Provisioning and Usage of GPUs at GridKa

Matthias Schnepf | 3. April 2025

Back in 2020

- Physicists: GPUs are great
  - more efficient reconstruction, e.g., Patatrack
  - fast sim
  - more efficient Matrix element calculation, e.g., MadGraph5 for GPUs
  - training of NN

Back in 2020

- Physicists: GPUs are great
  - more efficient reconstruction, e.g., Patatrack
  - fast sim
  - more efficient Matrix element calculation, e.g., MadGraph5 for GPUs
  - training of NN
- Experiments: we do not support GPUs because there are no in the Grid

Back in 2020

- Physicists: GPUs are great
  - more efficient reconstruction, e.g., Patatrack
  - fast sim
  - more efficient Matrix element calculation, e.g., MadGraph5 for GPUs
  - training of NN
- Experiments: we do not support GPUs because there are no in the Grid
- Grid sites: we do not provide GPUs because the experiments do not use them

Back in 2020

- Physicists: GPUs are great
  - more efficient reconstruction, e.g., Patatrack
  - fast sim
  - more efficient Matrix element calculation, e.g., MadGraph5 for GPUs
  - training of NN
- Experiments: we do not support GPUs because there are no in the Grid
- Grid sites: we do not provide GPUs because the experiments do not use them
- KIT: let's buy some GPUs and provide them to the Grid and the local HEP group

TOpAS at KIT

- Particle Physics Institute at KIT (ETP) wanted an analysis cluster at GridKa
- Throughput Optimized Analysis System (TOpAS)
- Agreement:
  - ETP pays the hardware
  - GridKa pays for power and cooling, maintains and administrates
  - ETP jobs have priority on their machines, GridKa jobs via backfilling with preemption
- 2020: first GPU machine with 8 NVIDIA V100 48 CPUs in TOpAS bought by ETP
- late 2020: three machines, each with 8 NVIDIA V100s, 192 CPUs, 1 TB
- 2021: three machines, each with 8 NVIDIA A100 40 GB, 192 CPUs, 1 TB

$\Rightarrow$ 56 NVIDIA datacenter GPUs available to the Grid since 2021

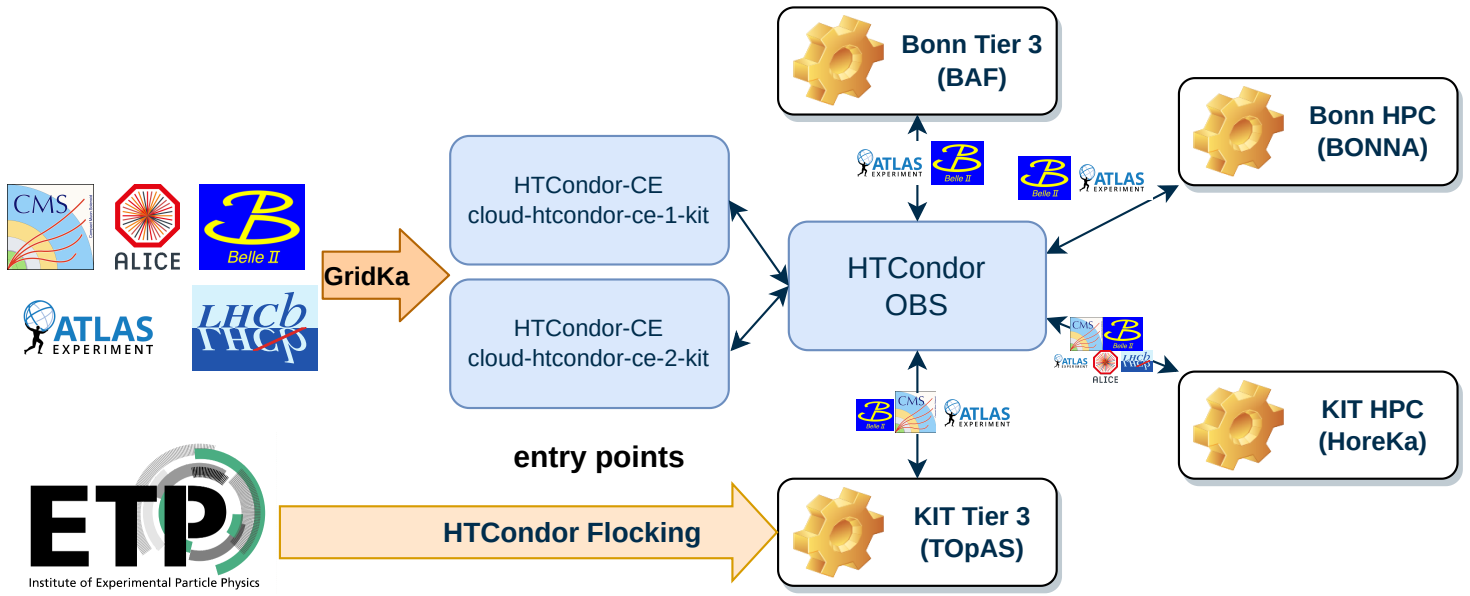
TOpAS at KIT

- Particle Physics Institute at KIT (ETP) wanted an analysis cluster at GridKa
- Throughput Optimized Analysis System (TOpAS)
- Agreement:
  - ETP pays the hardware
  - GridKa pays for power and cooling, maintains and administrates
  - ETP jobs have priority on their machines, GridKa jobs via backfilling with preemption
- 2020: first GPU machine with 8 NVIDIA V100 48 CPUs in TOpAS bought by ETP
- late 2020: three machines, each with 8 NVIDIA V100s, 192 CPUs, 1 TB
- 2021: three machines, each with 8 NVIDIA A100 40 GB, 192 CPUs, 1 TB

$\Rightarrow$ 56 NVIDIA datacenter GPUs available to the Grid since 2021

- -1 (we lost one a few weeks ago)

Provisioning of GPUs

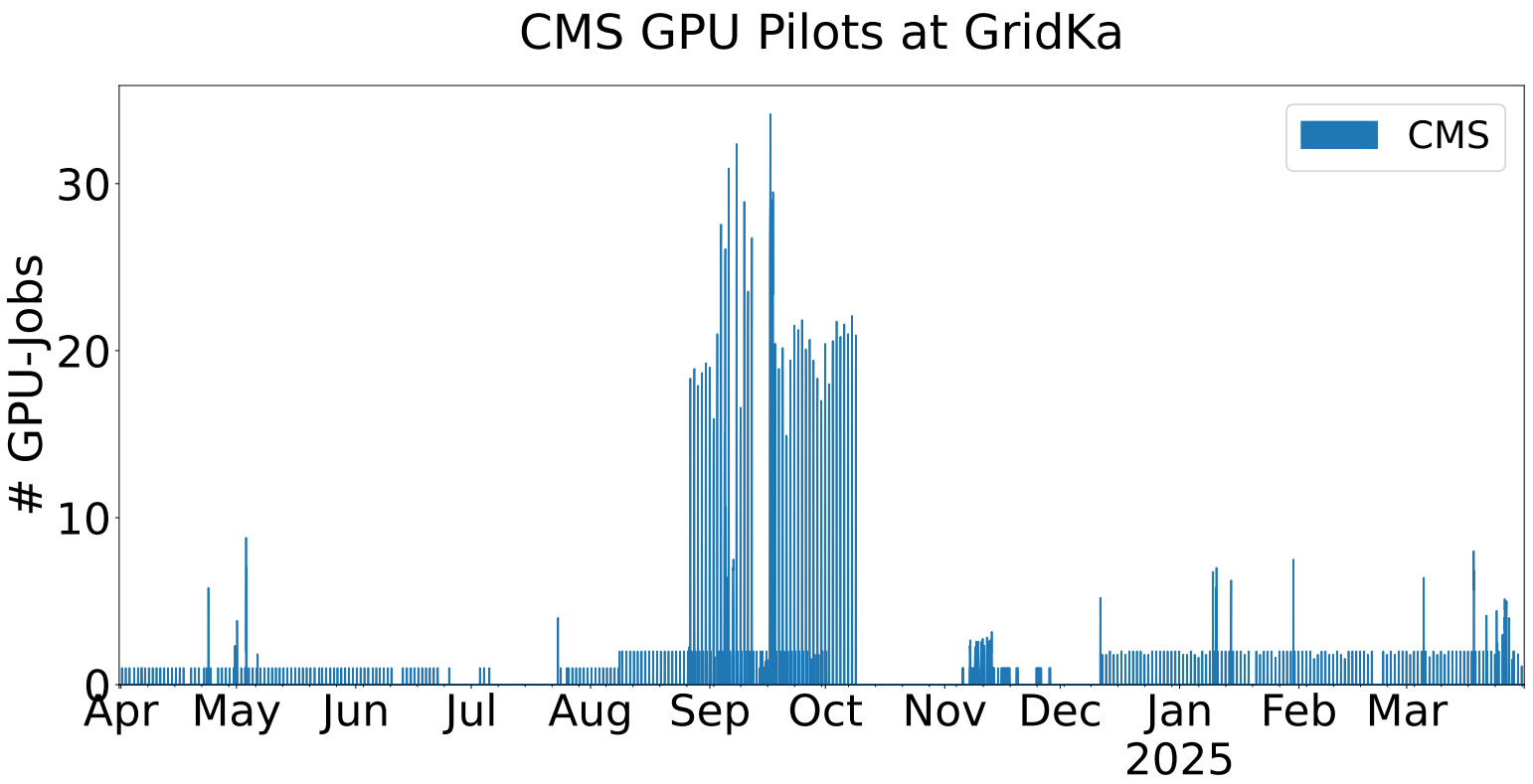


- TOpAS is a separated cluster (not part of the GridKa batch farm)
- ETP sends jobs from their condor system to the system of TOpAS via flocking
- resources are provided via the GridKa cloud to the Grid
  - integrated via drones (1GPU, 8 CPUs 20GB RAM)
  - nodes also accept CPU jobs
  - blocking of CPU and RAM for GPU jobs

Usage of GPUs

CMS Grid

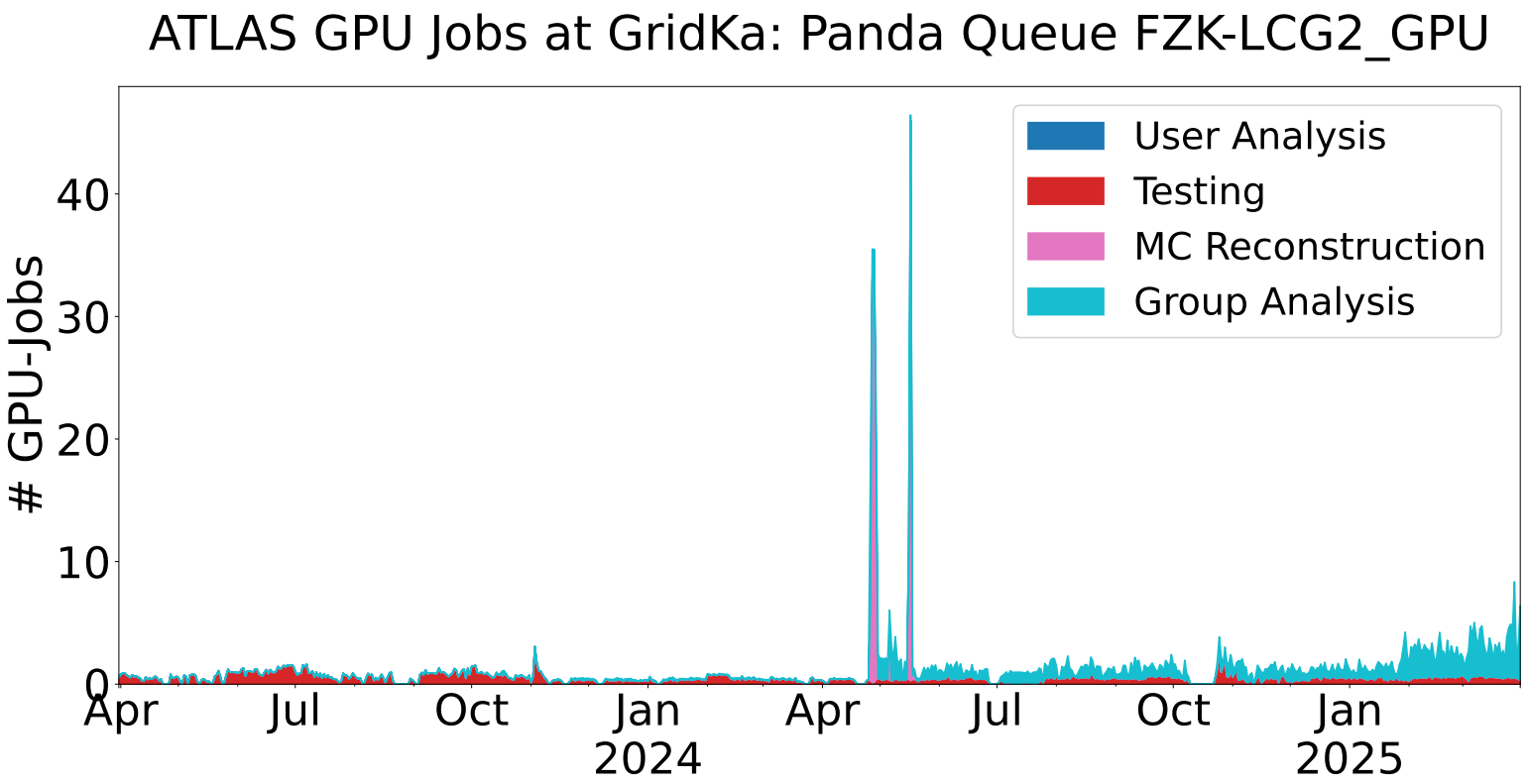
- about 2 test jobs per day since 2022
- got some release validation jobs for testing
- since last week, some user jobs



Usage of GPUs

ATLAS Grid

- ATLAS sends constantly testing jobs since Oct. 2022
- short tests with MC Production
- mostly used for analysis
- usage increasing slowly

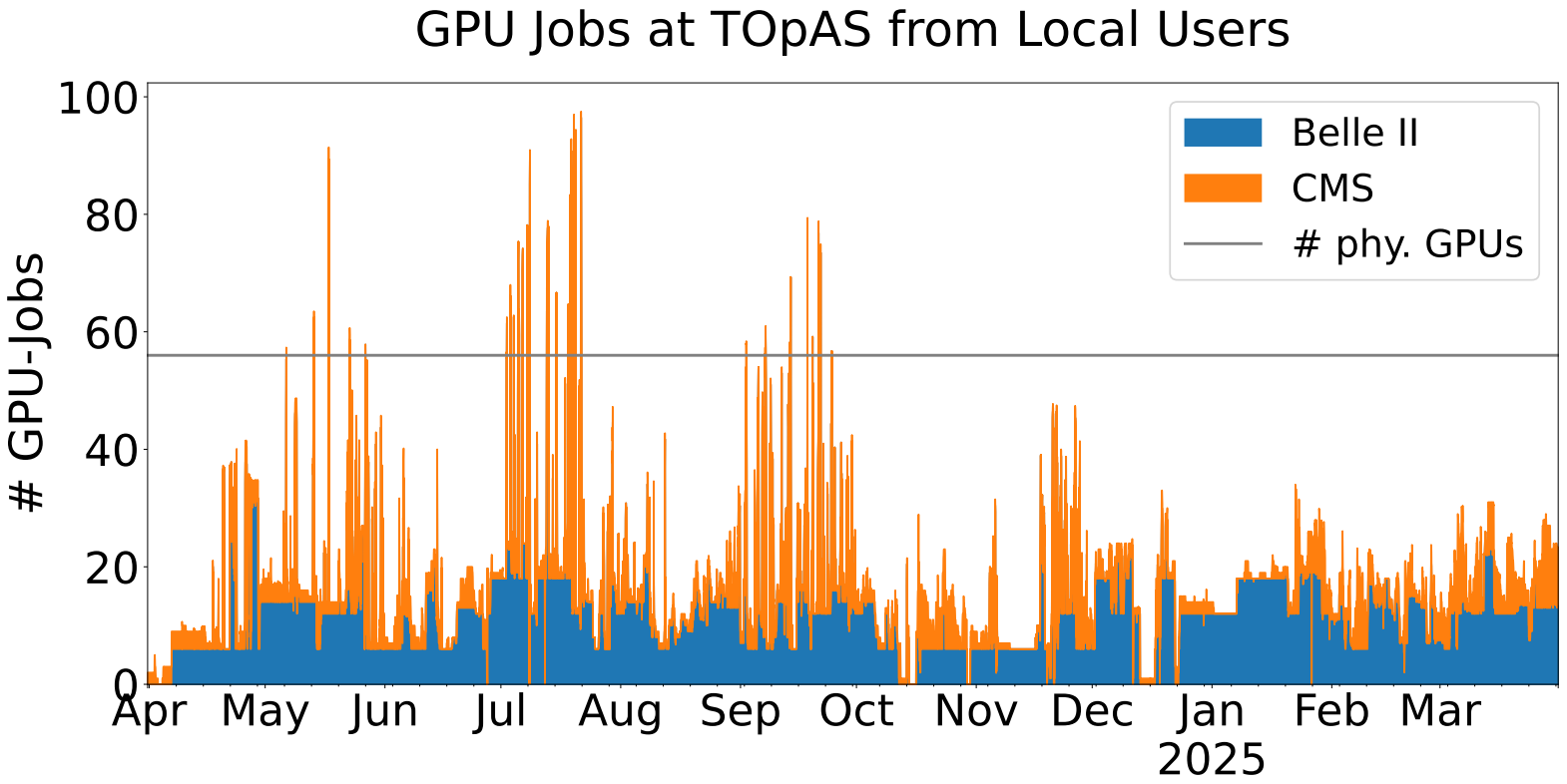


ATLAS Monitoring

Usage of GPUs

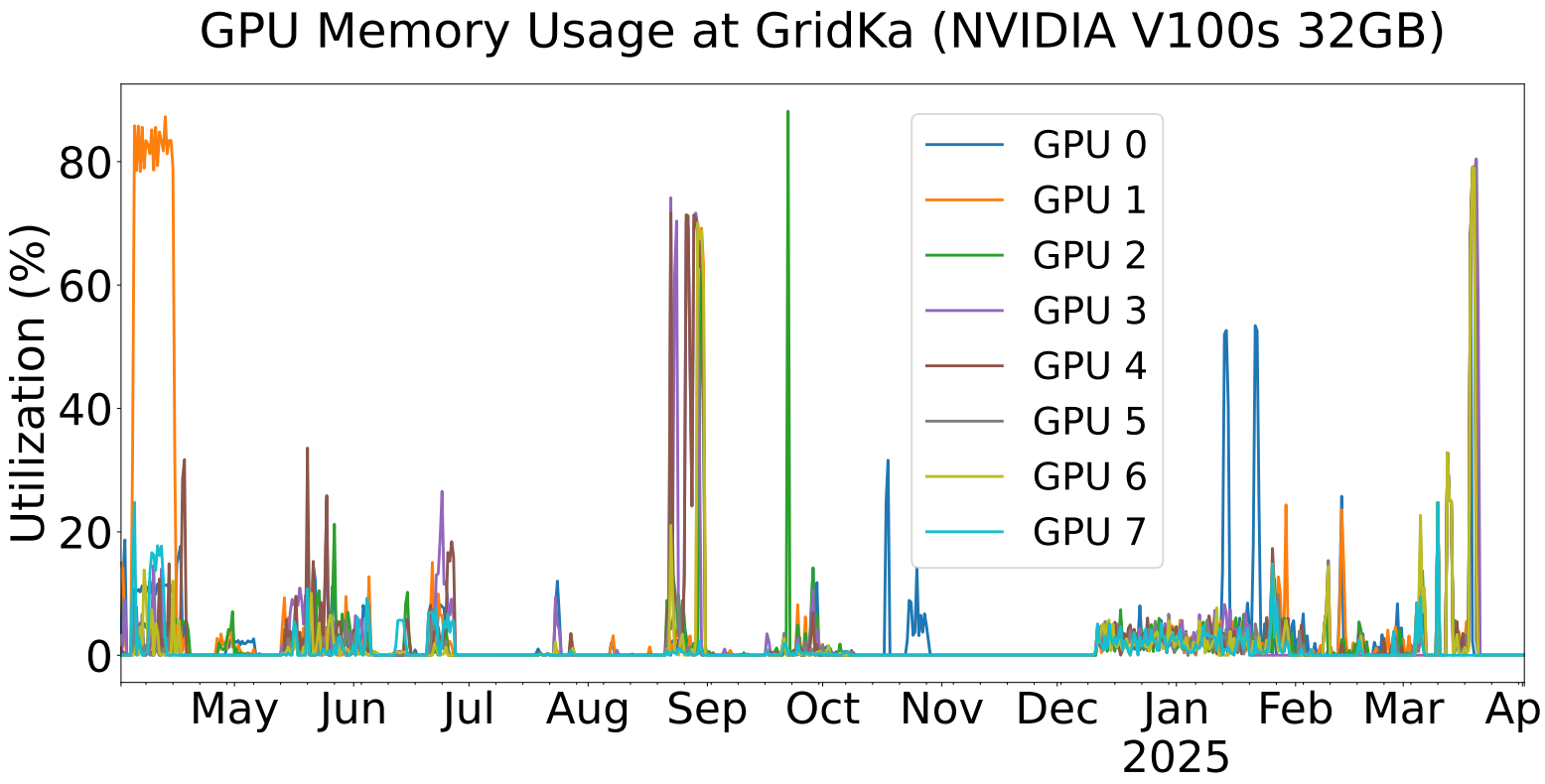
Local Users

- different kinds of workloads (mostly training)
  - hundreds of small (use 20% of a GPU) trainings
  - big trainings (6 NVIDIA A100 GPUs, 200 GB RAM, two weeks runtime)
- user and site gain experience
  - flexible slot size
  - data access via Grid protocol
- about 100 GPU jobs on 56 physical GPUs



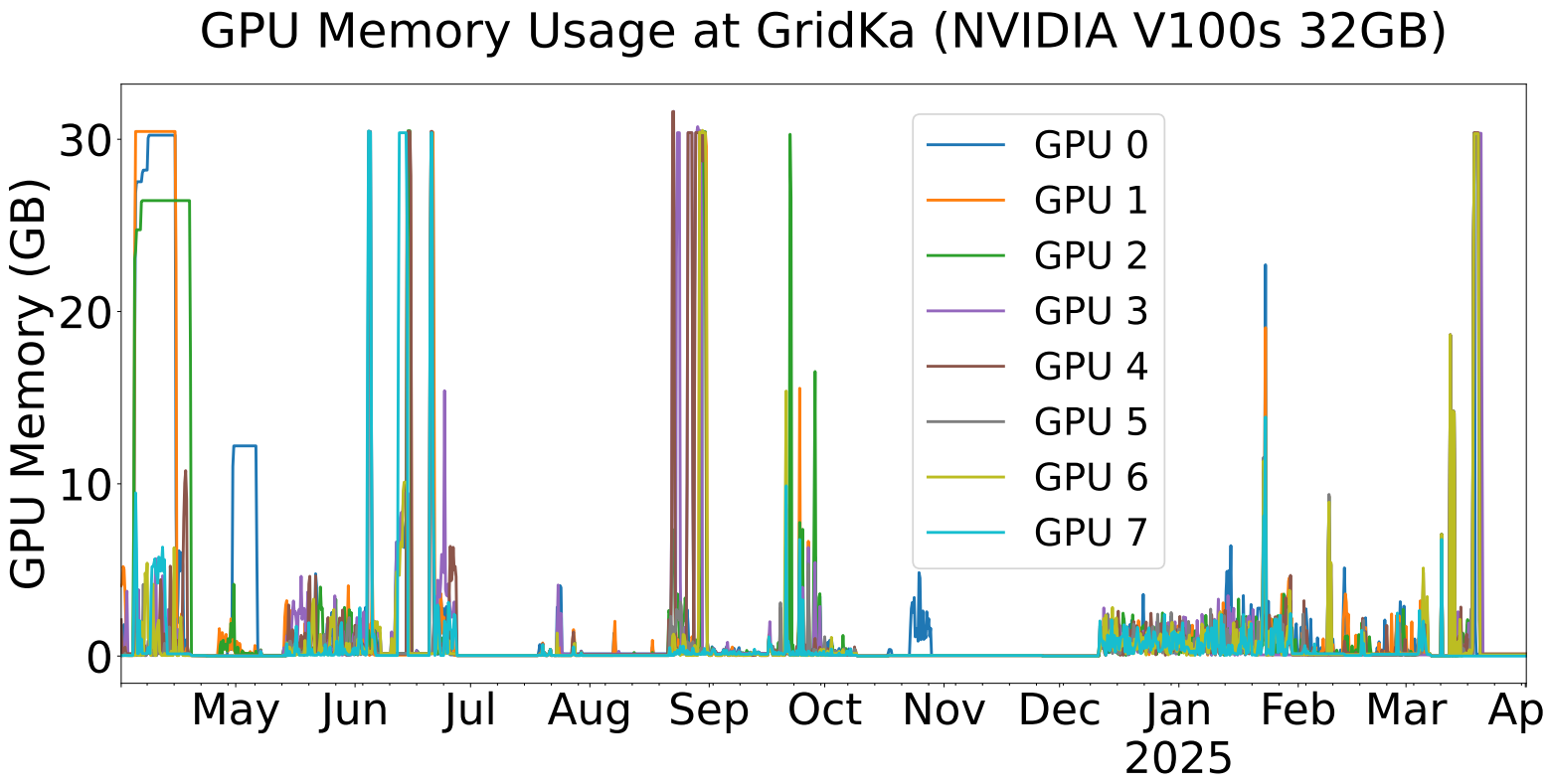
GPU Utilization

- several workloads with low GPU utilization
- out of GPU memory kills all GPU processes on the GPU



GPU Utilization

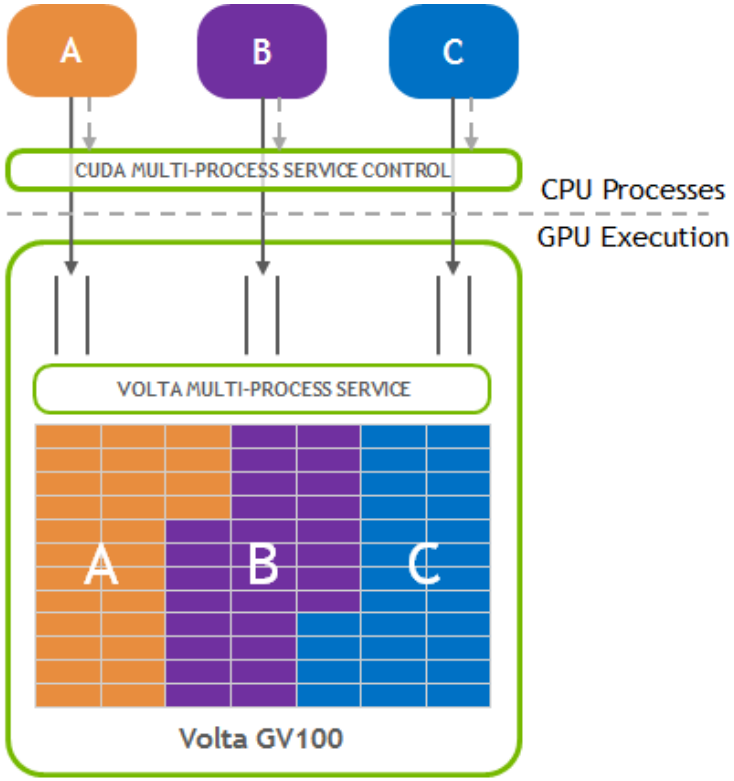
- several workloads with low GPU utilization
- out of GPU memory kills all GPU processes on the GPU
- several workloads with low GPU memory usage
- limit GPU memory per job to use several jobs on the same physical GPU



Multijob GPUs

- work from Tim Voigtlaender
- limit GPU memory
  - MIG (multi instance GPU) only available
    - + separation on the hardware level
    - - only NVIDIA GPUs since A100
    - - no flexible partitioning
  - MPS (Multi-Process Service)
    - + flexible memory limitation
    - - only CUDA
    - - single user per GPU

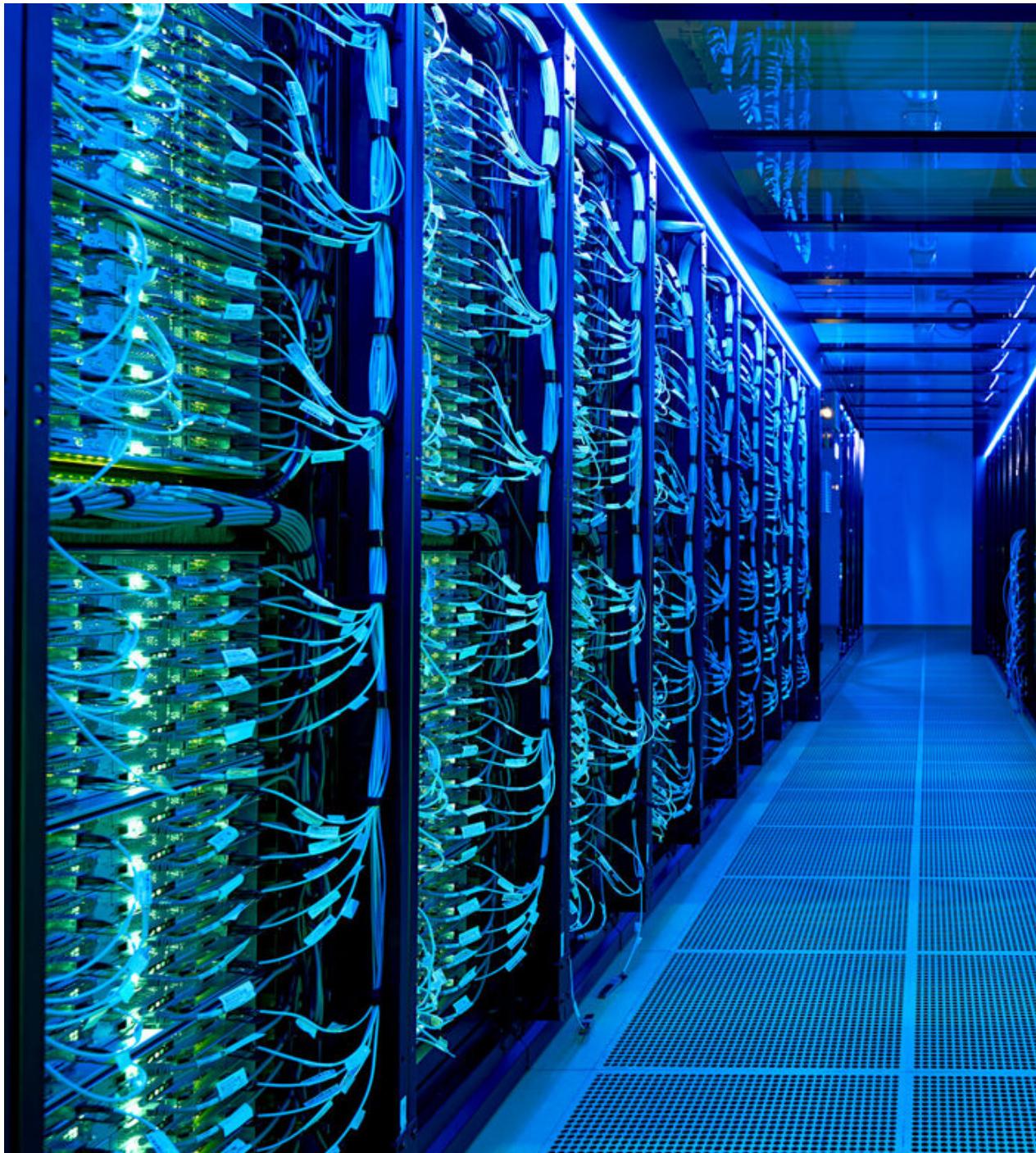
- use big GPU (NVIDIA A100 32 GB) machines for single and multi GPU jobs
- add extra HTCondor classad for the memory of sub-GPU jobs
- use smaller GPU machines for sub- and single-GPU jobs and run MPS
  - one MPS service per server
  - per physical GPU, one partitionable slot
  - CPU, RAM, and Disk is split between the partitionable slots
  - only processes of one Linux user can run concurrently per GPU
- low GPU computing power has lower priority



NVIDIA

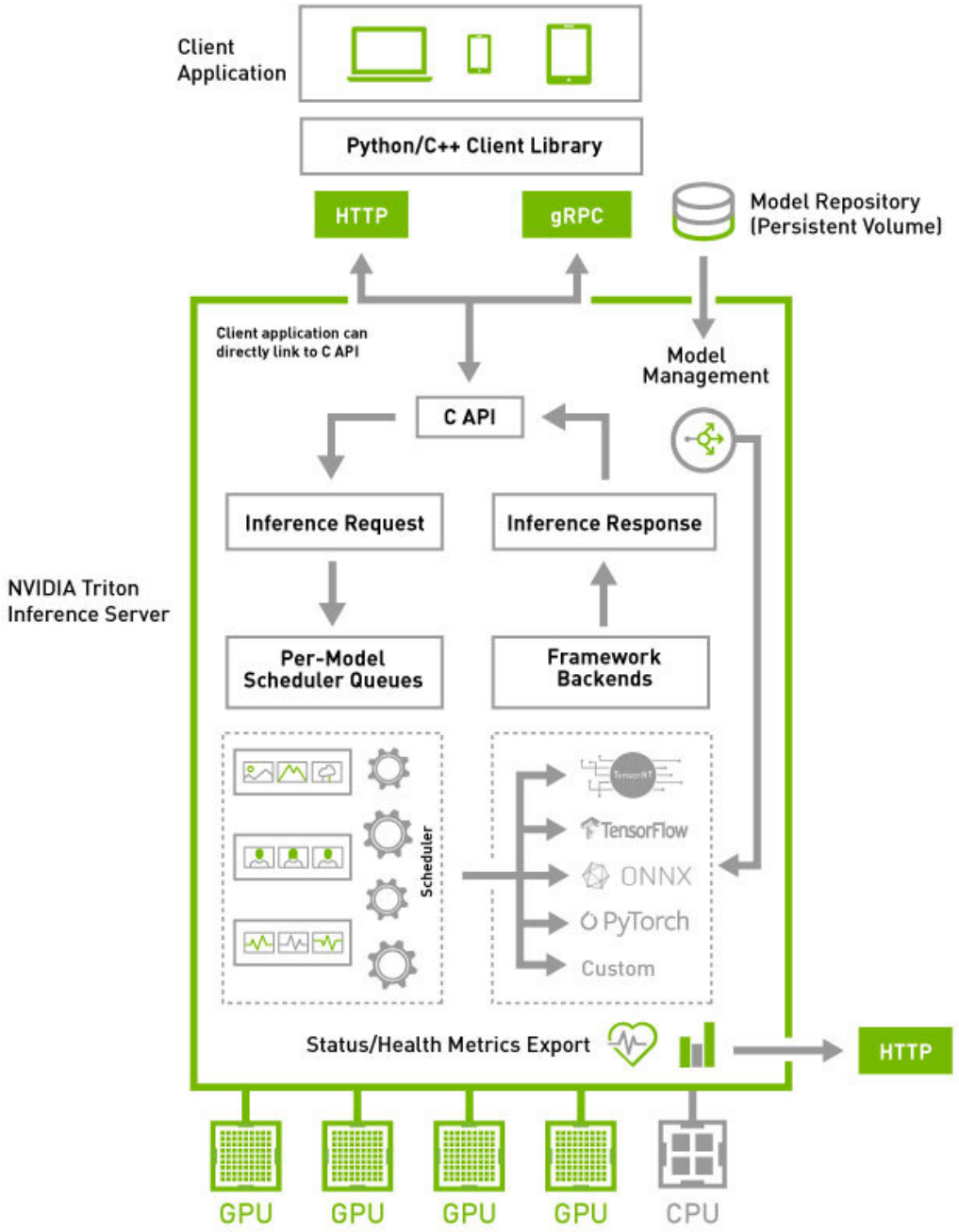
GPUs from HPC

- HoreKa: HPC Cluster at KIT next to GridKa
- integrate CPU resources from HoreKa already
- HoreKa provides
  - 668 NVIDIA A100 40 GB GPUs
  - 88 NVIDIA H100 95 GB GPUs
- dynamic integration of GPUs from HoreKa, similar to TOpAS



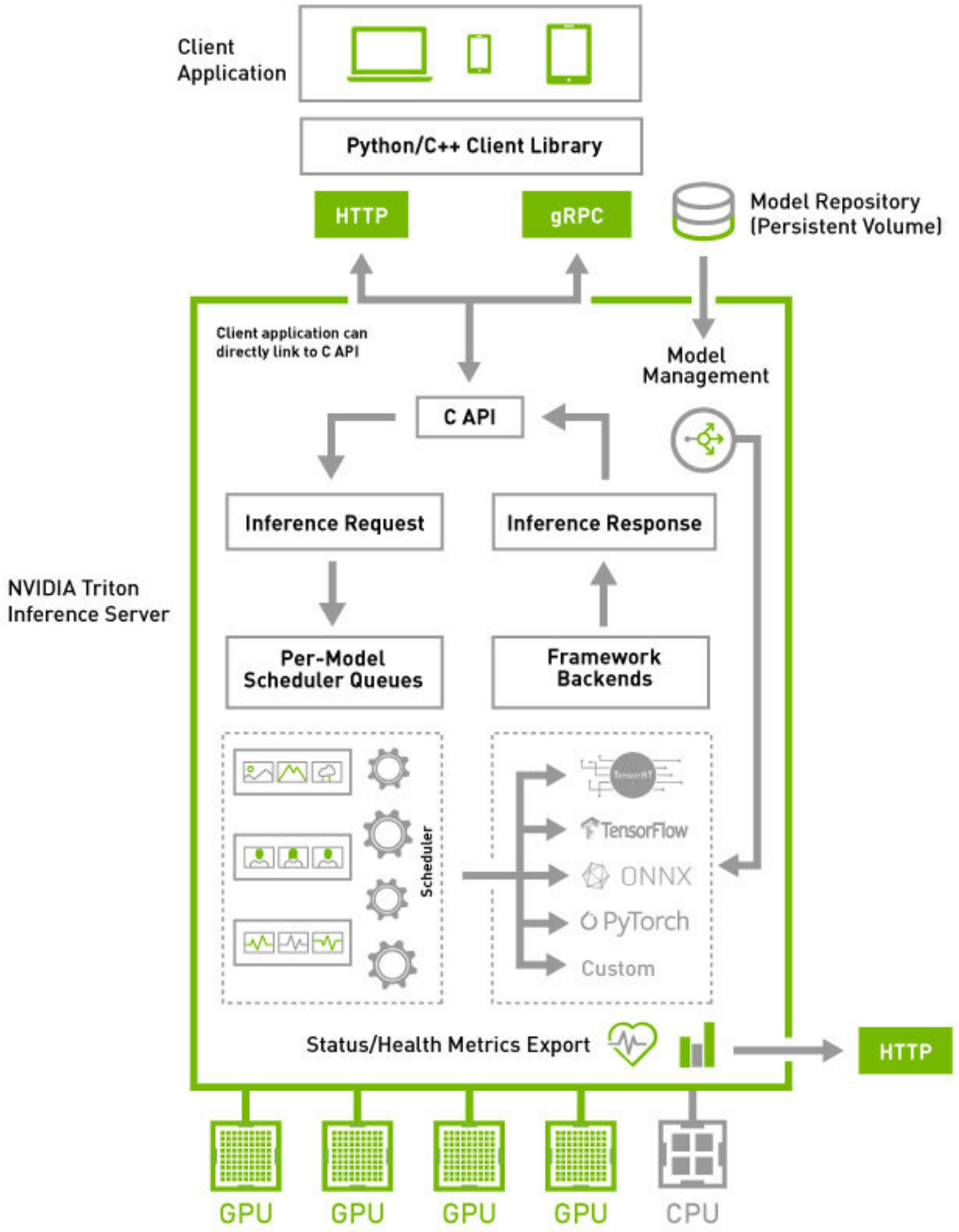
Centralized Inference

- upcoming idea
- several jobs run inference
- current HEP NN are small
- inference profit from GPUs
- idea of Triton by NVIDIA
  - service that runs on a GPU machine and accepts inference request
  - run several inferences concurrent for high GPU utilization



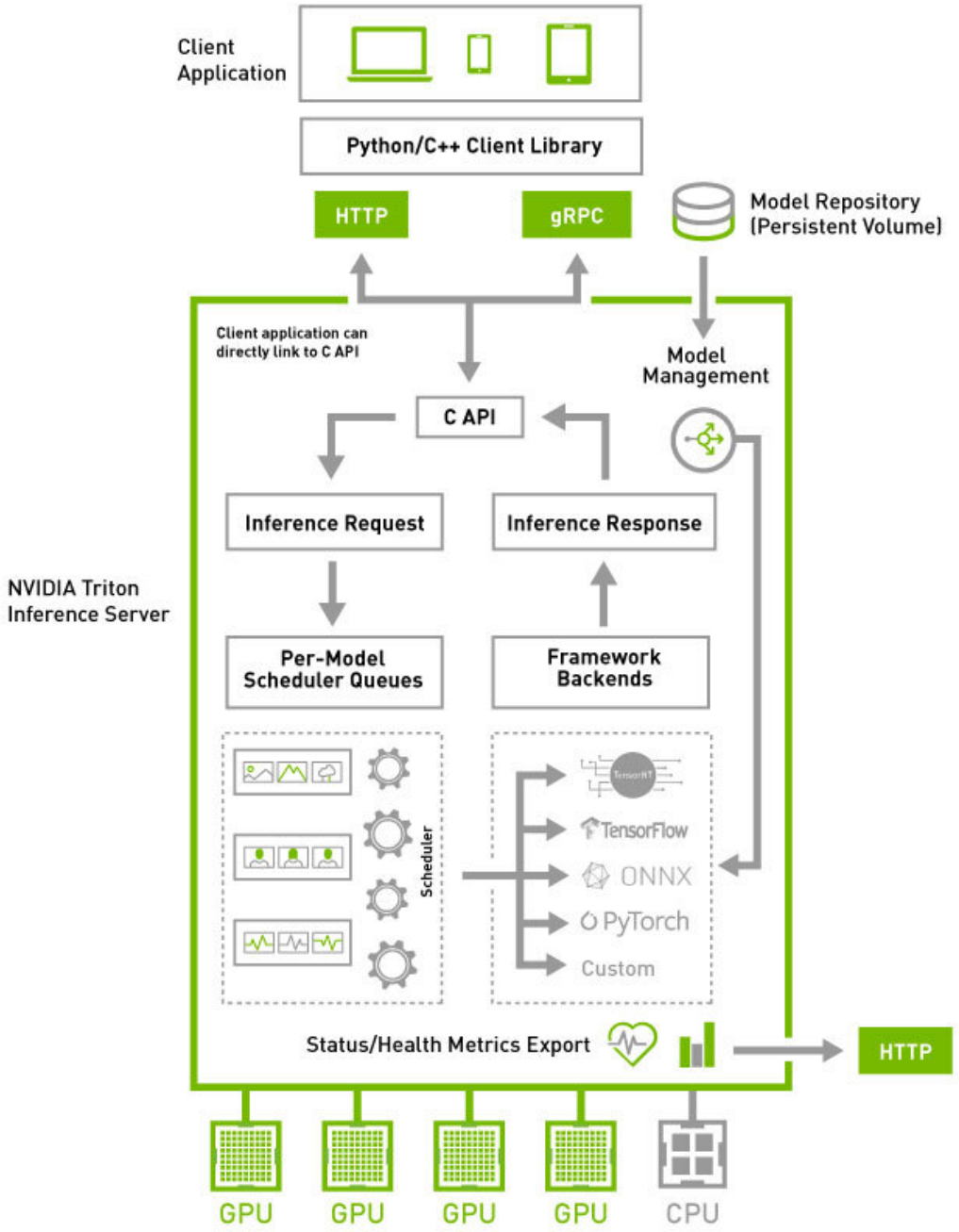
Centralized Inference

- upcoming idea
- several jobs run inference
- current HEP NN are small
- inference profit from GPUs
- idea of Triton by NVIDIA
  - service that runs on a GPU machine and accepts inference request
  - run several inferences concurrent for high GPU utilization
- operation model
  - how many servers? One per Site?
  - accounting?
  - sites without such a service?



Centralized Inference

- upcoming idea
- several jobs run inference
- current HEP NN are small
- inference profit from GPUs
- idea of Triton by NVIDIA
  - service that runs on a GPU machine and accepts inference request
  - run several inferences concurrent for high GPU utilization
- operation model
  - how many servers? One per Site?
  - accounting?
  - sites without such a service?
- Does someone have experience with that?



What I wanted to say

Experiments/Grid

- Experiments still begin to use GPUs
- Please test so we know what you and we need
- Is a GPU slot roughly defined? CPU, Memory, Disk?

End-User

- using GPUs more and more
- different-sized jobs (mostly small)
- GPU and system memory is important

GridKa

- looked into efficient usage of GPUs nodes
- GPUs are out of warranty; probably no replacement in GridKa
- looking to integrate GPUs from HoreKa on-demand

Personal View and Tips

- Do not buy the most powerful GPUs unless you have to
- Try to use GPUs from a HEP friendly HPC center if you like to provide GPUs
- Please test so we know what we should provide in the future

CMS tried to use GPUs GGUS#154188

- GGUS ticket 154188
- CMS complains about failing jobs on cloud resources
- jobs only fail on GPU machines
- reason
  - CMSSW 12.0 tries to use GPUs when detected
  - pilots run in container and cuda libs only bind mounted when GPU requested
  - pilots had no access to cuda-libs since on GPU was requested
- solution
  - send pilots that request GPUs
  - we did not see GPU pilots after that :-(