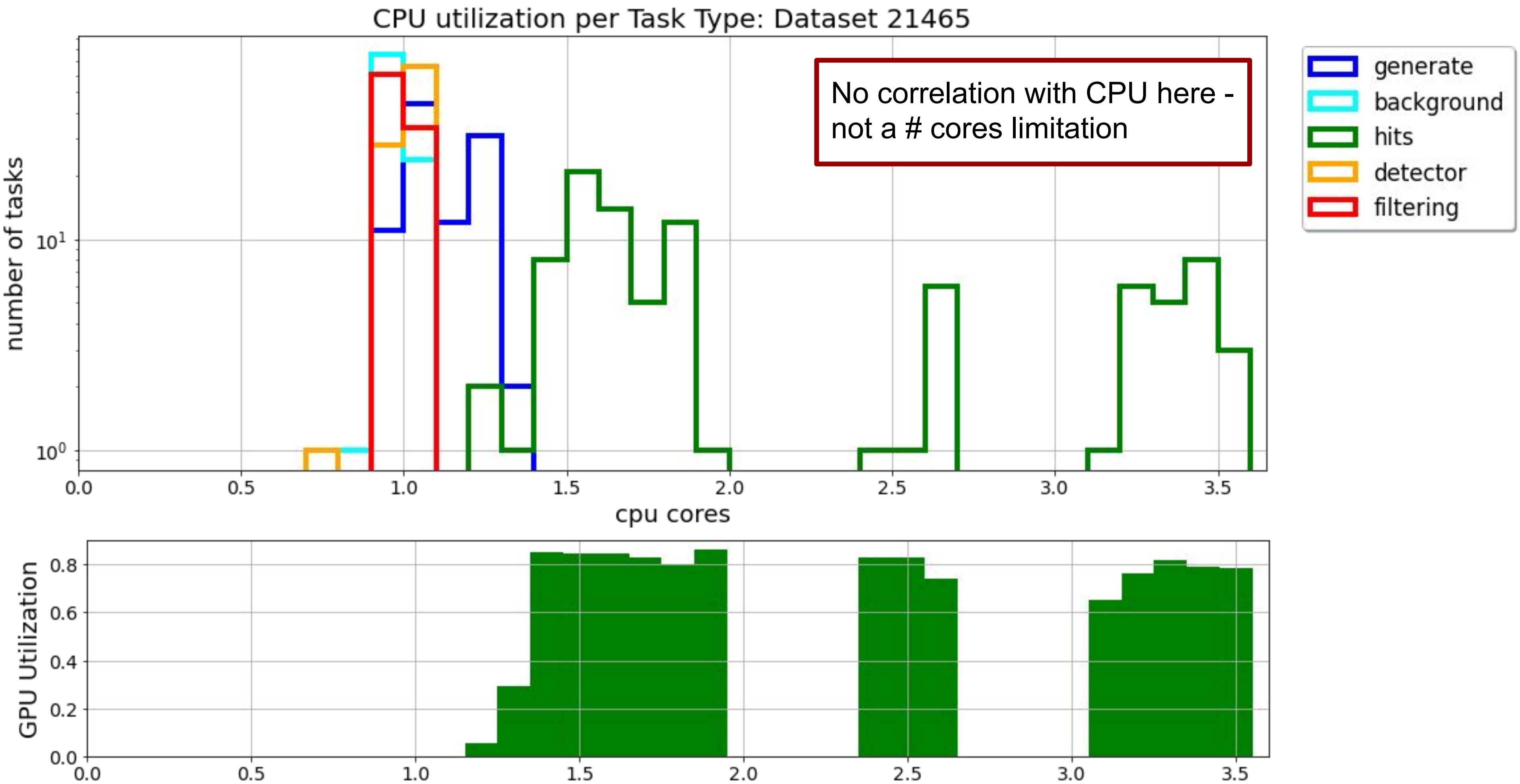
Worse news - Nvidia is promising to make our GPU code even faster, making this problem more visible

Resource Usage Observations

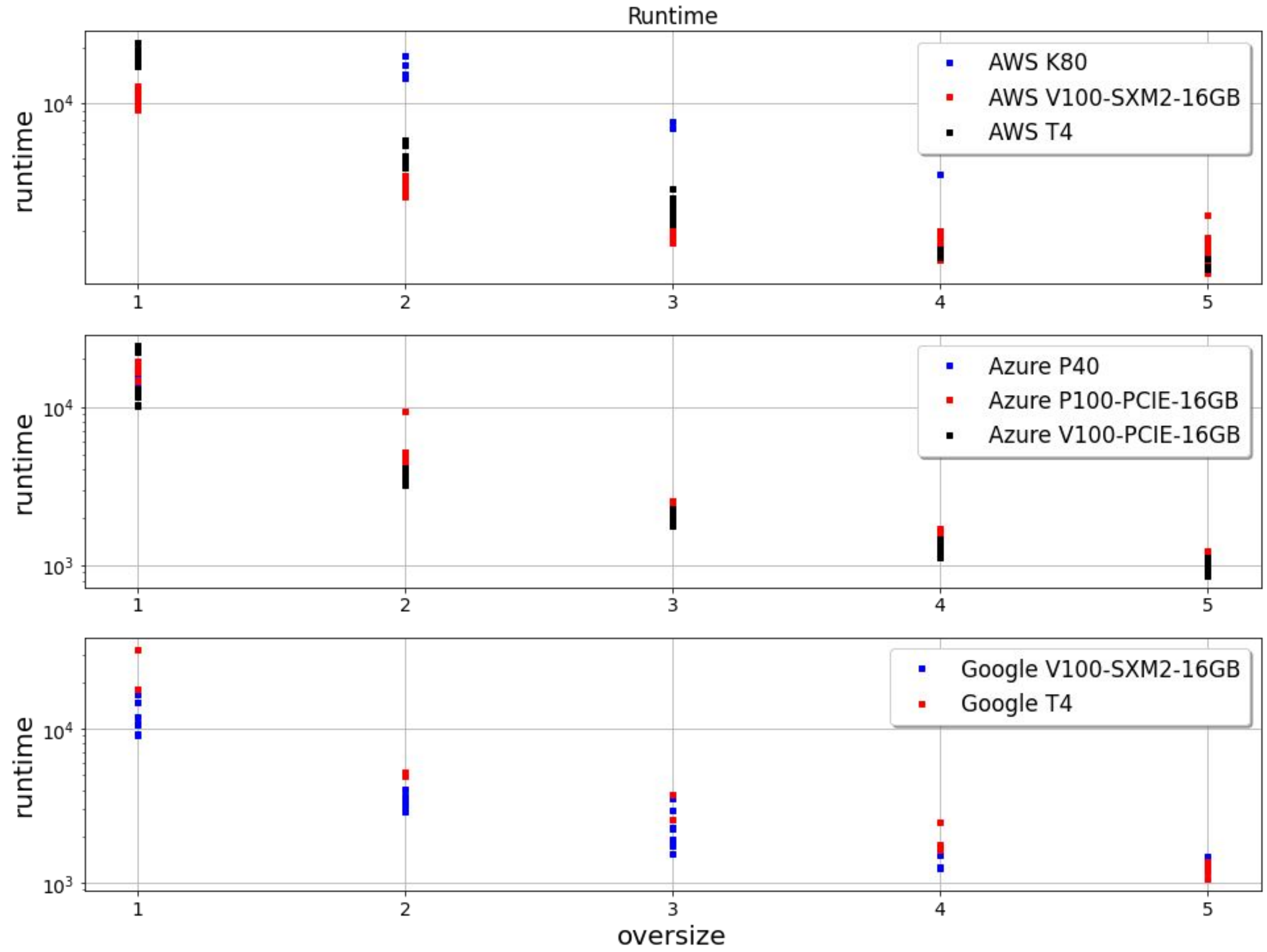
Overall Grades:

- CPU: acceptable
- GPU: poor
- Memory: acceptable
- Disk: exceeds expectations
- Runtime: dreadful
- Failures: troll

Backup (more plots)



No correlation with CPU here - not a # cores limitation


Runtime
runtime
10^4
1
2
3
4
5
AWS K80
AWS V100-SXM2-16GB
AWS T4
runtime
10^4
10^3
1
2
3
4
5
Azure P40
Azure P100-PCIE-16GB
Azure V100-PCIE-16GB
runtime
10^4
10^3
1
2
3
4
5
Google V100-SXM2-16GB
Google T4
oversize