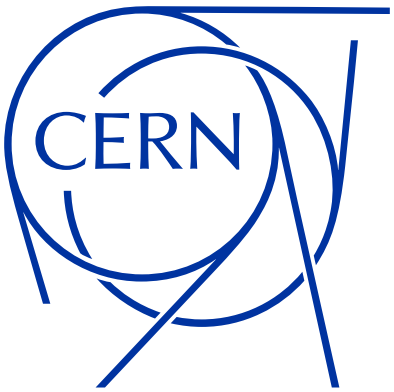
Exploration and Evaluation of non-LHC GPU Benchmarks for HEPScore4GPU

September 2026

AUTHOR:
Gracjan Adamus

AGH University of Kraków

SUPERVISORS:
Robin Hofsaess
Natalia Diana Szczepanek



PROJECT SPECIFICATION

The integration of GPU workloads into HEPScore follows the same philosophy as for the CPU ones: The benchmark should consist of real-world HEP applications for being as representative as possible for the WLCG environment.

Although GPU applications in HEP are on the rise, the current preliminary HEPScore4GPU benchmark mainly consists of development versions of the experiment's applications. Therefore, general purpose GPU benchmarks, if available, could be an interesting intermediate solution for assessing and comparing the performance of GPU servers for the WLCG. This would allow progress during the present period in which GPU workflows from HEP are stabilizing and are not yet appropriate for a final benchmark.

Since GPU-applications are fundamentally different to CPU, it will also be an interesting test case for the long-term comparison to HEP applications, as well as a valuable cross-validation, eventually even as an extension to the HEP-only benchmark in the future.

The project focused on the following objectives: At first, research and preliminary implementation of existing open-source GPU benchmarks with the aim of obtaining an overview of the current landscape and understanding their design, scope, and usage models. After that, gaining practical experience with selected benchmarks in order to assess their functionality, requirements, and limitations. And lastly, evaluating the relevance and applicability of these benchmarks for our use cases.

ABSTRACT

The increasing use of GPUs in High Energy Physics (HEP) workflows requires an extension of HEPScore, the Worldwide LHC Computing Grid's (WLCG) standard benchmark framework. While the CPU performance of WLCG resources is quantified by HEPScore23, an analogous benchmark for GPUs, HEPScore4GPU, is under development and will be following the same philosophy by incorporating real HEP applications. However, GPU-accelerated experiment workflows are still maturing, creating a practical gap. General-purpose GPU benchmarks could therefore serve as an intermediate solution for assessing and comparing GPU servers for the WLCG today.

This report explores and evaluates non-LHC GPU benchmarks as candidates for HEPScore4GPU. Starting from the suite used by the EuroHPC JU in the procurement of the Leonardo supercomputer, three scalable applications with mature GPU support (SPECFEM3D Globe, Quantum ESPRESSO, and MILC) were selected, containerized, and executed through the HEP Benchmark Suite for automated execution and data acquisition, and evaluated. Each application was initially characterized on an NVIDIA A100 through profiling of GPU utilization, power draw, and memory usage, and validated with 40 repeated runs using the coefficient of variation to measure their reproducibility. SPECFEM3D Globe was selected from the above and executed on four NVIDIA GPU architectures and, via its HIP port, on an AMD GPU.

The results show that all three applications produce reproducible time-to-solution values, but differ markedly in how they utilize the GPU. Quantum ESPRESSO shows a variable, mixed CPU/GPU pattern and MILC leaves the GPU idle for a substantial fraction of the run. Only SPECFEM3D Globe keeps the device fully saturated, indicated by a power-draw close to the maximum and a variation of 0.4 %. The throughput was very stable with a variation of 0.02 %. Its runtime scales with hardware capability, from 314 s on the A100 to 8 000 s on the AI inference-oriented L4, which is explained by the FP64 performance of the cards. SPECFEM3D Globe, therefore, emerges as the most suitable workload for an intermediate benchmark among the tested applications, combining full utilization, excellent stability and portability across vendors. The report closes with an outlook toward a production-ready HEPScore4GPU: extending the candidate pool, separating benchmarks by floating-point precision, and profiling and classifying the available HEP GPU workflows.

ACKNOWLEDGMENTS

I want to sincerely thank my great supervisors, Robin Hofsaess and Natalia Diana Szczepanek. Working with you was a delight, and your support throughout my whole stay at CERN was substantial and highly appreciated.

I also want to thank all the great people I met in the office, from both the LCG section and the CE group. Meetings and lunches with you were always a pleasure.

And of course, I want to thank CERN for the opportunity to be part of the CERN Summer Student Programme, and to spend the whole summer in a place of academic excellence!

Lastly, I want to thank CERN's openlab [1] for providing GPU resources that were used to conduct the studies presented within this work.

TABLE OF CONTENTS

1 Introduction

Benchmarking is a fundamental process in high-performance computing enabling the systematic comparison of hardware performance across diverse systems and architectures. In the context of the Worldwide LHC Computing Grid (WLCG), which unites approximately 170 computing centers across more than 40 countries to process and analyze the vast data volumes produced by CERN's Large Hadron Collider (LHC) experiments [2], benchmarking plays a critical role for its fair-sharing model and is required for resource planning and procurement decisions.

For CPU-based workloads, this role is fulfilled by the HEPiX Benchmarking framework [3], which comprises three major components:

- **HEP-Workloads** [4]: A collection of real-world HEP applications that are containerized and encapsulate the software and input data for representative physics tasks provided by various experiments. Each workload is provided as a standalone container image and provides an individual result which is normalized against a reference machine.
- **HEP-Score** [5]: The framework responsible for orchestrating the execution of multiple workloads and computing a composite benchmark score. It sequentially executes a selected set of workloads and calculates the overall score as the geometric mean of the individual, normalized results. Today's current standard set is referred to as HEPScore23 [6], which is the standard CPU benchmark for the WLCG. It consists of seven experiments from five experiments: ATLAS, CMS, ALICE, LHCb, and Belle II.
- **HEP Benchmark Suite** [7]: An overarching orchestration framework that manages the execution environment, collects various system metrics through a plugin system, and compiles the results into a standardized JSON output for further analysis. It can be used with HEPScore23 and various legacy benchmarks, such as HS06, SPEC CPU 2017, and DB12.

With the approaching High-Luminosity LHC (HL-LHC) era, computational demands are expected to grow substantially, and Graphical Processing Units (GPUs) have emerged as a key technology to meet these requirements. Their massive parallelism and high throughput make them particularly well suited for certain computationally intensive tasks, such as tracking and event reconstruction, simulation, and machine-learning inference which are a major components of many modern HEP workflows. Accordingly, the HEPiX Benchmarking working group is extending HEPScore to GPU resources in anticipation of their wide adoption within the WLCG in the future: HEPScore4GPU aims to provide an analogous, representative benchmark for GPU-based workloads.

While the philosophy of HEPScore, both for CPUs and GPUs, is that benchmarks should be composed of real-world HEP applications to ensure maximum representativeness, GPU-accelerated applications within the experiments are still in development and no production-ready applications are available yet. This creates a practical gap: The WLCG already needs a way to assess and compare GPU server performance now, while HEP GPU workflows are still being developed and improved.

2 Selection and Overview of non-LHC GPU Benchmarks

To identify suitable general-purpose GPU benchmarks for the WLCG, we surveyed existing benchmark suites already adopted by major HPC procurement initiatives. A particularly relevant reference is the benchmarking framework developed by the European High Performance

Computing Joint Undertaking (EuroHPC JU) [8] for the procurement of the Leonardo pre-exascale supercomputer (SMART 2019/1084, Lot 3), hosted at CINECA in Italy [9].

The EuroHPC benchmark procedure was designed to evaluate candidate systems using metrics directly relevant to scientific computing: *time-to-solution*, *energy-to-solution*, and the minimum system scale required to execute each workload [10]. The benchmark suite was divided into two categories:

- **Synthetic kernels**, which evaluate system-wide performance independently of any specific scientific workload.
- **Scalable applications**, which represent real-world scientific problems and are expected to scale to a large fraction of the system.

The full suite comprised six benchmarks, summarized in Table 1.

Table 1: Benchmark suite used in the EuroHPC Lot 3 (Leonardo) procurement.

#	Benchmark	Type	Domain
1	HPL	Synthetic kernel	Dense linear algebra
2	HPCG	Synthetic kernel	Sparse linear algebra
3	Quantum ESPRESSO	Scalable application	Electronic structure
4	PLUTO	Scalable application	Astrophysical fluid dynamics
5	SPECFEM3D Globe	Scalable application	Seismology
6	MILC	Scalable application	Lattice QCD

The two synthetic kernels (HPL and HPCG) measure peak and sustained floating-point throughput but do not reflect the algorithmic aspects of real workloads, and were therefore not considered for HEPScore4GPU. Of the four scalable applications, PLUTO targets compressible magnetohydrodynamics but lacks GPU-acceleration, leaving three candidates with established GPU support: SPECFEM3D Globe, Quantum ESPRESSO, and MILC. These three were selected for further investigation in this project.

An additional selection criterion was licensing. All three implementations are released under permissive open-source licenses, which is a prerequisite for redistribution across the WLCG's globally distributed infrastructure and for integration into HEPScore.

The following subsections describe each selected application, its scientific domain and implementation languages.

2.1 SPECFEM3D Globe

SPECFEM3D Globe is a seismological and geodynamic simulation code that models global and regional (continental-scale) seismic wave propagation [11]. It is widely used in Earth sciences to simulate how seismic waves travel through planetary interiors, with applications extending beyond Earth to Mars and the Moon.

The code is primarily written in Fortran, with some parts implemented in C. It supports distributed-memory parallelism via MPI (tested in this project with OpenMPI) and GPU acceleration through CUDA [12], with a HIP port also available for AMD architectures [13].

The computational workload is governed by the spatial resolution of the spectral-element mesh and the duration of the simulated seismic event, both of which can be adjusted to scale the problem size over several orders of magnitude. For the purposes of this benchmark study,

a regional example (`regional_Greece_small`) from the SPECFEM3D Globe distribution is used, with parameters tuned to achieve a suitable balance between memory footprint and computation time.

2.2 Quantum ESPRESSO

Quantum ESPRESSO (*opEn-Source Package for Research in Electronic Structure, Simulation, and Optimization*) is an integrated suite of open-source codes for electronic-structure calculations and materials modeling at the nanoscale [14]. It is based on density-functional theory (DFT), plane-wave basis sets, and pseudo-potentials, and is one of the most widely used *ab initio* simulation packages in condensed-matter physics and chemistry.

The codebase is written predominantly in Fortran. Parallelism is achieved through MPI for distributed-memory execution and CUDA for GPU acceleration [15]. The input datasets used in this project are taken from the CINECA EuroHPC benchmark repository [16].

The workload can be steered through several physical parameters that control the plane-wave energy cut-offs and the density of the Brillouin-zone sampling grid, allowing the user to trade memory footprint against computation time.

2.3 MILC

MILC (MIMD Lattice Computation) is a collaborative software project for Lattice Quantum Chromodynamics (QCD) simulations [17]. It is used to study the strong interaction between quarks and gluons by discretizing space-time onto a four-dimensional Euclidean lattice and evaluating the QCD path integral numerically using hybrid Monte Carlo methods.

The code is written primarily in C. Distributed-memory parallelism is provided through MPI (tested with OpenMPI), while GPU acceleration is achieved with the aid of the QUDA library, which supplies optimized kernels for the lattice Dirac operator on NVIDIA GPUs [18]. Input datasets are borrowed from the CINECA EuroHPC benchmark repository [16].

The problem size is controlled by the spatial and temporal extents of the lattice as well as the number of Monte Carlo trajectories to be computed, providing a mechanism for scaling the workload.

3 Technical Setup and Implementation

3.1 Containerization Strategy: Docker and Apptainer

Deploying complex scientific applications across the heterogeneous computing centers of the WLCG is notoriously difficult due to varying host operating systems, compiler versions, and GPU driver stacks. To ensure reproducibility, isolate dependencies, and simplify distribution, this project adopts a fully containerized workflow.

Docker [19] is utilized during the development and build phase. Its layered filesystem and build cache allow for the efficient creation of multi-stage environments. We compile the applications and all dependencies in a full build stage, and copy only the runtime artifacts into a minimal runtime stage, keeping the distributed images as small as possible.

However, Docker requires root privileges, which poses security risks and is usually disallowed in shared (HPC) environments. Therefore, for actual execution on the WLCG test machines, the Docker images are exported and converted to the **Apptainer** [20] (formerly Singularity)

format. Apptainer is the current standard container runtime in the WLCG. It fully operates without root privileges and integrates seamlessly with cluster job schedulers.

3.2 Repository Architecture and Automation

To manage the diverse build requirements of SPECFEM3D Globe, Quantum ESPRESSO, and MILC, a unified and automated repository structure was designed.

Repository Structure

```
benchmarks/
  create_benchmark.sh     # Builds, tests, and exports container images
  milc/
     Dockerfile           # Container build definition for MILC
     run_benchmark.sh     # Workload execution and metric extraction
  quantum_espresso/
     Dockerfile           # Container build definition for Quantum ESPRESSO
     inputs/              # Small and large input decks
     pseudo/              # Pseudopotential files
     run_benchmark.sh     # Workload execution and metric extraction
  specfem3d_globe/
     Dockerfile.cpu       # CPU-only build definition
     Dockerfile.cuda      # NVIDIA CUDA build definition
     Dockerfile.hip       # AMD ROCm/HIP build definition
     run_benchmark.sh     # Workload execution and metric extraction
```

The core of this automation is the `create_benchmark.sh` script. It abstracts the complexity of the underlying build systems, allowing the user to specify the target hardware backend (e.g., `cuda`, `hip`, or `cpu`). The script dynamically selects the appropriate `Dockerfile`, executes the build, and finally exports the image as a tarball. This tarball can then be transferred to a server and converted into an Apptainer image (`.sif`).

3.3 Integration with the HEP Benchmark Suite

A primary objective of this project was to evaluate how these non-LHC applications could be executed within the existing WLCG benchmarking ecosystem. To achieve this, the benchmarks were orchestrated using a specialized development branch of the HEP Benchmark Suite [7]. Unlike the standard release, which is strictly configured to run the predefined HEP benchmarks, this specific branch allows the Suite to accept and execute arbitrary container images. By leveraging this modified Suite, we were able to fully automate the execution and data acquisition of our custom GPU benchmark containers, ensuring a standardized and reproducible evaluation process without having to build a custom orchestration framework from scratch.

3.4 Execution Workflow

Inside the container, the `run_benchmark.sh` script acts as the bridge between HEPScore (executed through the Suite) and the scientific application. It is responsible for dynamically configuring the workload size, setting up the execution environment, and optionally invoking hardware profilers.

To ensure the benchmarks are representative and scalable, the script allows the workload size and execution time to be controlled via application-specific parameters. For SPECFEM3D Globe, the spatial resolution of the seismic mesh is controlled by the number of spectral elements along the ξ and η axes (`NEX_XI` and `NEX_ETA`), while the simulation duration is governed by `RECORD_LENGTH_IN_MINUTES`. Because SPECFEM3D Globe requires the number of elements per MPI rank to be a strict multiple of 8 for optimal GPU memory coalescing and vectorization, the script dynamically calculates the optimal 2D MPI domain decomposition based on the chosen mesh size and the available hardware resources.

For Quantum ESPRESSO, the computational cost and memory footprint are scaled by adjusting the plane-wave energy cutoffs for the wave function (`ecutwfc`) and the charge density (`ecutrho`), alongside the density of the Brillouin-zone sampling grid (defined by the number of `K-points` along the three lattice vectors).

For MILC, the workload is defined by the dimensions of the four-dimensional space-time lattice (`nx`, `ny`, and `nz` for the spatial axes, and `nt` for the temporal axis) as well as the total number of Monte Carlo trajectories (`trajecs`) to be computed.

By parameterizing these inputs, a single container image can be reused to generate different workload profiles, e.g., short validation runs versus long or memory-heavy benchmark runs, without the need to rebuild the container.

Furthermore, the script supports hardware profiling to identify GPU utilization bottlenecks. By setting an environment variable, the execution can be wrapped in NVIDIA Nsight Systems [21] for timeline analysis or Nsight Compute [22] for kernel-level profiling. These tools produce artifacts that reveal whether the applications are primarily compute-bound, memory, or memory-bandwidth-bound, or limited by host-to-device data transfers.

Finally, upon completion of the run, the script parses the application's native output files to extract the wall-clock time. This timing data is then formatted and emitted as a standardized result token, which is subsequently ingested by the HEP Benchmark Suite to calculate the final performance metric.

4 Initial Results and Profiling

4.1 Measurement and Analysis Methodology

All example runs presented in this section were executed on a server equipped with an NVIDIA A100 GPU (40 GB VRAM). Although the A100 offers ample device memory, the workload configurations (Table 2) were deliberately chosen such that the GPU memory footprint of every application remains below 16 GB, so that the same container images and input configurations can later be executed on smaller GPUs available elsewhere in the WLCG.

During each run, the GPU utilization, GPU power draw, and the allocated GPU memory were sampled at one-second intervals by the monitoring component of the HEP Benchmark Suite, which queries and digests `nvidia-smi` on the host. The post-processing and statistical evaluation of the resulting traces was carried out in Jupyter notebooks with Python.

The raw traces contain non-representative phases that are not representative of the actual GPU utilization steady-state behavior of the applications, such as initialization, kernel warm-up and teardown. These are removed by selecting an analysis window that extends from the first to the last sample whose value exceeds a threshold chosen clearly above the idle baseline. All samples outside this window are excluded from the statistics. The retained window is highlighted in yellow in Figures 1–3.

For SPECFEM3D Globe, the trace contains both the mesh generation (`xmeshfem3D`) and the wave-propagation solver (`xspecfem3D`) phases. The mesher executes almost entirely on the

host and leaves the GPU idle, it is therefore excluded from the analysis, and only the solver phase is taken into consideration.

SPECFEM3D Globe		**Quantum ESPRESSO**		**MILC**	
NEX_XI	144	ecutwfc [Ry]	20	n_x	32
NEX_ETA	144	ecutrho [Ry]	160	n_y	32
Record length [min]	30	K-points	4 4 3	n_z	32
				n_t	32
				Trajectories	4

Table 2: Workload configurations ("big") used for the example runs. All configurations were sized to remain below 16 GB of GPU memory.

4.2 Evaluation Criteria for a GPU Benchmark

Before discussing the individual applications, it is useful to define the behavior expected from a workload that should serve as a GPU benchmark for the WLCG. The measured time-to-solution should reflect the compute capability of the device itself, rather than being limited by the host, the interconnect or the I/O subsystem. Concretely, the following properties are sought:

- **Stable GPU utilization:** the utilization should remain consistently close to 100 % throughout the run indicating that the device is continuously saturated.
- **Maximum power draw:** the power consumption should approach the sustained power envelope (TDP) of the GPU. Power draw directly reflects the silicon activity. A workload that drives the device persistently to its power limit fully exploits the hardware's potential, making the resulting score sensitive to differences between GPU models.
- **No long idle periods:** the trace must not contain extended intervals in which the GPU returns to its idle state, as these indicate host-side bottlenecks, synchronization or I/O phases that would dilute the benchmark's sensitivity to GPU performance.
- **Compute-bound kernels:** the dominant kernels should be compute-bound rather than memory- or latency-bound, so that the benchmark scales with the floating-point capability of the device. This is verified through kernel-level profiling (Section 4.7).

4.3 SPECFEM3D Globe

The SPECFEM3D Globe trace consists of two clearly separated phases (Figure 1). During the mesher phase, the GPU remains at its idle baseline of approximately 44 W. Once the solver starts, the utilization remains at 100 % and the power draw stabilizes at a nearly constant level, with a mean of 248.2 W, a median of 248.8 W and a 85th percentile (q85) of 254 W. The q85 is chosen as a good indicator of a stable workload's general power consumption, as it ignores uncommon outliers. It lies slightly above the card's nominal TDP of 250 W. This is not a contradiction: the TDP represents the maximum sustained power for a typical workload rather than an absolute physical limit, so spikes above the TDP can occur, while the mean power draw remains within it. The small spread between mean, median and q85 indicates a steady, saturated state for the duration of the solver, with no idle periods within the analyzed window.

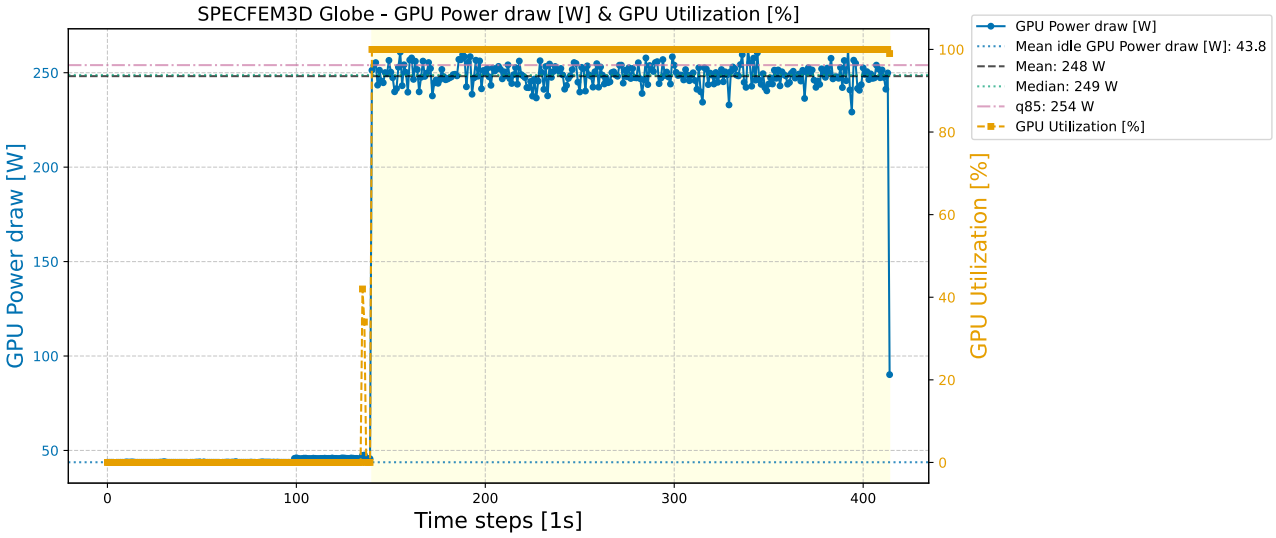


Figure 1: GPU power draw and GPU utilization during an example SPECFEM3D Globe run, sampled at 1 s intervals. The first phase corresponds to the mesher, which runs on the host and leaves the GPU idle, only the solver phase is considered for the hardware analysis. The window relevant for the statistics is highlighted in yellow.

4.4 Quantum ESPRESSO

Quantum ESPRESSO shows a markedly more irregular pattern (Figure 2). The utilization repeatedly reaches 100 %, but is interrupted by frequent short dips, and the power draw oscillates between roughly 70 W and peaks above 350 W, yielding a mean of 210 W and a median of 240 W. This illustrates why, for irregular workloads, the mean is the more conclusive statistic. The median, read on its own, would suggest a device operating close to its power envelope, whereas the mean properly accounts for the frequent dips and reveals the more accurate picture of the power draw. Within the analyzed window, the GPU does not return to the idle baseline for extended periods, but the utilization is not stable in the sense defined above.

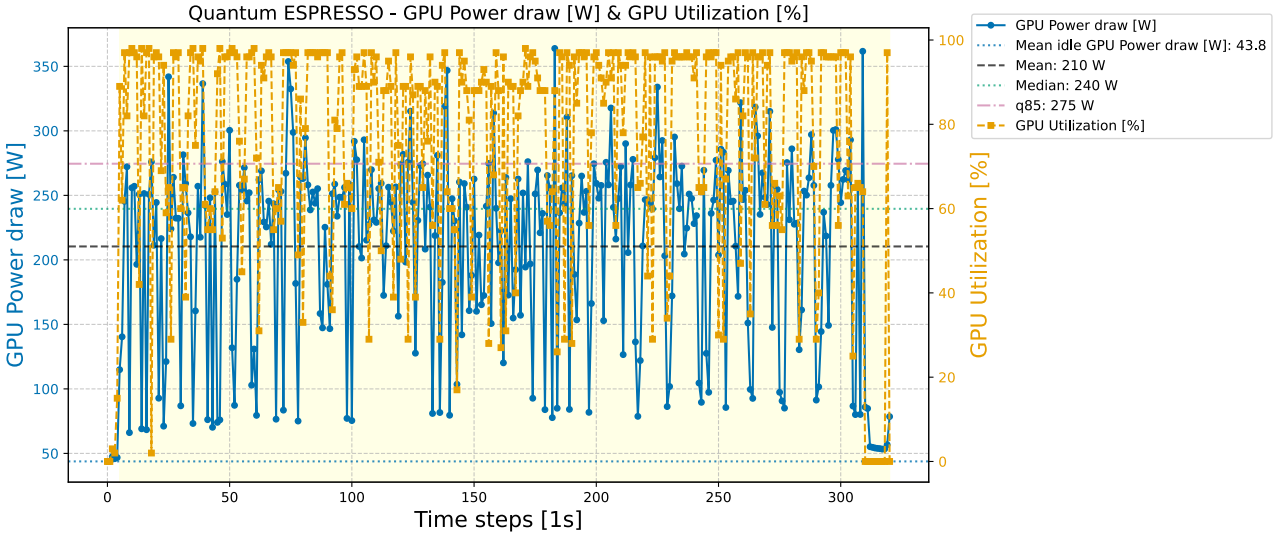


Figure 2: GPU power draw and GPU utilization during an example Quantum ESPRESSO run, sampled at 1 s intervals. The window retained for the statistics is highlighted in yellow.

4.5 MILC

The MILC trace has a different character (Figure 3). The execution is organized in four discrete bursts of high utilization, corresponding directly to the four configured trajectories (`trajecs = 4`), separated by intervals during which the power draw returns to the ∼44 W idle baseline. Similarly to Quantum ESPRESSO discussed above, the GPU utilization is also unstable during the trajectory computing phase. Over the analyzed window, the mean power draw is 145.2 W and the median is 93.4 W, both kept low by the idle intervals, while the q85 of 255.8 W is dominated by the active phases. The GPU is therefore idle for a substantial fraction of the run, and the measured time-to-solution includes host-side phases that are not representative for GPU performance.

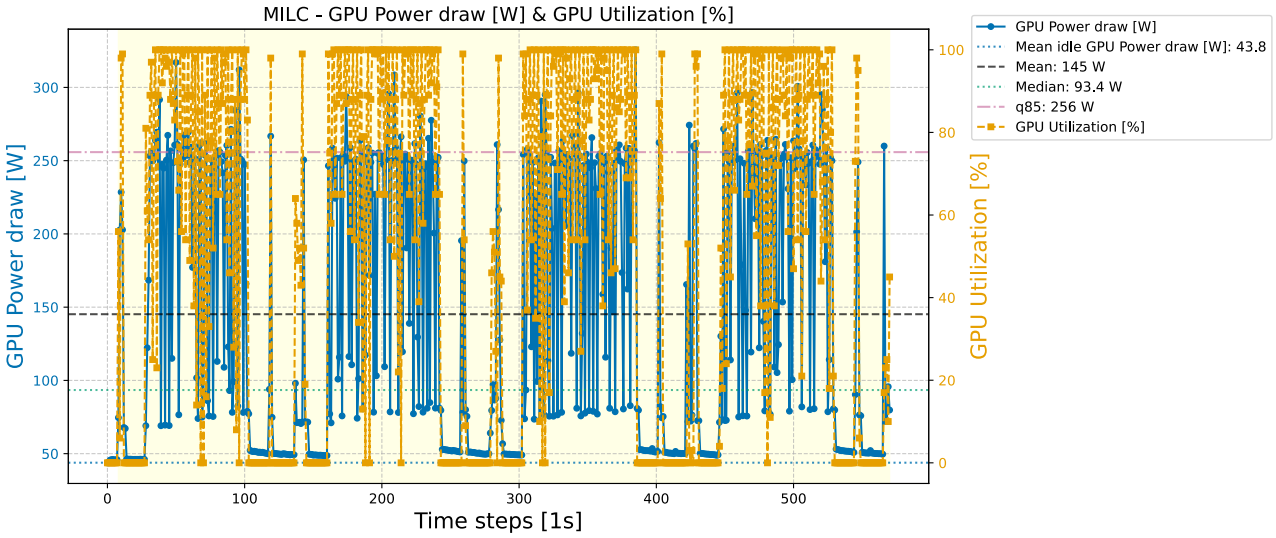


Figure 3: GPU power draw and GPU utilization during an example MILC run, sampled at 1 s intervals. The four trajectory computations are visible as four active phases of full utilization. The window retained for the statistics is highlighted in yellow.

4.6 GPU Memory Footprint

The allocated GPU memory remained essentially constant during the active phases of all three applications, as shown in Figure 4. No substantial growth or instability was observed over the run time. All configurations remain within the 16 GB target defined for portability across WLCG machines.

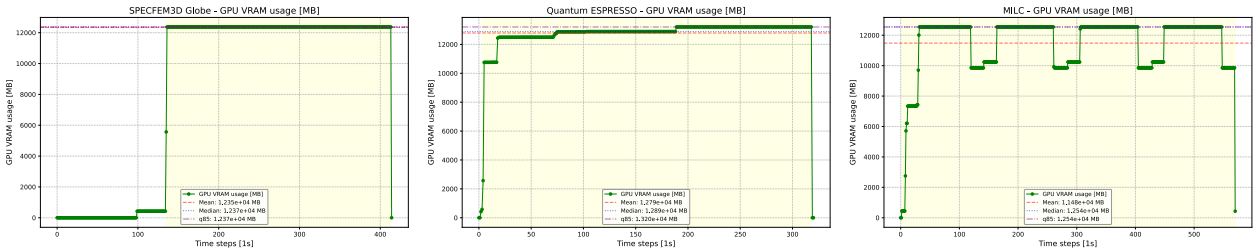


Figure 4: Allocated GPU memory during the example runs of SPECFEM3D Globe (left, 12 371 MB peak), Quantum ESPRESSO (center, 13 205 MB peak) and MILC (right, 12 543 MB peak).

4.7 Kernel-Level Profiling

To complement the one-second utilization samples obtained from the Suite's measurements, each application was additionally profiled with NVIDIA Nsight Systems, and the achieved SM warp occupancy was inspected. The results are shown in Figure 5.

SPECFEM3D Globe maintains a consistently high SM warp occupancy over the analyzed window, which is consistent with compute-bound solver kernels. Quantum ESPRESSO exhibits two regimes: an initial phase with intermittent kernel activity separated by idle gaps followed by a phase of sustained but strongly oscillating occupancy, mirroring the power-draw trace. For MILC, the occupancy is moderate but oscillating within each trajectory burst, while no kernel activity is recorded between bursts, indicating a poor overall GPU utilization.

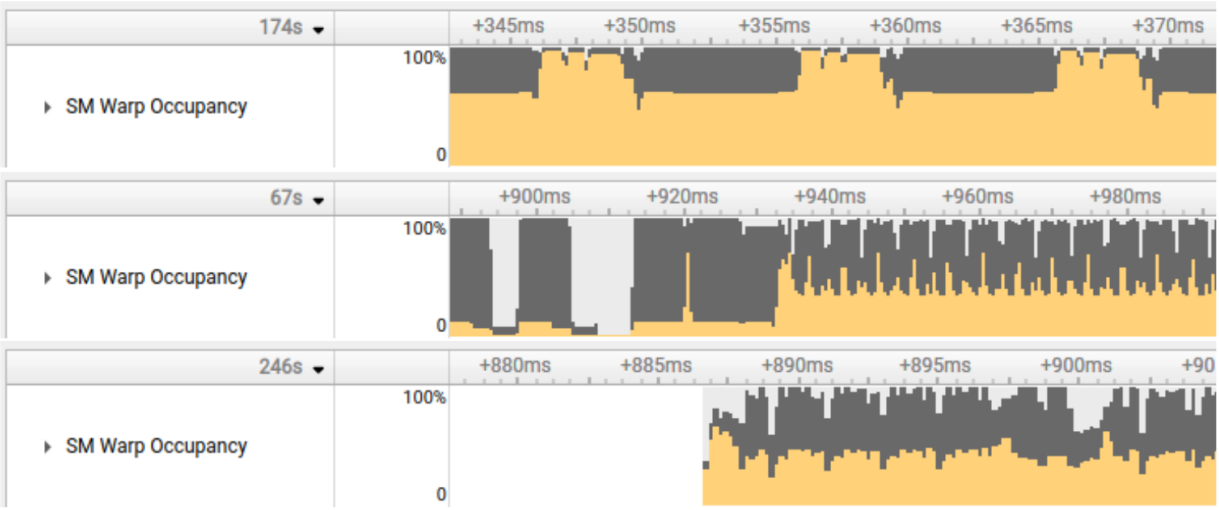


Figure 5: SM warp occupancy timelines obtained with Nsight Systems for SPECFEM3D Globe (top), Quantum ESPRESSO (middle) and MILC (bottom).

The attribution of the idle intervals to host-side computation, communication, or data movement is based on the correlation of the monitoring traces with the occupancy timelines. A definitive disambiguation would require inspecting the CPU, and memory-copy rows of the Nsight Systems timeline for the corresponding intervals. Since the main goal is the selection of appropriate workloads rather than the optimization of the investigated, this is neglected for this work, as the results are conclusive enough already.

The implications of these observations for the final benchmark selection are discussed in Section 6.

5 Validation

The initial example runs of Section 4 explored the qualitative GPU behavior of the three candidates. A benchmark intended for procurement and resource accounting in the WLCG must also be **reproducible**. Repeated executions of the same workload on the same hardware must yield (nearly) identical results. To validate this property, each application was executed 40 times on the NVIDIA A100 test system using the same "big" workload configuration, and the distributions of the resulting metrics were analyzed.

For each run, two quantities are of primary interest: the mean GPU power draw over the trimmed analysis window, and the execution time. Following the HEPScore23 philosophy that a benchmark score should be proportional to the throughput of the system, the execution time

is expressed as its inverse. With this convention, a faster machine yields a higher value, so the metric can be used directly as a "higher-is-better" score when comparing different machines, exactly as HEPscore23 is used for CPUs.

5.1 The Coefficient of Variation as Stability Criterion

To judge the reproducibility of the candidates, the coefficient of variation (CV) is used as the central stability criterion. It is defined as the ratio of the standard deviation to the mean of a sample,

$$\mathrm{CV} = \frac{\sigma}{\mu}, \tag{1}$$

and is therefore a dimensionless measure of the relative spread of a distribution. This normalization has two important advantages for our use-case. First, it makes different physical quantities (power draw in watts, throughput in inverse seconds, memory in megabytes) comparable on a common scale, despite their entirely different units and magnitudes. Second, it makes the stability of a workload comparable across different machines: a metric that varies by $\pm 1\,\mathrm{W}$ is excellent for a 250 W workload but unacceptable for a 20 W one, and only the relative spread captures this.

For a benchmark, the CV should be as small as possible. A small relative spread means that the reported score is representative of the capabilities of the hardware under test. As a rule of thumb, a CV of 0.1 % or below indicates excellent reproducibility. Values of a few percent correspond to visible variability and would render the metric too unstable to serve as a reliable benchmark score.

5.2 Statistical Stability of the Candidates

Table 3 summarizes the aggregated statistics of the 40 validation runs per application.

	SPECFEM3D Globe	Quantum ESPRESSO	MILC
Mean power draw [W]	249.0	202.3	146.5
Median power draw [W]	249.1	202.3	146.5
Power CV [%]	0.40	7.07	4.18
Mean performance reference $(1/T)$ $[\frac{1}{s}]$	0.0032	0.0030	0.0011
Performance reference $(1/T)$ CV [%]	0.019	0.21	0.042

Table 3: Aggregated statistics over the 40 validation runs per application. Power statistics were computed over the trimmed analysis windows, the runtime is the wall-clock time of the benchmarked phase only.

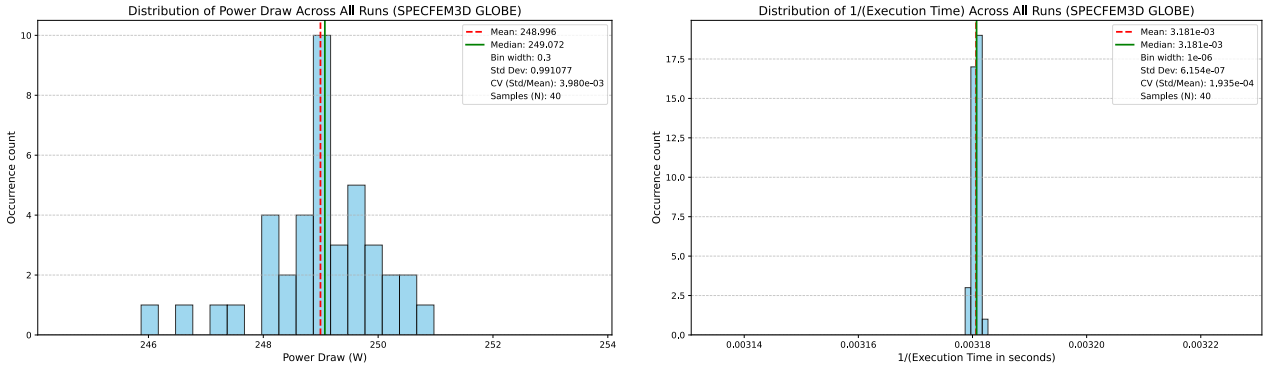


Figure 6: SPECFEM3D Globe: distributions of the per-run mean power draw (left) and of the performance reference $1/T$ (right) over the 40 validation runs. Both distributions are tightly concentrated around their means.

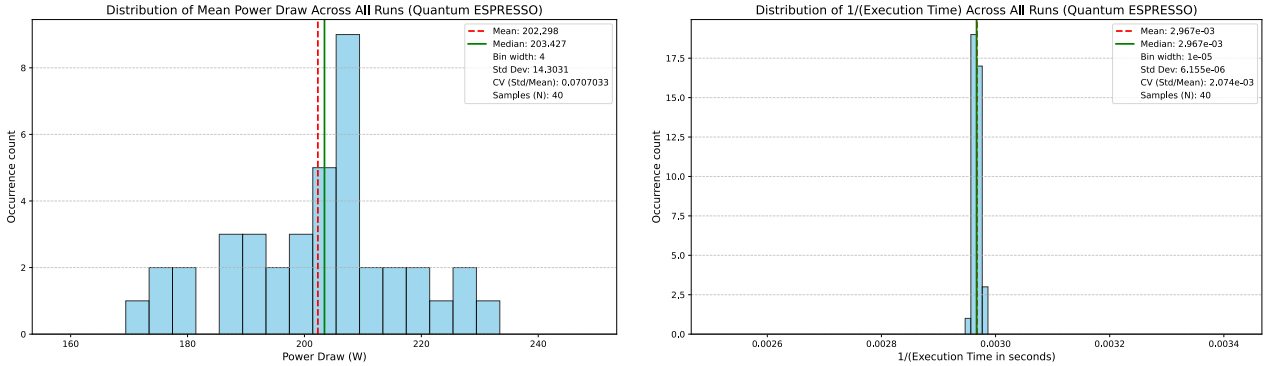


Figure 7: Quantum ESPRESSO: Distributions of the per-run mean power draw (left) and of the performance reference $1/T$ (right) over the 40 validation runs.

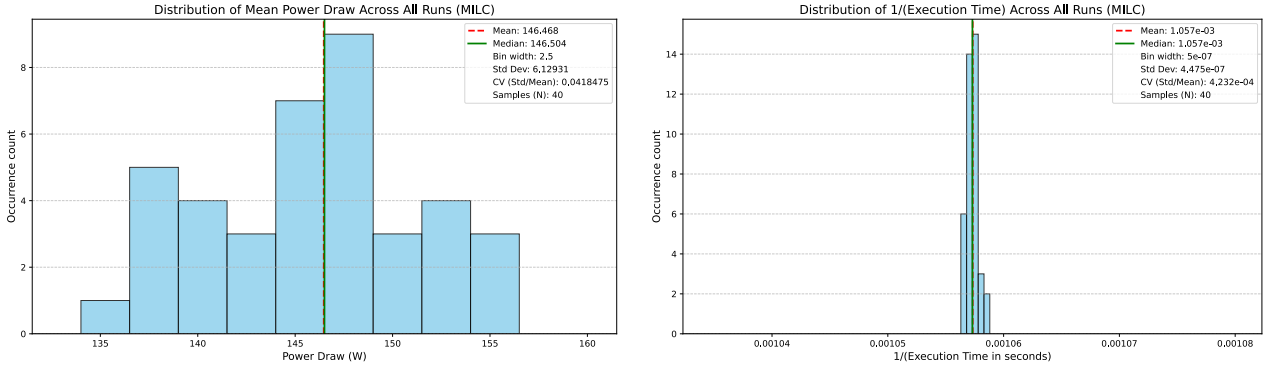


Figure 8: MILC: Distributions of the per-run mean power draw (left) and of the performance reference $1/T$ (right) over the 40 validation runs.

SPECFEM3D Globe shows the most reproducible behavior of the three candidates (Figure 6). Its throughput distribution is concentrated in a window of less than $1 \times 10^{-4}\,\mathrm{s}^{-1}$ around the mean of $3.181 \times 10^{-3}\,\mathrm{s}^{-1}$, corresponding to a CV of only 0.019 %. The power draw is also stable with a CV of 0.40 % around a mean of 249.0 W. Therefore, repeated executions

of SPECFEM3D Globe produce essentially the same score and always reach the performance limits of the GPU, which is precisely the property required for a benchmark.

The corresponding distributions for Quantum ESPRESSO and MILC, as shown in Figure 7 and Figure 8 respectively, are visibly broader. Their performance reference remains well reproducible, with CVs of 0.21 % and 0.042 %, which is still small in absolute terms. The power draw, however, varies substantially more: Quantum ESPRESSO exhibits a CV of 7.07 % around a mean of 202.3 W, and MILC a CV of 4.18 % around 146.5 W. This is consistent with the observations of Section 4: the alternating CPU/GPU phases of Quantum ESPRESSO and MILC make the instantaneous power draw sensitive to the exact phase mix falling into the analysis window, whereas the fully saturated SPECFEM3D Globe solver draws a nearly constant power in every run.

5.3 Cross-Hardware Evaluation of SPECFEM3D Globe

Given its stability on the A100, SPECFEM3D Globe was additionally executed on three further NVIDIA GPUs (V100, T4, L4) using the same container image and workload configuration, and on one AMD device (see below). Figure 9 summarizes the mean runtime, the mean power draw, and the achieved fraction of each device's maximum power draw.

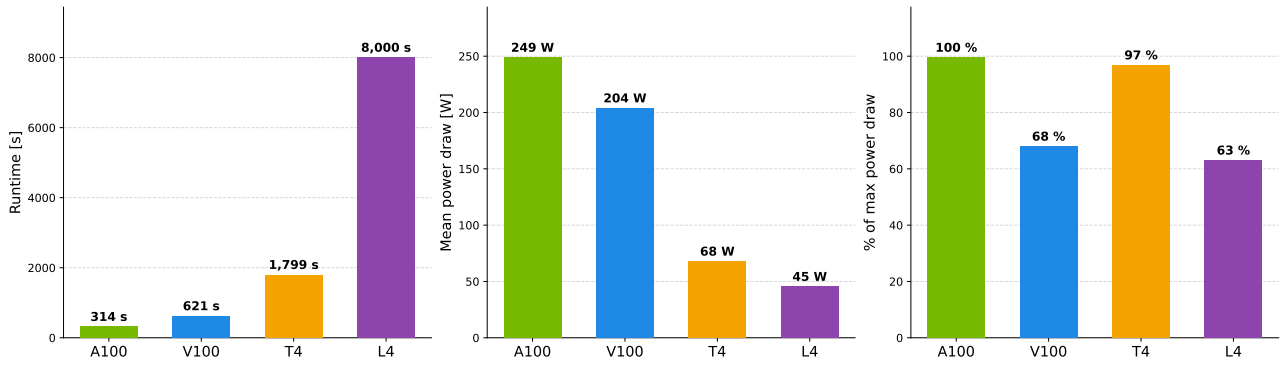


Figure 9: SPECFEM3D Globe on four NVIDIA GPUs: mean runtime (left), mean power draw (centre), and achieved fraction of the maximum power draw (right) of the validation runs.

The runtimes follow the expected ordering of the devices' suitability for the selected workload: 314 s on the A100, 621 s on the V100, 1 799 s on the T4, and 8 000 s on the L4. The factor of approximately two between the A100 and the V100 is consistent with their relative double-precision throughput. The T4 and the L4, in contrast, are inference-oriented devices with a very small number of FP64 units. The disproportionate slowdown by factors of roughly six and twenty-five relative to the A100 reflects this architecture mismatch rather than a general performance deficit of the devices. For the L4, an additional contributing factor is the set of CUDA compilation targets of the binary used in this work. A `cuobjdump` inspection shows that it contains only Ampere (sm_80) code. In particular, Ada Lovelace (sm_89), the architecture of the L4, is not among the compilation targets supported by SPECFEM3D Globe. The L4 therefore executes the Ampere code path via NVIDIA's binary compatibility (PTX) within major architecture version 8, rather than kernels tuned for its own architecture.

Overall, this is precisely the behavior desired from a WLCG benchmark. The workload is sensitive to the double-precision compute capability of the hardware and discriminates clearly between the devices.

The power measurements support this picture. The mean power draw scales with the TDP class of each device (249 W, 204 W, 68 W and 45 W for the A100, V100, T4 and L4, respectively). Expressed as a fraction of the maximum power draw, the workload drives the A100 and the T4 close to their power envelopes (100 % and 97 %), whereas on the V100 and the L4 only 68 % and 63 % are reached, indicating that on these devices the execution is limited by other resources, mainly FP64 throughput, but maybe also memory bandwidth or thermal throttling.

AMD Portability Run Another relevant aspect for a benchmarking workload feasible for the WLCG is portability. A full vendor lock-in needs to be avoided to foster an open environment. Beyond the NVIDIA ecosystem, SPECFEM3D Globe was therefore also executed on an AMD Radeon AI Pro R9700 (32 GB VRAM, 300 W TDP) using the HIP build of the container. The run completed in 1 239.16 s, approximately four times the A100 runtime, which is consistent with the lower double-precision capability of the device. The corresponding power and utilization trace, as shown in Figure 10, indicates the same qualitative behavior as on the NVIDIA GPUs. After the host-side mesher phase, the utilization remains at 100 % throughout the solver and the power draw stabilizes at a mean of 257 W (median 258 W, q85 260 W) above an idle baseline of 24.1 W. That's roughly 86 % of the card's 300 W TDP. The AMD device therefore exhibits the same saturated, stable execution profile as the CUDA builds, confirming that the HIP port utilizes the hardware in the same manner. Together, the runtime ordering and the stable power profile demonstrate that the containerized benchmark executes correctly and meaningfully on AMD hardware, check-marking another demand on the workload.

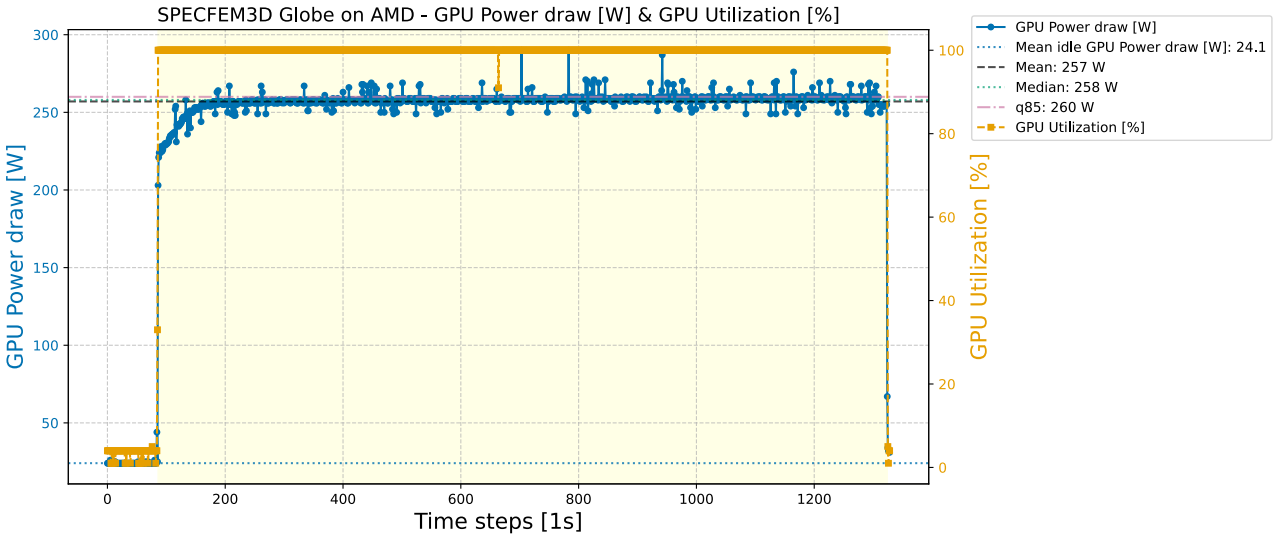


Figure 10: SPECFEM3D Globe on the AMD Radeon AI Pro R9700 (HIP build): GPU power draw and GPU utilization sampled at 1 s intervals. The window retained for the statistics is highlighted in yellow.

Taken together, the validation campaign shows that all three applications produce reproducible time-to-solution values. SPECFEM3D Globe additionally delivers a highly stable and saturated power profile, which is what we are looking for, and its score scales sensibly across different GPU architectures.

6 Conclusions

The objective of this project was to determine whether general-purpose, non-LHC GPU benchmarks can serve as an intermediate solution for HEPScore4GPU during the period in which the GPU workflows of the LHC experiments are still evolving. Starting from the benchmark suite used by the EuroHPC JU in the procurement of the Leonardo supercomputer, three applications with mature GPU support (SPECFEM3D Globe, Quantum ESPRESSO, MILC) were selected, containerized, and evaluated with a development branch of the HEP Benchmark Suite. The main investigation was conducted on an NVIDIA A100 server provided by CERN openlab.

The evaluation through qualitative profiling, a 40-run statistical validation campaign, and cross-hardware as well as cross-vendor executions, shows that all three candidates satisfy the basic reproducibility requirement of a benchmark: the coefficient of variation of the performance reference stays at or below 0.21 % for every application. The candidates differ strongly, however, in how they load the GPU.

Quantum ESPRESSO uses the device intensively but with a variable pattern of alternating GPU kernels and host-side or communication phases, reflected in a power-draw CV of 7.07 %. MILC, in the tested configuration, organizes its work into discrete trajectory bursts separated by extended idle intervals, so that the GPU remains idle for a substantial fraction of the run and the median power draw stays far below the device's power envelope. SPECFEM3D Globe, in contrast, keeps the GPU fully saturated throughout the run. The utilization remains at 100 %, the power draw stabilizes at approximately 249 W with a CV of only 0.40 %, and the throughput is reproducible within 0.019 %.

Based on these findings, only SPECFEM3D Globe was selected for further investigations, as it fully meets the demands on a potential benchmark workload. The cross-hardware runs confirm that its score is sensitive to the capability of the device under test. The mean runtime scales from 314 s on the A100 to 8 000 s on the L4, correctly tracking the double-precision compute capability of the four NVIDIA GPUs and even exposing the absence of an Ada Lovelace compilation target on the L4. The HIP build reproduces the same saturated execution profile on an AMD Radeon AI Pro R9700, driving the card to roughly 86 % of its TDP. This, however, must not be interpreted as a too small utilization, since the relevant pipelines are fully saturated (FP64). SPECFEM3D Globe therefore emerges as the best of the tested applications, combining full GPU utilization, excellent run-to-run stability, and portability across vendors and architectures. Together, these properties make it a strong intermediate benchmark workload candidate for HEPScore4GPU, able to provide meaningful GPU server comparisons for the WLCG while HEP-specific GPU workflows continue to stabilize.

It should nevertheless be kept in mind that SPECFEM3D Globe, like the two other candidates, is not an CERN-centric application. Its role is to characterize the hardware, not to represent the WLCG workload mix. Establishing the representativeness of the final benchmark remains the purpose of the ongoing HEPScore4GPU development, to which the present work contributes a validated methodology and another candidate workload.

7 Outlook

The findings of this project suggest three main directions until a HEPScore4GPU, mainly based on HEP applications, becomes available.

Investigation of further applications. The survey and evaluation performed in this work should be extended to additional general-purpose or non-LHC GPU applications. A broader

candidate pool increases the chance of identifying workloads with complementary characteristics (memory access patterns, communication profiles) and provides additional points of cross-validation for the hardware ranking obtained with SPECFEM3D Globe. Beyond classical scientific codes, workloads from other GPU-intensive domains could also be interesting benchmark candidates. VFX rendering and AI inference or training, for instance, account for a large and growing share of real-world GPU usage in the lower precision domains, which might also be adapted in High-Energy Physics in the future, where feasible.

Separate applications for different floating-point precisions. The cross-hardware results, in particular the disproportionate slowdown observed on the inference-oriented T4 and L4, demonstrate that a single benchmark cannot characterize all devices, as their capabilities differ strongly between floating-point precisions. Since HEP GPU workflows themselves might employ a mixture of double, single and reduced precision, a GPU benchmark for the WLCG would benefit from separate benchmark applications and a dedicated score for the different precisions, so that FP64-oriented HPC devices and FP32/FP16-oriented accelerators can each be assessed meaningfully.

Profiling and classification of existing HEP workflows. As the GPU applications of the experiments reach production maturity, they should be profiled with the methodology applied in this report: utilization, power draw, memory footprint, warp occupancy, and floating-point precision pipeline usage. Even though a high demand of FP64 precision capability is expected for most HEP applications, a detailed investigation can provide an overview of the current trends and how they develop along the way of becoming production-ready. Eventually, such a classification defines the target behavior that HEPScore4GPU must reproduce, allows the non-LHC benchmarks to be cross-validated against real CERN HEP workloads, and ultimately guides the composition of the final, HEP-representative benchmark.

8 References

[1] CERN. *CERN openlab*. URL: https://openlab.cern/ (visited on 09/04/2026).

[2] CERN. *The Worldwide LHC Computing Grid (WLCG)*. URL: https://home.cern/science/computing/grid/ (visited on 08/26/2026).

[3] D. Giordano et al. "HEPiX Benchmarking Solution for WLCG Computing Resources". In: *Comput Softw Big Sci 5* 28 (2021). DOI: 10.1007/s41781-021-00074-y.

[4] CERN. *HEP Workloads*. URL: https://gitlab.cern.ch/hep-benchmarks/hep-workloads (visited on 09/04/2026).

[5] CERN. *HEPScore*. URL: https://gitlab.cern.ch/hep-benchmarks/hep-score (visited on 09/04/2026).

[6] D. Giordano et al. "HEPScore: A new CPU benchmark for the WLCG". In: *EPJ Web of Conferences* 295 (2024). DOI: 10.1051/epjconf/202429507024.

[7] CERN. *HEP Benchmark Suite*. URL: https://gitlab.cern.ch/hep-benchmarks/hep-benchmark-suite (visited on 08/26/2026).

[8] EuroHPC Joint Undertaking. *European High Performance Computing Joint Undertaking*. URL: https://eurohpc-ju.europa.eu/ (visited on 08/27/2026).

[9] CINECA. *Leonardo Supercomputer*. URL: https://www.hpc.cineca.it/systems/hardware/leonardo/ (visited on 09/02/2026).

[10] EuroHPC Joint Undertaking. *LOT 3 Benchmark Information: Precursors to Exascale Supercomputers (SMART 2019/1084)*. URL: https://gitlab.hpc.cineca.it/eurohpc/benchmark/-/blob/master/Lot3-Benchmark_Information_v2.0.1.docx (visited on 08/26/2026).

[11] Laura Carrington et al. "High-frequency simulations of global seismic wave propagation using SPECFEM3D_GLOBE on 62K processors". In: *SC '08: Proceedings of the 2008 ACM/IEEE conference on High Performance Computing, Networking, Storage and Analysis*. IEEE, 2008. DOI: 10.1109/SC.2008.5215501.

[12] Dimitri Komatitsch, David Michéa, and Gordon Erlebacher. "Porting a high-order finite-element earthquake modeling application to NVIDIA graphics cards using CUDA". In: *Journal of Parallel and Distributed Computing* 69.5 (2009). DOI: 10.1016/j.jpdc.2009.01.006. URL: https://doi.org/10.1016/j.jpdc.2009.01.006.

[13] SPECFEM3D Globe team. *SPECFEM3D Globe Documentation*. URL: https://specfem3d-globe.readthedocs.io/en/latest/02_getting_started/ (visited on 08/27/2026).

[14] P. Giannozzi et al. "QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials". In: *Journal of Physics: Condensed Matter* 21.39 (2009). DOI: 10.1088/0953-8984/21/39/395502.

[15] P. Giannozzi et al. "QUANTUM ESPRESSO toward the exascale". In: *The Journal of Chemical Physics* 152.15 (2020). DOI: 10.1063/5.0005082.

[16] CINECA. *EuroHPC Benchmark Repository*. URL: https://gitlab.hpc.cineca.it/eurohpc/benchmark (visited on 08/27/2026).

[17] MILC Collaboration. *MILC Lattice QCD Code*. URL: https://github.com/milc-qcd/milc_qcd (visited on 08/27/2026).

[18] M. A. Clark et al. "Solving lattice QCD systems of equations using mixed precision solvers on GPUs". In: *Computer Physics Communications* 181.9 (2010), pp. 1517–1528. DOI: 10.1016/j.cpc.2010.05.002.

[19] Dirk Merkel. "Docker: Lightweight Linux Containers for Consistent Development and Deployment". In: *Linux Journal* (2014). URL: https://www.linuxjournal.com/content/docker-lightweight-linux-containers-consistent-development-and-deployment.

[20] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. "Singularity: Scientific containers for multipurpose users". In: *PLOS ONE* 12.5 (2017). DOI: 10.1371/journal.pone.0177459. URL: https://doi.org/10.1371/journal.pone.0177459.

[21] NVIDIA. *Nsight Systems User Guide*. URL: https://docs.nvidia.com/nsight-systems/ (visited on 08/27/2026).

[22] NVIDIA. *Nsight Compute Documentation*. URL: https://docs.nvidia.com/nsight-compute/NsightCompute/ (visited on 08/27/2026).