Containers Are (Almost) Free: Energy and Performance Analysis of AI Workloads under CPU Virtualization and GPU Power Capping

Michał Kościowski[0009-0004-0953-0697][1], Jerzy Proficz[0000-0003-2975-9339][2], and Paweł Czarnul[0000-0002-4918-9196][1]

[1] Faculty of Electronics, Telecommunications and Informatics
[2] Centre of Informatics — Tricity Academic Supercomputer and networK (CI TASK)
Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland
michal.kosciowski@pg.edu.pl, j.proficz@task.gda.pl, pawel.czarnul@pg.edu.pl

Abstract. In contemporary computing, we can observe simultaneous: adoption of virtualization that has become mainstream, popularity of AI workloads, as well as very considerable increase of energy costs. The latter result from high prices and huge requirements of the aforementioned workloads. In this context, we present a performance- and energy-aware comparison of virtual machines and containers against bare metal using three modern AI workloads: Single Shot Detector (SSD) from MLPerf Inference, and two PyTorch benchmarks – ResNet50 training and hf_BERT inference – both in 32-bit floating-point precision. Compared to the state-of-the-art, we contribute by: using a multi(8)-GPU environment, considering another variable being a power cap imposed on each of the GPUs, and assessment of not only resulting execution times and energy consumption but also the Energy Delay Product (EDP) metric. We found that Docker has a negligible impact on performance (0.03% on average), while KVM causes a performance drop of 21.22% on average compared to bare metal. For energy consumption overheads for Docker and KVM are -0.03% and 1.63%, while for EDP 0% and 23.14%, respectively. We have also determined that power caps that minimize EDP are different for KVM and Docker/bare metal, which is an important observation for optimization of that essential performance-energy trade-off.

Keywords: Energy-aware computing · Virtual machines · Containers · Power capping · AI workloads

1 Introduction

Modern AI workloads require computationally intensive GPU processing. Such computations are performed on servers in data centers, which can have power demands reaching hundreds of megawatts. Reducing energy consumption can lead to significant savings.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

The main motivation of this work is to provide a performance and energy-aware comparison of virtual machine (VM) and container based environments against a bare metal system for contemporary AI workloads. Furthermore, compared to the existing works on comparative assessment of such environments, we contribute by providing comparison for:

1. a parallel workload running in a multi-GPU environment, specifically run on a machine with eight NVIDIA QUADRO RTX6000 GPUs,
2. various power caps imposed on the GPUs,
3. not only execution time but also energy and EDP metrics [6], for power caps in the available range.

The results indicate that KVM introduces average overheads of 1.63% in energy consumption (up to 2.71%) and 21.22% in execution time (up to 36.49%), while container-based systems remain at a very low overhead level, with average and maximum values of -0.03% and 0.26% for energy, and 0.03% and 0.31% for execution time, respectively.

The following section reviews related work. Next, we present the motivation and main contributions of the paper, followed by a description of the measurement methodology, the experimental setup and results. The paper is concluded with final remarks and directions for future work.

2 Related work

In the following articles, the authors examine the impact of virtualization on GPU and CPU performance in HPC, rack server, desktop, SoC, grid, and cloud environments, as well as research findings and state-of-the-art tools that improve energy efficiency and execution time.

In work [8], MPI applications are evaluated on CPUs using Docker and bare metal. Experiments show Docker overheads of 7.59%, 13.29%, and 15.3%, for 64, 128 and 256 application processes respectively.

Authors of paper [4] compare the performance of Docker containers and KVM VMs using bare metal as reference. Benchmarks covering CPU, memory, storage, and network performance on Linux show that Docker incurs up to 2% overhead, while KVM overhead reaches up to 40% in I/O workloads.

In paper [13], Docker containers are compared with bare metal. CPU overhead for HPL is below 1% while for HPCG it is 0.244% and 0.556%, for desktop and server-grade, respectively. GPU tests report a performance difference of 0.61% and 1.72%, respectively. Sequential I/O tests show differences of 0.636% and 3.448%, random I/O performance is up to 100% higher on container thanks to use of Multi-Tier Cache Technology.

Authors of work [2] evaluate ease of deployment and performance of KVM- and VirtualBox-based VMs and Docker containers against bare metal. The start-up time ranges from 49–59 s and 39–41 ms; CPU performance shows 27% degradation and no difference for VMs and containers, respectively. I/O and network

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

bandwidth overheads are highest for VMs, while container shows moderate overhead.

In paper [5], authors evaluate VMs (Xen, KVM, VirtualBox, VMware) and containers (Docker, Podman) against bare metal and OpenStack bare metal on an OpenStack-managed cluster. A Linux kernel compilation test shows overheads of 2.0–2.3% for containers, synthetic benchmarks 0.1–3.0% and while I/O tests are up to 8.7% better than bare metal. For VMs, CPU performance and network bandwidth is reduced by 0.2–9.3% and 0.4–24.8%,respectively. Synthetic memory tests for Xen and KVM exceed bare metal by 0.4% and 5.5%.

In work [11], authors evaluate virtualization overhead on an ARM-based Cubieboard2 SoC, comparing Docker containers vs KVM VMs against bare metal. The results demonstrate CPU overheads of 0.1% vs 4.27% (NBench), 0.25% vs 1.17% (Sysbench) and 34.77% vs 991.65% (Linpack); storage I/O overheads of 3.1% vs 66.6% (Bonnie++) and 1.4% vs 81.15% (dd); memory overheads of 0.28% vs 2.54% (STREAM); and network overheads of 2.8% vs 22.9% (TCP) and 0.6% vs 18% (UDP) for containers vs VMs.

In paper [1], authors review optimization metrics and methods for controlling energy consumption in systems based on single devices, grids, clusters and cloud solutions. They analyze possibilities offered by CPUs and GPUs and provide a comprehensive analysis of methods for optimizing power and energy consumption using schedulers, DVFS/DFS/DCT and power capping using APIs provided by hardware manufacturers such as Intel RAPL, NVIDIA NVML, AMD APM, and IBM EnergyScale.

In work [10], authors propose an automatic GPU energy profiling approach based on the NVIDIA NVML API and introduce the EnergyProfiler tool, which measures energy consumption, execution time, and average power. By evaluating multiple power limits, the authors managed to save up to 30% of energy with a 12% drop in performance.

In paper [9], authors analyze the performance-energy trade-off in training of Deep Convolutional Neural Networks for image recognition on NVIDIA Quadro RTX6000 and V100 GPUs. Using the NVIDIA NVML API for power monitoring and power capping, they minimize Energy, EDP, and EDS. Energy savings reach 28.5–32.5% and 24–33%, EDP 25–28% and 23–27%, and EDS 22–27% and 23.5–27.3%, with performance losses of 4.5–15.4% and 4.5–21% for RTX6000 and V100, respectively.

3 Experiments and testbed environment

In this work, we used the StEP (Static Energy Profiler) tool from the SPLiT [3] package setup for NVIDIA GPUs. The StEP tool was presented and discussed in more detail in paper [9]. For each test, a power cap was set separately before execution using the StEP tool. Power monitoring was also performed using StEP. Preliminary tests showed that the overhead of such monitoring is well below 1%, amounting to 0.18% of CPU utilization.

[3] https://projects.task.gda.pl/akrz/split

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

We used three different benchmarks. Single Shot Detector (SSD) [4] used in MLPerf Inference Benchmarks from the MLCommons reference implementation, based on ResNeXt50_32x4d [12] and subset of OpenImagesV7 [5] reduced to 12 classes. Two benchmarks from PyTorch benchmark suite [6]: image recognition network ResNet50 [7] training benchmark and pretrained transformer from Hugging Face – hf_Bert [3] inference benchmark, 32-bit floating point precision. We selected benchmarks so that the GPU load would be as high as possible throughout the entire duration of the tests.

We test the impact of the virtualization environment in the form of Docker and VM running in KVM vs bare metal using power capping thanks to the StEP tool. Docker tests include 8 parallel containers, KVM tests include 8 parallel virtual machines. Each environment has it own GPU. Bare metal tests are run as 8 parallel processes. We compare training time, energy and EDP at GPU power set to $P_{cap} = 100 + 8 \times n, \forall n \in \{1, \cdots, 20\}$ on each GPU. Each test for every configuration was run 10 times.

We conducted the tests on a rack server machine equipped with 2 Intel Xeon Silver 4210 CPUs, 384 GB RAM and 8 NVIDIA Turing QUADRO RTX6000 GPUs. The operating system used during testing on bare metal and as host system was Ubuntu 22.04.3 LTS with CUDA Toolkit 12.2, Python 3.10 and PyTorch 2.7.1. The software versions are the standard versions available in the OS being used. The selected LTS version of the operating system is a stable and proven distribution widely used in commercial solutions, thanks to its 5-year support period, which can be extended to 10–15 years with the Pro version. The same software configuration was used in VMs and containers. For the VMs, we allocated 32 GB of RAM and 8 CPU cores to each machine, the driver used to communicate with the GPU was vfio-pci.

4 Results and analysis

The results can be divided into three sets, all presented in the context of power limits imposed by power-capping mechanisms. The first set concerns workload performance, where execution times (T) were measured. The second set enables direct comparison of energy consumption (E). The third set represents a trade-off between these two aspects and is expressed using the Energy-Delay Product, $EDP = E \cdot T$. These metrics were averaged over individual workloads running on the different GPUs; each workload was executed 10 times and the results were averaged, and such values are presented next for various power constraints.

The observed execution times differ across the compared configurations; however, the container-based approach is only marginally slower than bare metal execution, with average and maximum overheads of 0.19% and 0.31%, respectively. For the SSD benchmark, the VM configuration performs significantly worse,

[4] https://github.com/mlcommons/training/tree/master/retired_benchmarks/ssd-v1/ssd
[5] https://storage.googleapis.com/openimages/web/download_v7.html
[6] https://github.com/pytorch/benchmark

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

exhibiting average and maximum overheads of 3.84% and 4.64%, respectively. As for the results related to the energy consumption, similarly to the performance data, the container-based configuration exhibits only a marginal increase in energy usage compared to bare metal execution, with average and maximum overheads of 0.12% and 0.26%, respectively. In contrast, the VM-based execution performs noticeably worse, showing average and maximum energy overheads of 1.78% and 2.51%, respectively. Concerning EDP, we observe that the container-based configuration continues to exhibit very low overheads, with average and maximum values of 0.31% and 0.45%, respectively. In contrast, the VM-based configurations perform significantly worse, showing average and maximum overheads of 5.67% and 7.25%, respectively. In the case of the hf_Bert benchmark, the overheads observed for the container-based configuration are practically imperceptible. Workload execution times are on average 0.02% faster and at most 0.1% slower than bare metal. Average energy consumption was 0.08% lower for container-based, while the maximum was 0.07% higher than bare metal. VM-based configurations are significantly worse. The execution time overhead averages at 28.9%, with a maximum of 36.47%, compared to bare metal. The energy consumption is not significantly higher, averaging at 1.45% and reaching a maximum of 2.06%. The EDP, calculated in the same way as in previous test, also shows an average decrease of 0.1% and a maximum increase of 0.15% for container-based configuration compared to bare metal. In the case of VM, EDP is significantly higher due to the long duration of individual workloads, averaging 30.74% and a maximum of 38.91% compared to bare metal. A very similar situation can also be observed for the resnet50 benchmark. The duration of the container-based workload is on average 0.08% shorter and at a maximum 0.27% longer, while the energy consumption is on average 0.14% smaller and at a maximum 0.21% higher than for bare metal. EDP, calculated in the same way as in previous tests, is on average 0.22% lower and up to 0.5% higher compared to bare metal. The VM also performs worse in this test. The workload duration has an overhead of 30.91% on average and 36.49% at a maximum. Energy consumption is 1.65% higher on average and 2.71% at a maximum. EDP is 33.02% higher on average and 39.42% at a maximum compared to bare metal. In all cases, we noticed that the time required for IO operations had a significant impact on the execution time, which confirms the observations noted in work [4]. Detailed results and charts are available in the repository [7].

Charts in Fig. 1 present the aforementioned results in a chart-based representation. The plots confirm that the overhead introduced by the container-based configuration is negligible overall. In contrast, the VM-based execution results in significantly higher overheads for both energy and performance.

Moreover, the gap observed for EDP is clearly larger than for its individual components considered separately. We can also observe that execution under lower power caps tends to be more unstable, and that the performance and energy differences between the VM and the bare metal/container configurations are smaller when lower power limits are imposed.

[7] https://kask.eti.pg.gda.pl/gitlab/mickosc2/iccs2026

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

Concluding the above results, we can note that the influence of containerization on GPU-based AI workloads is negligible, in contrast to the VM-based configuration, where the performance and energy penalties are significant, although they decrease under more aggressive power caps. This demonstrates that CPU-side virtualization overhead remains visible even for highly GPU-dominated AI workloads.

Therefore, we strongly recommend using container-based configurations deployed directly on bare metal, without an additional VM layer, wherever such an approach is feasible and does not conflict with higher-priority constraints, such as security policies.

5 Summary and future work

In this paper, we provided a comparison of execution of AI workloads in Virtual Machine (KVM) and containerized (Docker) environments against a bare metal system, analyzing metrics such as execution time, energy consumption and Energy-Delay Product (EDP). Furthermore, we performed comparative assessment in a multi 8 GPU system for a range of power caps imposed on the GPUs, starting from the default value of 260 W per GPU down to 108 W per GPU. We conclude that the overhead of Docker versus bare metal is virtually negligible – with average and maximum values across power caps tested: 0.03% and 0.31% for the execution time, -0.03% and 0.26% for energy and 0% and 0.5% for EDP. For KVM, we observe visibly larger penalties of 21.22% and 36.49% for the execution time, 1.63% and 2.71% for the energy consumption and 23.14% and 39.42% for EDP. An interesting observation is that these penalties become smaller for more restrictive (smaller) power caps.

Within future work, we plan to extend this research by testing additional virtualization solutions – both VM-based and container-based ones – such as Xen, VMware, and Podman, Singularity, across a broader range of systems and workloads. We will also explore the impact of software and hardware configurations, including different versions of components from the compiler and CUDA to AI frameworks, as well as diverse computing platforms and GPUs such as NVIDIA A100, H100, and AMD GPUs.

Acknowledgments. The research was supported in part by project "Cloud Artificial Intelligence Service Engineering (CAISE) platform to create universal and smart services for various application areas", No. KPOD.05.10-IW.10-0005/24, as part of the European IPCEI-CIS program, financed by NRRP (National Recovery and Resilience Plan) funds. The computations were performed at the Department of Computer Architecture, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology. During the preparation of this work, the authors used LLM to improve the language, grammar, and general text flow of the manuscript. Following the use of this tool, the authors reviewed and edited the content as needed and take full responsibility for the final content of the publication.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

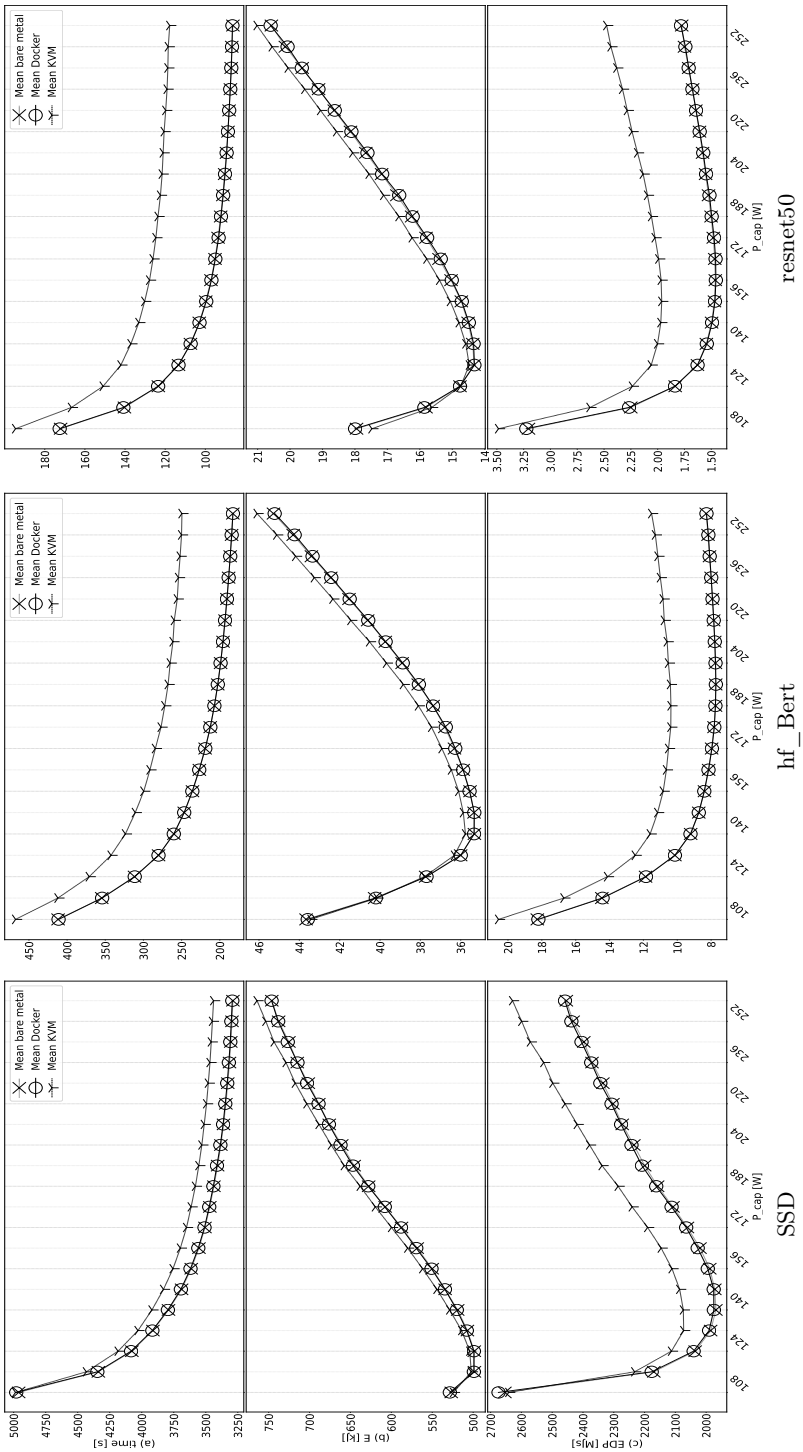

Fig. 1: Experimental results for SSD, hf_Bert and resnet50 benchmarks, (a): execution time [s], (b): energy consumption [kJ], (c) Energy–Delay Product (EDP) [MJs] under different power caps.

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42

References

1. Czarnul, P., Proficz, J., Krzywaniak, A.: Energy-Aware High-Performance Computing: Survey of State-of-the-Art Tools, Techniques, and Environments. Scientific Programming pp. 1–19 (Apr 2019). https://doi.org/10.1155/2019/8348791
2. Deochake, S., Maheshwari, S., De, R., Grover, A.: Comparative Study of Virtual Machines and Containers for DevOps Developers (Apr 2023). https://doi.org/10.48550/arXiv.1808.08192
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (May 2019). https://doi.org/10.48550/arXiv.1810.04805
4. Felter, W., Ferreira, A., Rajamony, R., Rubio, J.: An updated performance comparison of virtual machines and Linux containers. In: 2015 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS). pp. 171–172. IEEE (Mar 2015). https://doi.org/10.1109/ISPASS.2015.7095802
5. Giallorenzo, S., Mauro, J., Poulsen, M.G., Siroky, F.: Virtualization Costs: Benchmarking Containers and Virtual Machines Against Bare-Metal. SN Computer Science p. 404 (Sep 2021). https://doi.org/10.1007/s42979-021-00781-8
6. Gonzalez, R., Horowitz, M.: Energy dissipation in general purpose microprocessors. IEEE Journal of Solid-State Circuits p. 1277–1284 (Sep 1996). https://doi.org/10.1109/4.535411
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (Dec 2015). https://doi.org/10.48550/arXiv.1512.03385
8. Kononowicz, T., Czarnul, P.: Performance Assessment of Using Docker for Selected MPI Applications in a Parallel Environment Based on Commodity Hardware. Applied Sciences p. 8305 (Jan 2022). https://doi.org/10.3390/app12168305
9. Krzywaniak, A., Czarnul, P., Proficz, J.: GPU Power Capping for Energy-Performance Trade-Offs in Training of Deep Convolutional Neural Networks for Image Recognition. In: Computational Science – ICCS 2022. pp. 667–681. Springer International Publishing (2022). https://doi.org/10.1007/978-3-031-08751-6_48
10. Krzywaniak, A., Czarnul, P.: Performance/Energy Aware Optimization of Parallel Applications on GPUs Under Power Capping. In: Parallel Processing and Applied Mathematics, pp. 123–133. Springer International Publishing (2020). https://doi.org/10.1007/978-3-030-43222-5_11
11. Ramalho, F., Neto, A.: Virtualization at the network edge: A performance comparison. In: 2016 IEEE 17th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM). pp. 1–6. IEEE (Jun 2016). https://doi.org/10.1109/WoWMoM.2016.7523584
12. Xie, S., Girshick, R., Dollár, P., Tu, Z., He, K.: Aggregated Residual Transformations for Deep Neural Networks (Apr 2017). https://doi.org/10.48550/arXiv.1611.05431
13. Xu, P., Shi, S., Chu, X.: Performance Evaluation of Deep Learning Tools in Docker Containers (Nov 2017). https://doi.org/10.48550/arXiv.1711.03386

ICCS Camera Ready Version 2026
To cite this paper please use the final published version:
DOI: 10.1007/978-3-032-29924-6_42