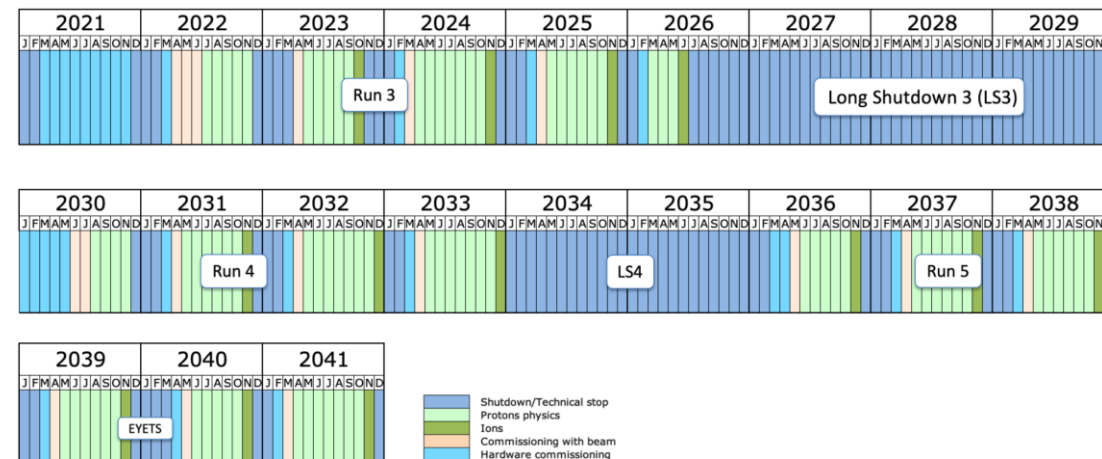


Report da "Heterogeneous Architectures in WLCG" CERN 3-5 December 2025

- **Took place at CERN**
 - **December 3-5, 2025**
- **Indico**
 - <https://indico.cern.ch/event/1550771>
 - [Detailed Live Note](#)
 - [Summary of the Live Notes](#)
- **161 participants**
 - **Experiments**
 - **Federations/Sites**
 - **WLCG management**
- **Goal**
 - Understand the current status of the code and workflows that can run on GPUs in WLCG
 - identify commonalities and differences
 - **estimate the fraction of workflows that can run on GPUs**
- (Re)discuss the needs of the experiments
 - how to evolve from today's small-scale use to scalable, production-level integration in WLCG
 - **including if and when pledges for GPUs could be accepted or required**
- Discuss on (agree?) the next steps towards a workplan for
 - **benchmarking, performance, cost effectiveness, provisioning, and integration of GPUs into WLCG**
- Next steps for the short (this coming year), medium (next 2-3 years) and long (ready for HL) term

Conclusion spoiler...

- The Workshop concluded that GPU acceleration is a vital R&D area across all major experiment workflows
 - Event generation
 - Simulation
 - Reconstruction
 - Machine learning
- However.....
 - no experiment is yet in a position to request or accept formal pledges of GPU resources from WLCG federations.
 - This status reflects the expected, ongoing nature of code development and R&D ahead of Run 4.
- It was noted that given that the process for requesting resources for the beginning of Run 4 (the first full year being 2031) begins a couple of years earlier, code bases and computing models are expected to be finalized by that time
- Defining firm resource needs requires the completion of several prerequisites, such as
 - the full integration of production workflows into the benchmarking suite
 - clearer finalization of computing models.
- Progress in these areas is foreseen and actively driven between now and the start of Run 4
 - With ATLAS and CMS foreseeing being able to offload 15% or less of their wallclock time of processing to GPUs by the start of Run 4



• Testing HEPscore:

- Experiments and Federations are testing the existing HEPscore for GPU benchmark suite
- focusing now on the GPU part of nodes
- **the ultimate goal is whole-machine benchmarking**
- Measurement of the relative performance of **different GPU vendors' models** seems possible even now, and may help guide initial GPU deployments by federations in early LS3
- This work is ongoing now and will hopefully continue

Setting Expectations

- ❑ No, we do not have an HEPscore23-GPU benchmark ready today
 - For all the reasons highlighted in this workshop
- ❑ Yes, we are on track proving that a HEP GPU benchmark is doable before Run4
 - As soon as experiment's GPU workloads are ready for Grid production activities

Preliminary HEPscore23-GPU Benchmark Results

- Additionally, able to run jobs on GPUs using this benchmark as payload in the Global Pool

What makes GPUs difficult

We have already mentioned issues with the availability of the various CUDA versions, and the difficulties that brings with scheduling and matching the workloads to the resources

- The old adage of "you have to use GPUs, as that's what's going to be available" is both inaccurate (there will always be CPUs) and ill defined (but which GPUs?) Also, cost/benefit...

Our impression is that the development cycle and therefore useable lifetime of GPUs is much less than CPUs, and that GPUs become antiquated or unsupported much sooner

- Sustainability is therefore a concern
- Contrast this to CPUs, where ATLAS regularly runs hardware more than ten years old

Measuring GPU performance is more difficult, again because of the many types, models

- Each have different strengths, and more variety than for CPUs
- Is one GPU slower than another because of real intrinsic differences, or because the test is unfair
 - Trend towards AI GPUs not always favourable for all our workloads
- Current model for AdePT involves **oversubscribing** the CPUs, not typical use of grid resources
- Nevertheless, it is vitally important to understand **what is in the future**

Few examples

- On one side, for simulations **LHCD plans to use certain calculations on GPUs and treat them as accelerators**
- **We do not fully know how to balance the CPU/GPU loads.**
- It means, IMHO, that heterogeneous nodes a-la HPCs will actually be needed
- Clearly, the processing efficiency will need to be calculated now.
- Training on ML is instead a rather "different beast". On one side higher GPU efficiency, but:
 - it is unclear if we'll even need to run ML "on the Grid" (how to train)
 - at first it does not seem to fit in our model of jobs-with-input-output

Practical lessons from WLCG runs: 2 GB / core limit

- In AdePT, the experiment framework-specific G4 code is called exactly like on CPU
- Copy back steps from GPU to CPU asynchronously
 - 100s of GB!
- We must start the GPU if the buffer is overflowing
 - optimal performance if full GPU memory is used for buffer
- CPU memory for transfer allocated in pinned memory
 - ~ 20-30 GB RAM per GPU
- Running a job on 8 cores + 1 GPU with a 2 GB / core limit (16 GB total) is suboptimal!
- **Also, AdePT is resource-hungry and should not share the GPU**

04/12/2025 2

Summary

- ❑ Establishing a HEP GPU benchmark is one of the multiple key components for the adoption of heterogeneous resources in WLCG
- ❑ HEPscore4GPU will be built on the experience on HEPscore for CPUs
 - Long-term goal: have a whole server benchmark (CPU + RAM + GPU + storage)
 - The production version of HEPscore4GPU will be based on Run4 production apps
- ❑ Meanwhile, we are building a demonstrator benchmark to enable the community to gain early experience



Experiment Readiness



Summary

- ALICE continues to improve its O2 software where it heavily employs GPUs to speed up online and offline processing.
 - 99% of online reconstruction on the GPU (no benefit from porting more) with smooth operation since 2022.
 - Today 50%-60% of full offline processing on GPU yielding 2x to 2.5x speedup.
 - Will increase to 80% with full barrel tracking (optimistic scenario), aiming for 5x speedup.
 - Successfully validated ITS tracking on GPU and GPU usage in the GRID at NERSC, aiming to deploy next year.
- Advent of AI/ML based solutions:
 - Cluster finder implemented into reconstruction workflow with goal to have it production ready in Run 4
 - Low momentum neutron "looper" simulation based on generative ML is already in production
- The code base for the non x86 architecture has been published and validated for both analysis and very recently MC simulations
- The JAliEn middleware is ready to handle job submission to both CPU (including Aarch4) and GPU enabled compute clusters

Roadmap towards a HL-LHC GPU strategy

- Continue **offloading more code onto GPUs**.
 - Training and creating new expertise is critical.
- Assess **workflow-specific GPU needs**.
 - Classic algorithm offloading or ML inference?
 - What are the precision requirements (reduced precision for ML vs. high precision for simulation/reconstruction)?
- Evaluate **GPU needs for core production and ancillary activities** (e.g., ML training, analysis).
- Develop a **framework for mixed hardware environments** (CPU + GPU) to ensure smooth operation across heterogeneous sites.
 - Improve **GPU resource management and scheduling** to optimize utilization across diverse workloads.
- Conduct comprehensive **benchmarking studies** to evaluate GPU utilization and derive realistic resource requirements.
 - **Integrate GPU benchmarks** into the WLCG pledging and accounting processes.
- Perform a **cost-benefit analysis** to understand what GPU resources should be pledged vs. acquired opportunistically.

Current hardware trends favor GPUs with lower-precision cores

Many of these efforts will require close collaboration with WLCG and external partners, in addition to dedicated work within CMS

LHCb Summary

| | LHCb software readiness | resources readiness (availability to LHCb users) | distributed computing (Dirac+X) |
|---------------------------|--------------------------------------|--|---------------------------------|
| ARM CPUs | Ready | via WLCG | Ready |
| GPUs for Sim accelerators | Ongoing. Needs double-precision GPUs | LHCb-owned, or via HPCs (opportunistically) | Ongoing |
| GPUs for ML | Ongoing | LHCb-owned, or via clouds | A very different paradigm |
| TPUs/NPUs/FPGA | Specific applications, online | LHCb-owned | Not planned |

Summary



- **Readiness and timeline of your offline codebase for running on WLCG heterogeneous resources (including reconstruction)**
 - ARM is fully supported for both reconstruction and simulation. Expect fullsim with GPU by Run 4. Significant progress with reconstruction + analysis (significant overlap with offline and online/NGT codebase); much of it can technically run using GPUs
 - ML likely to be run in parallel to more conventional approaches at start: traditional approaches for commissioning new detector - need to be able to see intermediate reconstruction steps.
- **Offload expectations: even with uncertainty, what fraction of your workload do you anticipate could benefit from GPU (or other accelerators)?**
 - ATLAS workflows dominated (30%) by FullSim, where we estimate ~50% can be offloaded i.e. 15% of our total compute workload could benefit from GPUs. Not considering FPGAs.
- **Integration with production systems: your current or planned readiness for submission of GPU-enabled workflows to the grid within your production frameworks.**
 - ATLAS has been able to submit to grid GPU queues for years, but active efforts to improve this with more focus on production workloads; monitoring is a significant problem
- **AI workflows: do you foresee integration of AI/ML workflows (e.g. training, inference, hybrid tasks) within your standard production or analysis frameworks?**
 - Training is likely to remain dedicated operation. Inference is already routinely used in ATLAS today

Experiment Readiness - 2

- **Offline Experiment Code for GPUs:** Experiments are making updated code available for inclusion in the development version of HEP Score for GPU as it becomes available
- [We agree that we should remain open to supporting different vendors' GPU models e.g. AMD.](#)
- **Open Question:** discussion on the expectation towards facilities to enable the ~15% of GPU offloading by the experiments?
 - how much is the 15% dependent on what facilities provide *during the time* leading up to Run4?
 - this is to get an idea of what sites need to plan for in terms of investment in hardware, platforms, or services during LS3 to enable the offloading

More Open Questions

- What are the **resource needs for ML training** and do they need to be provided by specialized ML training facilities or can HPC allocations, university clusters, or national platforms fulfill these needs?
- Quantify (when possible) what GPU capacity will be available on experiments' (new or previous) **HLT farms and their availability for offline use?**

ARM @ATLAS

- Standard batch cluster behind a standard CE
- All the ATLAS dependencies are built for ARM, lots of contributors
 - cvmfs, rucio client(gfal), el9 container, prmon
- Subset of SW production releases built and validated for ARM
 - G4 sim & reco mean we can keep resources full with production
 - a task runs on mix of architectures (x86_64|aarch64)-el9-gcc13-opt
 - Evgen is built, but Gridpacks more challenging, due to arch dependent component
- Analysis SW built, but not used transparently for users
 - needs 2 user SW builds, or auto-build
- Sites can pledge up to 50% ARM cpu
 - non-ARM builds, including analysis, still needs some x86_64 near the storage
- Sites can get competing quotes for ARM or x86 hardware

I promised a slide on aarch64

- lxplus-arm.cern.ch exists
- We have 2240 aarch64 cpus in LxBatch
 - They are always full
- aarch64 nodes can be requested via OpenStack
- Here endeth the information about aarch64, it is for us an unexciting, small, and easy to support section of the batch farm
- It feels like we don't need to really do much more with aarch64 at the moment, but shout if this is not the case

@CERN



04/12/2025

Ben Jones | GPUs @ CERN

12

In general....ARM is a solved case


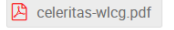
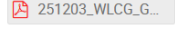
Software Readiness

14:00 → 15:30 Progress of Experiments and Software Projects: Session C

Progress of experiments and software projects, and how they will interact with a heterogeneous WLCG environment:

- * Experiments
- * Common Software (SFT, NGT)
- * Simulation
- * Generators

Conveners: Alessandro Di Girolamo (CERN), James Letts (Univ. of California San Diego (US)), Stefan Roiser (CERN)

| | | |
|-------|---|-------|
| 14:00 | Adept detector simulation | 🕒 15m |
| | Speaker: Severin Diederichs (CERN) | |
| |  | |
| 14:30 | Celeritas detector simulation | 🕒 15m |
| | Speaker: Seth Johnson (Oak Ridge National Laboratory (US)) | |
| |  | |
| 15:00 | Root and GPUs | 🕒 15m |
| | Speaker: Danilo Piparo (CERN) | |
| |  | |

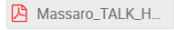
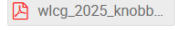
15:30 → 16:00 Coffee break 🕒 30m

16:00 → 17:30 Progress of Experiments and Software Projects: Session D

Progress of experiments and software projects, and how they will interact with a heterogeneous WLCG environment:

- * Experiments
- * Common Software (SFT, NGT)
- * Simulation
- * Generators

Conveners: Alessandro Di Girolamo (CERN), James Letts (Univ. of California San Diego (US)), Stefan Roiser (CERN)

| | | |
|-------|---|-------|
| 16:00 | Madgraph event generation | 🕒 15m |
| | Speaker: Daniele Massaro (CERN) | |
| |  | |
| 16:30 | Pepper (Sherpa on GPU) event generation | 🕒 15m |
| | Speaker: Max Knobbe (Fermilab) | |
| |  | |

- **Understanding Resource Requirements**
 - In the context of the broad classes of workflows:
 - “traditional” GPU calculations
 - ML inference
 - ML training
 - and physics analysis
 - Understanding the memory and CPU-to-GPU ratio requirements
 - performance and scalability of different deployment or provisioning models
 - “traditional”
 - IaaS
 - floating-point precision needs.
 - Note that the different GPU provisioning models were not seen as contradictory, especially as they may serve different use cases, and when considering eventual large physics models (LPMs) which were seen to be too big to be hosted on single nodes. It was also discussed that we have to be very careful in not relying on opportunistic resources (like HPC) for critical activities.

- **Workflow Management: R&D** needed to evolve WM systems to consider efficiently co-scheduling CPU and GPU workflows
 - According to the application
 - simulation, reconstruction, but also inference and training
 - handling partitionable GPUs (such as with MIG) and checkpointing (e.g, training and HPO)
- **WLCG Information System**
 - Discovery and advertising of heterogeneous resources as well as an agreed-on description schema was seen as needing a proposal for an improved WLCG information system.
 - The inherent heterogeneity itself (GPU models, CUDA driver versions, etc.) was seen as a challenge with respect to job failures, compilation difficulties, portability of code bases, etc.

- **Monitoring:** Understanding how to improve monitoring the occupancy and performance of GPUs.
- **Accounting:** Based on improved monitoring, Improving the accounting of GPU performance and utilization

Cost Evaluation

- **Cost Estimates:**

- it would be useful to establish shared cost predictions for GPU
- could be done in the context of the HEPiX TechWatch working group.
- to guide WLCG federations in optimizing multi-year funding requests and procurements

- **Total Cost of Ownership:** It is important to understand the “total cost of ownership” of GPUs

- power consumption,
- GPU lifecycle,
- failure rates
- embedded carbon.

- While not discussed during the Workshop, the work of both the WLCG Sustainability Forum and the HEPiX TechWatch Working Group could be very important here.

- **Longer term concerns**

- GPUs aging out (software support, performance, etc)
 - What are the GPUs life cycles and what implications are there for costs over time
- Newer GPUs may need more infrastructure support (power, cooling)
- Difficulty to scale up user support (for AF and other ways to access/use GPU)

Dirk Hufnagel, December 4, 2025

Heterogeneous Architectures in WLCG - GPU Experience and Plans in U.S.CMS

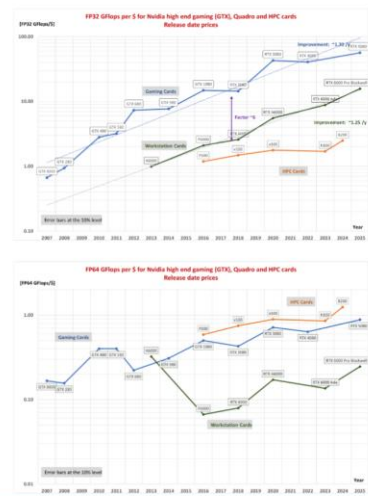
GPU cost trends

Historically, price/performance has always gone down

However, strong AI-driven demand and memory price is causing an increase of costs

At this stage, making predictions on the cost evolution is extremely difficult

- If there is an “AI bubble”, it might take years to burst



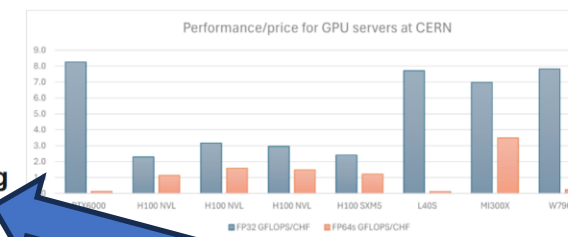
GPU cost trends

In addition, new GPUs are being optimized (only) for ML

- FP32/64 performance not increasing much, exacerbating the problem

Hopefully O(10%) price variations do not change the conclusion about whether using GPUs for our workloads is cost-effective

- Unless of course the savings are modest to begin with
- Different models can be enormously different in performance/cost



From the CERN talk

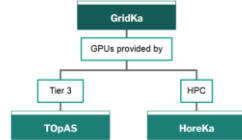
From the Sites

HPC made easy: HPC Bubbles

- Cloud-based HPC cluster instantiated via user-friendly interfaces
- Intermediate step between the Cloud/users workstation and extreme-scale supercomputers

Now @ GridKa / KIT

- Experiments:
 - No GPUs in the Grid → no need to support them
- Grid sites:
 - No support of GPUs → no need to provide them
- But:
 - Everyone thinks we can benefit a lot from GPUs



21/5 04/12/25 Nikita Shadskiy - Experience in Germany (GridKa/KIT)



- we need a way to pledge GPUs - it all comes down to money”
 - It was discussed if having a (good) accounting of the resources could be enough for the next FAS’ funding round
 - if it comes in the next couple of years
- GPUs available in several sites, how to access them is still not an easily solvable problem - different solutions
- In sites like e.g. Manchester there are GPUs available on the Grid since 2018, but only occasionally used - users prefer accessing GPUs locally. Similarly for KIT
- We observe that there is still a gap between the experiments needs - as of now they don’t see the need to pledge GPUs, the estimate of required GPUs for Run4 is relatively small (topping at 10-15%), the facility observations - GPUs are available, there is spiky usage, but overall the usage is not more than 50%, and users - maybe we hear only the vocal ones but there is still the impression that there are not enough GPUs available. - we are working to define a plan to make these views closer together

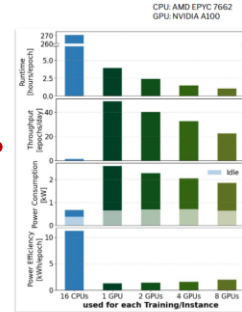
Outlook

UK

- One of the biggest pools of production ARM resources in WLCG.
- GPU usage still dominated mainly by Online, User development and AI/ML workflows.
- Interactive access preferred over GRID
- Need for production workflows, benchmarking (HEPSpec equivalent) and WLCG pledge mechanism for GPUs
- Important development activities across UK Cloud:
 - At RAL Tier1: Opportunistic vGPU queues from STFC Cloud, scalable as per the requirement and load.
 - At Glasgow: GRID Federated ML studies and additional GPU based HEP workloads: [Check Bruno's talk for details](#)
- Sufficient non-x86 resources to meet the current requirements with capabilities to scale up resources as per the WLCG/user needs.

Use-cases on GPUs (TOPAS)

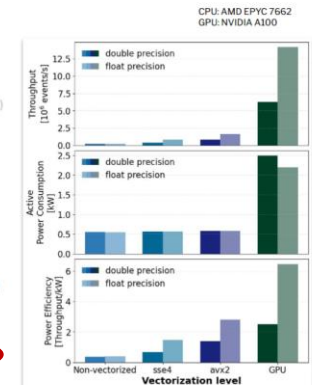
- Heavyweight ML
 - Example: Tau Transformer training (tau lepton identification)
 - Large training datasets (~ 96 million events)
 - Models large / complex enough to fully utilize GPUs
 - More GPUs → faster runtime for an individual training, but smaller throughput overall
 - Same or lower power consumption vs. efficiency



Source: Tim Vogtlander

Use-cases on GPUs (TOPAS)

- Heavyweight ML
 - Example: Tau Transformer training (tau lepton identification)
 - Large training datasets (~ 96 million events)
 - Models large / complex enough to fully utilize GPUs
- Lightweight ML
 - Example: analysis event classification training
 - Training datasets (~10 -100) smaller
 - Simpler models utilize only a parts of a GPU
- Event simulation
 - Example: MadGraph4GPU matrix element calc. (gg → ttg)
 - Comparison to vectorized CPU usage
 - Float precision with better power efficiency
 - CPU vectorization with potential



Source: Tim Vogtlander

12/15 04/12/25 Nikita Shadskiy - Experience in Germany (GridKa/KIT)



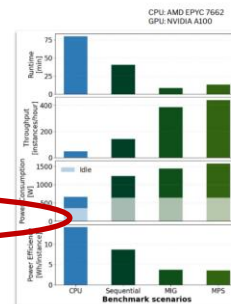
Common issues: low usage

- Mostly used by local experts (<10 per site)
 - Site admins don't see high demand
 - Lots of idle GPU
- AFs have better user occupancy
 - Hard to monitor what the GPU are exactly used for
 - ML training? Analysis? GPU algorithm development?
- OTOH, users often ask for more GPU
 - There is a contradiction here
 - Should be solvable with training
 - Recognized problem and put effort on fixing it



Use-cases on GPUs (TOPAS)

- Heavyweight ML
 - Example: Tau Transformer training (tau lepton identification)
 - Large training datasets (~ 96 million events)
 - Models large / complex enough to fully utilize GPUs
- Lightweight ML
 - Example: analysis event classification training
 - Training datasets (~10 -100) smaller
 - Simpler models utilize only a parts of a GPU
 - Significant increase in throughput using MIG/MPS with similar power consumption



Source: Tim Vogtlander

Report

11/15 04/12/25 Nikita Shadskiy - Experience in Germany (GridKa/KIT)



- **Education and Training:**
 - Providing training and education to enable better matching of user needs (e.g, interactive GPU access) with initially deployed resources
 - documenting how to use GPU resources on HPCs for training ML models with non-experiment-supported workflow submission systems