Evaluating the Influence of Measurement Frequency on Energy Readings Using Intel RAPL and NVIDIA NVML

MAXIMILIAN DAUNER, Munich University of Applied Sciences HM, Germany
MANUEL STEINBERG, Munich University of Applied Sciences HM, Germany
ANDREAS BRUNNERT, Munich University of Applied Sciences HM, Germany
BENEDIKT SCHICKER, Munich University of Applied Sciences HM, Germany
BENEDIKT ZÖNNCHEN, Munich University of Applied Sciences HM, Germany

Software energy attribution tools, from application-level profilers to carbon accounting frameworks, increasingly rely on hardware telemetry exposed through Intel RAPL and NVIDIA NVML. This paper quantifies how sampling interval affects RAPL- and NVML-based energy measurements to inform robust counter usage in energy and carbon attribution tooling. Across 588 controlled experiment runs, covering compute-bound, memory-bound, and mixed workloads from the NAS Parallel Benchmarks suite on Intel and AMD CPUs as well as NVIDIA data-center-, workstation-, and consumer-class GPUs, we evaluate sampling intervals from 0.5 ms to 1 s. Our results show that sampling interval affects RAPL and NVML in fundamentally different ways. This result is also visible when comparing both interfaces against external measurements collected during the benchmark runs. NVML cumulative-energy counters show frequent stale reads for intervals below 100 ms and strong platform-dependent behavior. On the consumer-class GPUs, the energy and power counters diverge at short intervals, leading to a mean energy underestimation of 95.4 % at 0.5 ms compared to integrated power measurements. In contrast, the workstation- and data-center-class GPUs show close agreement between cumulative energy and integrated power despite high stale-read rates, indicating that NVML reliability depends on device and sampling interval. For RAPL, energy readings remain stable at sampling intervals of 10 ms or coarser, whereas sub-millisecond sampling increases overhead without improving accuracy. Energy attribution tools should therefore avoid unvalidated NVML cumulative-energy counters on NVIDIA GPUs, prefer power integration, and sample RAPL at moderate, rather than sub-millisecond, intervals.

CCS Concepts: • **Hardware → Power estimation and optimization**.

Additional Key Words and Phrases: Software Energy, Measurement Frequency, NVML, RAPL

1 Introduction

Software energy attribution increasingly depends on hardware telemetry exposed by processor and accelerator interfaces. In cloud and shared computing environments, where external power meters are rarely available to users, Intel Running Average Power Limit (RAPL) [10] and the NVIDIA Management Library (NVML)[1] are often the practical basis for estimating CPU, DRAM, and GPU energy consumption. These interfaces are widely used because they are scalable, low-overhead, and available directly from software. This matters for energy and carbon attribution tools such as CodeCarbon[2], Kepler [1], PowerJoular [13], METRION [19], CPPJoules [16], and Scaphandre[3]. These tools are part of a larger group of software-based power meters[4] [4, 9] that derive higher-level estimates from these low-level telemetry sources. If the underlying readings are sensitive to sampling interval, software-level energy and carbon estimates can inherit systematic error. Consequently, sampling interval is not only a measurement setting but a downstream decision point for profilers, carbon accounting systems, per-job attribution, and energy-aware resource management: these tools may rank workloads, assign emissions, or trigger optimizations based on artifacts of the telemetry interface rather than actual energy use. Short intervals can introduce overhead and disturb the measured system [11, 15, 17], while long intervals reduce temporal resolution and can smooth short-lived workload phases [11, 18].

Despite this, the effect of sampling interval on RAPL and NVML energy readings is not well understood across CPU and GPU platforms. Prior work has studied RAPL accuracy, access mechanisms, and measurement overhead [8, 10, 15, 18], while GPU studies have shown that built-in sensors may exhibit lag, quantization, partial sampling, or hardware-specific behavior [3, 5, 6, 20]. Energy attribution tools often assume that the RAPL and NVML counters can be sampled at arbitrary intervals and that the resulting deltas are stable. While most RAPL-based tools rely on cumulative energy readings, NVML exposes both instantaneous power readings and, on more recent NVIDIA architectures [14], cumulative energy counters, which may not agree under all measurement configurations.

This paper quantifies how sampling interval affects RAPL- and NVML-based energy measurements. We evaluate 588 controlled NAS Parallel Benchmark runs across Intel and AMD CPUs and NVIDIA consumer-, workstation-, and data-center-class GPUs, using sampling intervals from 0.5 ms to 1 s. The primary experiment contains 504 runs across two platforms, three workload families, and three energy measurement interfaces (RAPL, NVML, external plug-level). Additional runs on latest-generation GPU platforms serve as supplemental validation. We compare energy readings at the 1 s interval against those at sub-second intervals, and cross-check power values derived from energy counters against directly reported power readings. The results reveal a fundamental asymmetry: NVML cumulative-energy counters show frequent stale reads below 100 ms with strong platform-dependent behavior, whereas RAPL package-plus-DRAM energy remains stable at 10 ms or coarser.

[1] https://developer.nvidia.com/management-library-nvml

[2] https://docs.codecarbon.io/latest/explanation/power-estimation/
[3] https://github.com/hubblo-org/scaphandre
[4] https://landscape.bundesverband-green-software.de/

Authors' Contact Information: Maximilian Dauner, maximilian.dauner0@hm.edu, Munich University of Applied Sciences HM, Munich, Germany; Manuel Steinberg, manuel.steinberg@hm.edu, Munich University of Applied Sciences HM, Munich, Germany; Andreas Brunnert, brunnert@hm.edu, Munich University of Applied Sciences HM, Munich, Germany; Benedikt Schicker, b.schicker@hm.edu, Munich University of Applied Sciences HM, Munich, Germany; Benedikt Zönnchen, zoennchen.benedikt@hm.edu, Munich University of Applied Sciences HM, Munich, Germany.

2 Related Work

Software energy measurement relies on hardware telemetry exposed through interfaces such as Intel RAPL and NVIDIA NVML. For CPUs, Khan et al. [10] validated RAPL against external power meters and found it useful for system-level estimation but subject to unpredictable update timing. Huang et al. [8] showed that polling more RAPL attributes increases measurement overhead, particularly under short workloads and high sampling rates. Raffin and Trystram [15] compared RAPL access mechanisms (powercap, perf-events, model-specific registers) and recommend adapting the acquisition rate to system load, but do not systematically evaluate how sampling interval affects energy stability. Weaver et al. [18] integrated power and energy measurement into PAPI and noted that coarse one-second sampling limits temporal resolution.

For GPUs, built-in sensors exhibit hardware-specific behavior that complicates energy measurement. Burtscher et al. [5] identified lag, tail energy, and inflated readings on NVIDIA K20 GPUs; Aslan and Yilmazer-Metin [3] reported similar lag and quantization effects on NVIDIA Jetson platforms. More recently, Yang et al. [20] described a part-time sampling effect on modern NVIDIA GPUs, where internal sensors only measure during part of the runtime.

Arafa et al. [2] show that energy profiles vary across GPU generations and can be obscured by driver abstractions, while Jay et al. [9] identify sampling interval and granularity as key differentiators between software-based power meters. In this context, Chung [6] recommends preferring NVML's cumulative-energy counter (two reads per workload) over polling and integrating instantaneous power, citing lower host overhead and no integration error. Prior work thus flags sampling interval as relevant but does not quantify its effect on cumulative-energy counter accuracy across RAPL and NVML. We close this gap, covering both cumulative-energy and power-derived estimates.

3 Experiment Setup

Our primary evaluation uses two representative platforms. The first is the data-center-oriented dual-socket Intel Xeon Gold 6326 server system with two NVIDIA A100 80 GB GPUs shown in Figure 1. The second is the consumer-class AMD Ryzen 9 5950X system with an NVIDIA RTX 4090 24 GB GPU shown in Figure 2. Total platform power is recorded using external energy meters (EMs), which are integrated directly into Rittal power distribution units (PDUs) in the server rack. A Shelly meter is used to record power usage between the power plug and the power supply unit (PSU) of the desktop. These energy and power measurements serve as an independent, system-level reference rather than as an isolated CPU or GPU ground truth.

To align our evaluation with prior work on RAPL access methods [15] and software-based CPU and GPU power meters [9], we use the C++ (NPB-PSTL[5]) and CUDA (NPB-GPU[6]) ports [12] of the NAS Parallel Benchmarks (NPB)[7]. We select three representative NPB workloads that cover distinct execution patterns. EP

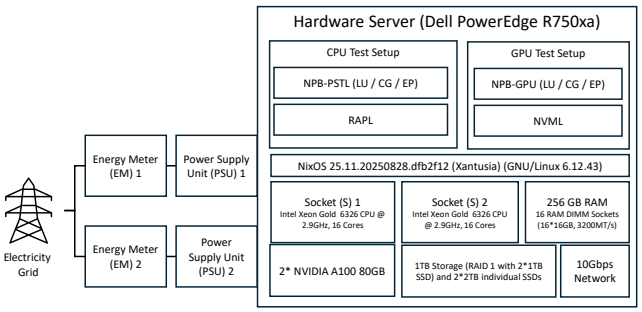


Fig. 1. Data-center-oriented server setup with two Intel Xeon Gold CPUs and two NVIDIA A100 GPUs.

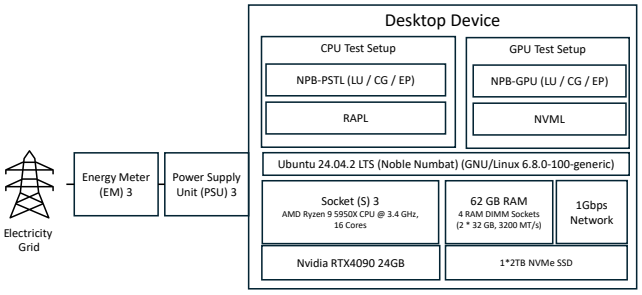


Fig. 2. Consumer-class setup with an AMD Ryzen CPU and an NVIDIA RTX 4090 GPU.

(Embarrassingly Parallel) generates Gaussian pseudorandom deviates via the Marsaglia polar method and accumulates 2D statistics; because it requires virtually no inter-process communication, it primarily exercises floating-point throughput and is compute-bound. CG (Conjugate Gradient) estimates the smallest eigenvalue of a sparse symmetric positive-definite matrix using the inverse power method with CG as the inner solver; it stresses irregular memory access, sparse matrix–vector multiplication, and cache locality [12]. LU (Lower-Upper Gauss-Seidel Solver) is a CFD-inspired pseudo application that solves the block lower- and upper-triangular systems from an unfactored implicit finite-difference discretization of the 3D Navier–Stokes equations; it combines compute demand with significant memory pressure.

NPB classes define benchmark problem sizes rather than a generic utilization level. We select classes separately for CPU and GPU runs to obtain sustained, sufficiently long, unthrottled workloads: CG uses class D on both CPU and GPU, LU uses class C on CPU and class D on GPU, and EP uses class D on CPU and class E on GPU. While these class choices offer problem sizes appropriate for stable energy measurement on different devices, they do not imply equal saturation of CPU, GPU, and memory resources. With these fixed benchmark-class configurations, the primary experiment combines the two platforms, two measurement interfaces, three benchmark families, and fourteen requested sampling intervals, each repeated three times, yielding 504 runs. CPU energy is read from RAPL package-plus-DRAM through powercap[8], and GPU telemetry

[5]https://github.com/GMAP/NPB-CPP
[6]https://github.com/GMAP/NPB-GPU
[7]https://www.nas.nasa.gov/software/npb.html
[8]https://www.kernel.org/doc/html/latest/power/powercap/powercap.html

from NVML [14] on the active benchmark GPU. As supplemental validation, we run the LU kernel on a consumer-class RTX 5060 Ti via VastAI[9] and a workstation-class RTX 6000 at our university's data center, covering all fourteen intervals with three repetitions and contributing 84 additional runs (588 in total). These supplemental runs test whether the NVML behavior observed on the RTX 4090 also appears on a latest-generation consumer GPU and whether a workstation-class RTX device follows the data-center-like pattern. Because they cover only LU and do not form a balanced full-factorial dataset, they are analyzed separately from the primary experiment.

The following fourteen sampling intervals were chosen to densely cover the expected lower-bound region around hardware and driver update periods (0.5–100 ms, where counter-update thresholds are most likely to appear [15, 20]) and to include coarser sentinel points (500 ms and 1 s) used by monitoring tools to test whether behavior has stabilized. We do not sample every point between 100 ms and 500 ms because the study focuses on the lower limit for reliable sampling rather than an exhaustive characterization of coarse-grained monitoring. We record both requested and achieved intervals because the measurement loop may not always achieve the requested interval at the short end.

$$0.5\,\text{ms},\ 1\,\text{ms},\ 10\,\text{ms},\ 20\,\text{ms}, \ldots,\ 90\,\text{ms},\ 100\,\text{ms},\ 500\,\text{ms},\ 1\,\text{s}.$$

Each run follows the same procedure. The automation framework (available in the replication package[10]) starts the external power reader and then the configured readers to read the RAPL and NVML data. A 10 s ramp-up precedes benchmark execution and a 10 s ramp-down follows it, after which the readers are stopped and the system idles for three minutes to limit thermal carry-over between configurations [7].

For analysis, the workload window is defined by the runtime parsed from the benchmark log. RAPL package and DRAM counters are corrected for hardware wraparound before energy deltas are computed. NVML cumulative energy is treated as a stepwise counter by subtracting the last counter value observed before the window start from the last counter value observed before the window end; energy readings at sub-second intervals are compared against the corresponding 1 s-window deltas, and per-interval power values derived from the energy counter are compared against NVML instantaneous power readings. For the stale-read analysis, we define a read as *stale* if the returned counter value is identical to the immediately preceding read, indicating that the hardware-visible energy counter has not updated within the sampling interval. External wall-power readings are compared against power values derived from RAPL and NVML energy and power sensors over the same window.

4 Experimental Results

Our main analysis is based on the balanced 504-run primary dataset, whereas the 84 additional runs on latest-generation NVIDIA GPUs serve as qualitative validation. Figures 3 and 4 structure the analysis: Figure 3 shows the self-consistency of RAPL and NVML across sampling intervals, and Figure 4 compares them to the external wall-power reference. All runs, including the supplemental ones, and the scripts to reproduce the figures are available in the replication package[10].

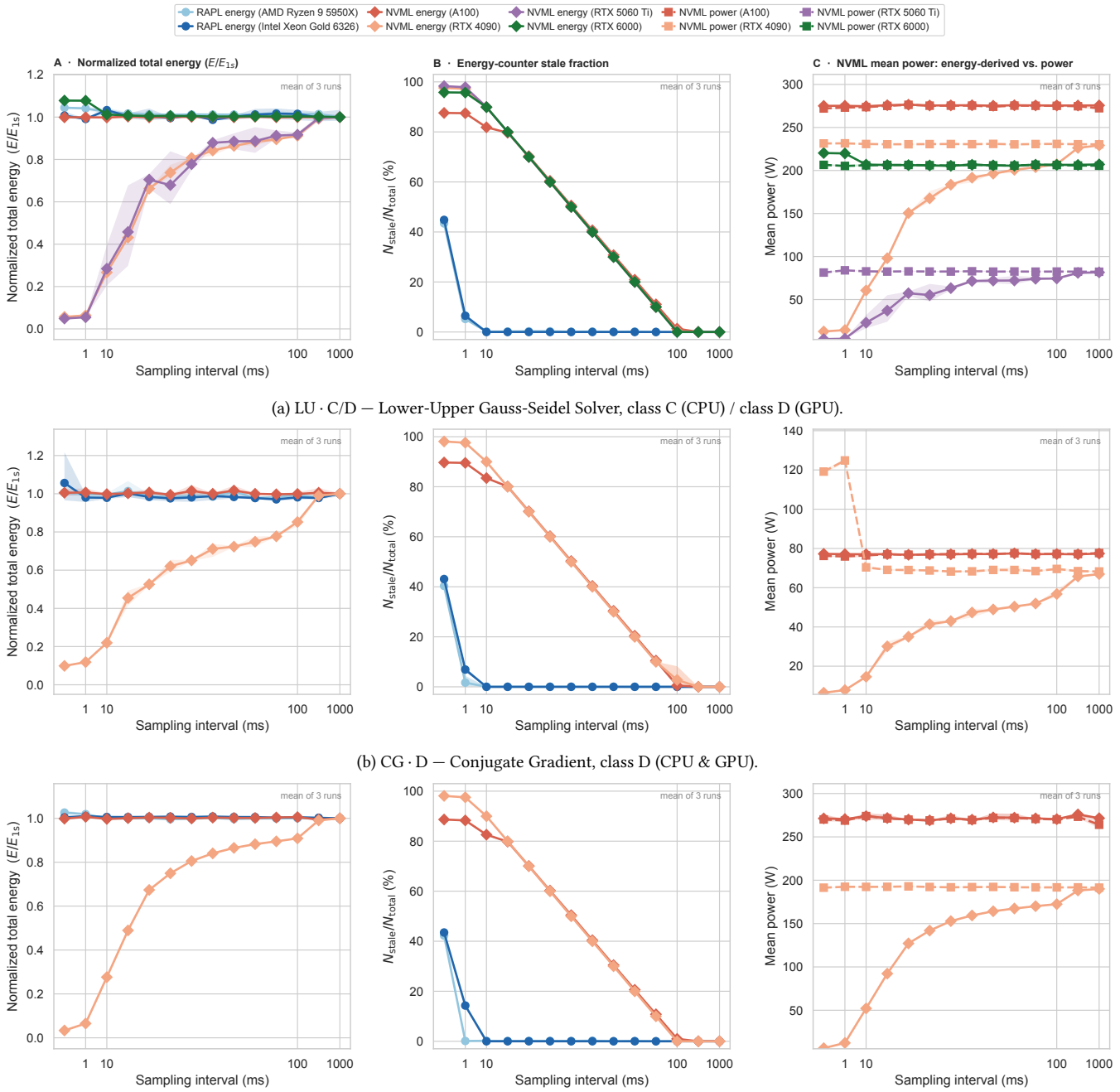
NVML stale reads expose an effective counter-update scale. The NVML cumulative-energy counter does not update faster than roughly 100 ms on any device tested as shown in panel B of Figure 3. Across the primary GPU runs, the mean stale-read fraction is 93.4 % at 0.5 ms, 86.3 % at 10 ms, and 79.9 % at 20 ms. It then drops to 1.0 % at 100 ms and reaches zero at 500 ms and beyond. Sampling below this scale therefore returns mostly repeated counter values rather than independent high-resolution samples.

The same stale-read pattern produces platform-dependent energy bias. Identical stale-read behavior does not translate into identical energy bias, as shown in panel A of Figure 3. On the data-center-class A100, the mean power derived from cumulative NVML energy differences, $\Delta E/\Delta t$, matches the directly reported NVML power readings within approximately ±0.2 % across all requested intervals (see panel C of Figure 3), despite substantial stale-read behavior in the cumulative energy counter. Stale reads here limit temporal resolution but not the aggregate delta. On the consumer-class RTX 4090, however, cumulative-energy-derived power no longer tracks integrated NVML power at short intervals. Over the full measurement window, the cumulative counter underestimates integrated NVML power by 95.4 % at 0.5 ms, 75.6 % at 10 ms, and 13.3 % at 100 ms, recovering to within 1.6 % only at 1 s. The mechanism is visible in the per-sample distribution. At fine intervals most NVML samples return ΔE=0 because the counter has not yet updated, while a few samples carry the entire accumulated energy as a single burst. This uneven distribution, rather than random noise, drives the underestimation in panel A of Figure 3 and the divergence between energy-derived mean power and instantaneous NVML power shown in panel C, binding the artifact to the sensor's update rate rather than to the requested sampling rate. Workloads with irregular power profiles, such as CG, remain more sensitive to where counter updates fall relative to workload boundaries. The supplemental consumer-class RTX 5060 Ti and workstation-class RTX 6000 follow the same stale-read curve and support the platform-dependent finding. The RTX 5060 Ti reproduces the RTX 4090 short-interval underestimation, whereas the RTX 6000 behaves closer to the A100 in full-window cumulative-energy bias.

NVML power telemetry is robust where the energy counter is not. NVML instantaneous power shows almost no monotonic association with requested sampling interval ($\rho = -0.005$ vs. $\log_{10}$ interval). As shown in panel C of Figure 3, mean reported power is stable for most platform–workload combinations, even when the cumulative energy counter underestimates by tens of percent. The exception is CG on the RTX 4090 at 0.5–1 ms, where reported mean power is elevated relative to coarser intervals while runtime is unchanged. Using NVML reported power draw is therefore the safer default on NVIDIA GPUs when the cumulative counter has not been validated, but very short power sampling intervals still warrant a workload- and platform-specific check.

RAPL is robust at 10 ms or coarser, but sub-millisecond sampling is not useful. After hardware-counter wraparound correction, RAPL

[9] https://vast.ai/
[10] https://github.com/hm-green-it-lab/hotc2026



(a) LU · C/D — Lower-Upper Gauss-Seidel Solver, class C (CPU) / class D (GPU).

(b) CG · D — Conjugate Gradient, class D (CPU & GPU).

(c) EP · D/E — Embarrassingly Parallel, class D (CPU) / class E (GPU).

Fig. 3. Measurement stability of RAPL and NVML across the NAS benchmark workloads. Lines show the mean across the three repetitions; shaded regions show the min–max range across those repetitions. Panel A shows total energy normalized to each sensor's 1 s reading. Panel B shows the stale-read fraction of cumulative energy counters, where a read is stale if it repeats the immediately preceding value. Panel C shows the difference between energy-derived power and NVML power. The primary results cover the Intel Xeon/A100 and AMD Ryzen/RTX 4090 platforms; for the LU benchmark, results for the RTX 5060 Ti and RTX 6000 are also included.

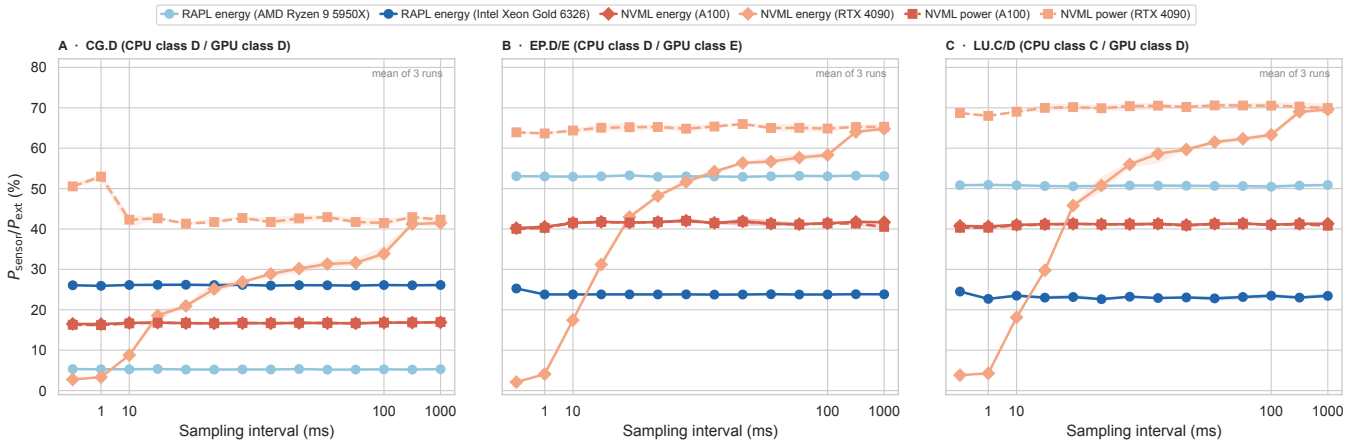


Fig. 4. Software-derived power relative to external wall power. RAPL and NVML cumulative-energy power are computed as workload energy divided by runtime. Lines show the mean across the three repetitions; shaded regions show the min–max range across those repetitions. NVML instantaneous power is shown as an additional telemetry path. Values below 100 % are expected because the external meter measures the complete platform, whereas RAPL and NVML report component-level telemetry.

package-plus-DRAM energy is stable for aggregate use. For sampling intervals of at least 10 ms, the mean absolute deviation from the platform- and benchmark-specific baseline is 0.57 % over the full measurement window and 0.69 % over the workload window. Representative full-window deviations are 0.51 % at 10 ms, −0.03 % at 100 ms, and −0.16 % at 1 s. At 0.5 ms, roughly half of all RAPL reads are stale (Figure 3 panel B), while the mean full-window aggregate energy shift is 6.1 % and the mean CPU runtime overhead reaches 2.4 %. At 1 ms, the stale-read fraction drops to 4.6 %, and the mean runtime overhead is about 0.9 %. By 10 ms, stale reads are reduced to 3.7 % and the overhead drops to 0.4 %. To evaluate whether this performance penalty develops into a continuous, long-term trend, we fitted a linear mixed-effects model across the entire span of evaluated intervals using the `statsmodels` Python library[11]. The sampling interval was modeled as the continuous fixed effect of interest, while hardware setup and benchmark type were included as random effects. The estimated effect of sampling interval on runtime was effectively zero and not statistically significant (coefficient $= -1.13 \times 10^{-4}$, $p = 0.996$). This high p-value confirms that the runtime overhead does not scale linearly across the broader spectrum of intervals; rather, the performance penalty is strictly asymptotic, manifesting exclusively at sub-millisecond frequencies. Consequently, outside of extreme sub-millisecond configurations, the additional measurement activity does not translate into a systematic or measurable runtime overhead across the evaluated benchmark types.

External wall power confirms the platform split. The EM comparison in Figure 4 confirms the NVML platform split against external wall power. EM measurements reflect complete-platform power, while RAPL covers CPU/DRAM and NVML only the active GPU. Therefore, it is expected that all component-level ratios will remain below 100 %. RAPL-derived power is stable across intervals, and NVML instantaneous power is similarly stable for most GPU scenarios. However, the power derived from the RTX 4090's cumulative energy data shows a different picture. On average, it reaches only 2.6 %, 3.7 %, and 13.5 % of wall power at 0.5 ms, 1 ms, and 10 ms, respectively. Meanwhile, NVML instantaneous power remains at 59.4 %, 58.9 %, and 54.6 %, respectively. Only at coarse intervals does the cumulative path recover, reaching 53.0 % at 500 ms and 53.7 % at 1 s, close to the corresponding NVML power ratios of about 55.6 %.

Implications for energy attribution. Two practical recommendations follow. First, NVIDIA GPU energy attribution should avoid unvalidated NVML cumulative-energy counters, especially at short intervals on consumer-class GPUs and for workloads whose boundaries are sensitive to the counter-update phase. Integrating NVML power is the safer default when power readings are stable for the target workload. Second, RAPL should not be polled at sub-millisecond intervals for aggregate accounting. Once wraparound is corrected, intervals of 10 ms or coarser preserve aggregate energy without redundant reads or unnecessary measurement overhead. The asymmetry between NVML's two paths is consistent with their differing platform support. NVIDIA documents `nvmlDeviceGetPowerUsage` for Fermi and newer devices, but `nvmlDeviceGetTotalEnergyConsumption` only for Volta and newer [14], indicating that the cumulative-energy interface has narrower and more recent hardware coverage. This narrower hardware coverage suggests that the cumulative-energy interface is a more recent addition to NVML, and its relative immaturity compared to the long-established power telemetry path may partly explain the stronger sampling-interval sensitivity we observe in our experiments.

5 Limitations and Threats to Validity

Our evaluation is bounded by the following factors. First, we test two CPU and four GPU models from Intel, AMD, and NVIDIA, covering consumer-class RTX 4090/RTX 5060 Ti, workstation-class

[11] https://www.statsmodels.org/stable/index.html

RTX 6000, and data-center-class A100 devices. The observed NVML behavior may not generalize beyond these specific generations, and AMD GPU telemetry via ROCm SMI is out of scope. Second, NPB exercises representative compute-, memory-, and mixed-bound patterns but not I/O- or network-dominated workloads, where sampling sensitivity may differ. Readers reproducing our results on a different environment may observe different stale-read thresholds, although our data consistently separates consumer-class behavior from workstation- and data-center-class behavior. NVML, firmware, and sensor internals are not documented at a level that would let us isolate whether the observed behavior stems from hardware counters, firmware aggregation, driver buffering, or API semantics.

6 Conclusion and Future Work

We quantified how sampling interval affects RAPL and NVML across 588 controlled NAS Parallel Benchmark runs on Intel and AMD CPUs and four NVIDIA GPUs. RAPL energy is stable for aggregate accounting at 10 ms or coarser, while sub-millisecond sampling adds runtime overhead without improving accuracy. NVML behavior depends strongly on device class: on the consumer-class devices (RTX 4090/RTX 5060 Ti), cumulative energy underestimates integrated NVML power by up to 95.4 % at 0.5 ms due to infrequent counter updates, whereas NVML instantaneous power remains stable across the interval range. On workstation- and data-center-class GPUs, such as the RTX 6000 and A100, cumulative energy remains consistent across all requested intervals despite many stale reads. Energy and carbon attribution tools should therefore validate NVML cumulative-energy counters per device, prefer NVML power on consumer-class hardware, and avoid sub-millisecond RAPL polling. Future work will extend the evaluation to AMD GPU telemetry via ROCm SMI, non-NPB workloads, and additional CPU/GPU generations to test the robustness of the observed platform-dependent threshold.

References

[1] Marcelo Amaral, Huamin Chen, Tatsuhiro Chiba, Rina Nakazawa, Sunyanan Choochotkaew, Eun Kyung Lee, and Tamar Eilam. 2023. Kepler: A framework to calculate the energy consumption of containerized applications. In *2023 IEEE 16th International Conference on Cloud Computing (CLOUD)*. 69–71. doi:10.1109/CLOUD60044.2023.00017

[2] Yehia Arafa, Ammar ElWazir, Abdelrahman ElKanishy, Youssef Aly, Ayatelrahman Elsayed, Abdel-Hameed Badawy, Gopinath Chennupati, Stephan Eidenbenz, and Nandakishore Santhi. 2020. Verified instruction-level energy consumption measurement for NVIDIA GPUs. In *Proceedings of the 17th ACM International Conference on Computing Frontiers* (Catania, Sicily, Italy) *(CF '20)*. Association for Computing Machinery, New York, NY, USA, 60–70. doi:10.1145/3387902.3392613

[3] Büşra Aslan and Ayse Yilmazer-Metin. 2022. A study on power and energy measurement of NVIDIA Jetson embedded GPUs using built-in sensor. In *2022 7th International Conference on Computer Science and Engineering (UBMK)*. 1–6. doi:10.1109/UBMK55850.2022.9919522

[4] Andreas Brunnert. 2025. Evaluating the accuracy of software energy consumption models for Java applications at process and transaction levels. In *Proceedings of the 33rd ACM International Conference on the Foundations of Software Engineering* (Clarion Hotel Trondheim, Trondheim, Norway) *(FSE Companion '25)*. Association for Computing Machinery, New York, NY, USA, 1441–1448. doi:10.1145/3696630.3728709

[5] Martin Burtscher, Ivan Zecena, and Ziliang Zong. 2014. Measuring GPU power with the K20 built-in sensor. In *Proceedings of Workshop on General Purpose Processing Using GPUs* (Salt Lake City, UT, USA) *(GPGPU-7)*. Association for Computing Machinery, New York, NY, USA, 28–36. doi:10.1145/2588768.2576783

[6] Jae-Won Chung. 2023. *Measuring GPU energy: Best practices*. ML.ENERGY Blog, Accessed: 2026-05-11. https://ml.energy/blog/energy/measurement/measuring-gpu-energy-best-practices/

[7] Luís Cruz. 2021. Green software engineering done right: A scientific guide to set up energy efficiency experiments. http://luiscruz.github.io/2021/10/10/scientific-guide.html. Blog post, Accessed: 2026-05-11. doi:10.6084/m9.figshare.22067846.v1

[8] Song Huang, Michael Lang, Scott Pakin, and Song Fu. 2015. Measurement and characterization of Haswell power and energy consumption. In *Proceedings of the 3rd International Workshop on Energy Efficient Supercomputing* (Austin, Texas) *(E2SC '15)*. Association for Computing Machinery, New York, NY, USA, Article 7, 10 pages. doi:10.1145/2834800.2834807

[9] Mathilde Jay, Vladimir Ostapenco, Laurent Lefevre, Denis Trystram, Anne-Cécile Orgerie, and Benjamin Fichel. 2023. An experimental comparison of software-based power meters: focus on CPU and GPU. In *2023 IEEE/ACM 23rd International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. 106–118. doi:10.1109/CCGrid57682.2023.00020

[10] Kashif Nizam Khan, Mikael Hirki, Tapio Niemi, Jukka K. Nurminen, and Zhonghong Ou. 2018. RAPL in action: Experiences in using RAPL for power measurements. *ACM Trans. Model. Perform. Eval. Comput. Syst.* 3, 2, Article 9 (March 2018), 26 pages. doi:10.1145/3177754

[11] Jens Lang and Gudula Rünger. 2013. High-resolution power profiling of GPU functions using low-resolution measurement. In *Euro-Par 2013 Parallel Processing*, Felix Wolf, Bernd Mohr, and Dieter an Mey (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 801–812.

[12] Júnior Löff, Dalvan Griebler, Gabriele Mencagli, Gabriell Araujo, Massimo Torquati, Marco Danelutto, and Luiz Gustavo Fernandes. 2021. The NAS Parallel Benchmarks for evaluating C++ parallel programming frameworks on shared-memory architectures. *Future Generation Computer Systems* 125 (2021), 743–757. doi:10.1016/j.future.2021.07.021

[13] Adel Noureddine. 2022. PowerJoular and JoularJX: Multi-platform software power monitoring tools. In *2022 18th International Conference on Intelligent Environments (IE)*. Biarritz, France, 1–4. doi:10.1109/IE54923.2022.9826760

[14] NVIDIA Corporation. 2026. NVIDIA Management Library (NVML) API reference: Device queries. https://docs.nvidia.com/deploy/nvml-api/group__nvmlDeviceQueries.html. Accessed: 2026-04-07. Documents `nvmlDeviceGetPowerUsage` (Fermi+) and `nvmlDeviceGetTotalEnergyConsumption` (Volta+).

[15] Guillaume Raffin and Denis Trystram. 2025. Dissecting the software-based measurement of CPU energy consumption: A comparative analysis. *IEEE Transactions on Parallel and Distributed Systems* 36, 1 (2025), 96–107. doi:10.1109/TPDS.2024.3492336

[16] Shivadharshan S, Akilesh P, Rajrupa Chattaraj, and Sridhar Chimalakonda. 2024. CPPJoules: An Energy Measurement Tool for C++. arXiv:2412.13555 [cs.SE] https://arxiv.org/abs/2412.13555

[17] Rubén Saborido, Venera Arnaoudova, Giovanni Beltrame, Foutse Khomh, and Giuliano Antoniol. 2015. On the impact of sampling frequency on software energy measurements. *PeerJ Preprints* (2015). doi:10.7287/peerj.preprints.1219v1

[18] Vincent M. Weaver, Matt Johnson, Kiran Kasichayanula, James Ralph, Piotr Luszczek, Dan Terpstra, and Shirley Moore. 2012. Measuring energy and power with PAPI. In *2012 41st International Conference on Parallel Processing Workshops*. 262–268. doi:10.1109/ICPPW.2012.39

[19] Benjamin Weigell, Simon Hornung, and Bernhard Bauer. 2026. METRION: A framework for accurate software energy measurement. arXiv:2512.06806 [cs.SE] https://arxiv.org/abs/2512.06806

[20] Zeyu Yang, Karel Adamek, and Wesley Armour. 2024. Accurate and convenient energy measurements for GPUs: A detailed study of NVIDIA GPU's built-in power sensor. In *SC24: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–17. doi:10.1109/SC41406.2024.00028