VSB TECHNICAL UNIVERSITY OF OSTRAVA | IT4INNOVATIONS NATIONAL SUPERCOMPUTING CENTER
VSB TECHNICAL UNIVERSITY OF OSTRAVA | FACULTY OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

HPCSE 2026
High Performance Computing in Science and Engineering

Energy Efficiency Runtime System for GPUs

Daniel Velička[1,2], Ondřej Vysocký[1], Lubomír Říha[1]

1 IT4Innovations National Supercomputing Center, VSB-TUO

2 Faculty of Electrical Engineering and Computer Science, VSB-TUO

As HPC systems and large-scale AI data centers continue to expand, energy consumption has increased substantially, driven largely by intensive GPU usage. To address this challenge, numerous techniques have been proposed to improve GPU utilization and, consequently, energy efficiency. Many of these approaches rely on runtime systems that dynamically adjust hardware parameters—such as GPU frequency—based on observed GPU performance counters, often without accounting for the latency between configuration changes and their observable effects, or modeling it as a fixed constant.

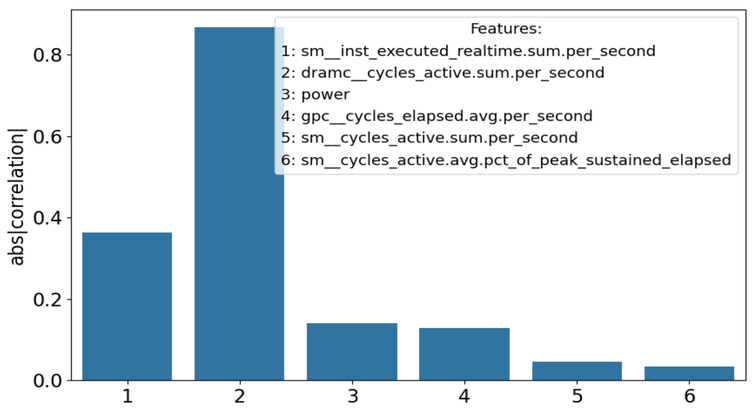
In this work, we present a GPU energy efficiency runtime system which utilizes the sampling of GPU performance counters to enable fine-grained, application-agnostic energy optimization. The design of this runtime system is based on the insights obtained using LATEST, the implementation of methodology for analyzing GPU frequency switching behavior. LATEST measures the reactivity of GPU to frequency change requests (switching latency), and the length of the frequency change (transition latency). This awareness of frequency switching latency enables frequency adjustments to be performed as often as permitted by the switching latency, while minimizing associated overhead.

Energy Efficiency Runtime System for GPUs (CUDA ENRUNTIME)

We introduce an open-source Energy Efficiency Runtime System designed for CUDA GPUs. The runtime system utilizes CUPTI (CUDA Profiling Tools Interface) PM (Performance Metric) Sampling API to estimate the arithmetic intensity. The optimal streaming multiprocessor frequency for the arithmetic intensity estimate is then set through the NVML (Nvidia Management Library) using a specialized daemon tool.

Performance Metric Analysis

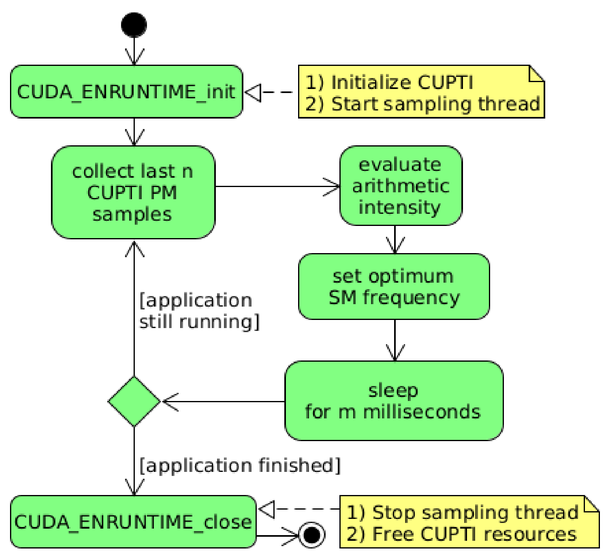
To enable real-time streaming multiprocessor (SM) frequency tuning based on current workload characteristics, correlation analysis with arithmetic intensity was performed on a set of CUPTI PM Sampling API metrics. These metrics were obtained from the artificial mandelbrot benchmarks with predetermined arithmetic intensities.



Correlations of selected CUPTI metrics against the arithmetic intensity

Runtime System Methodology

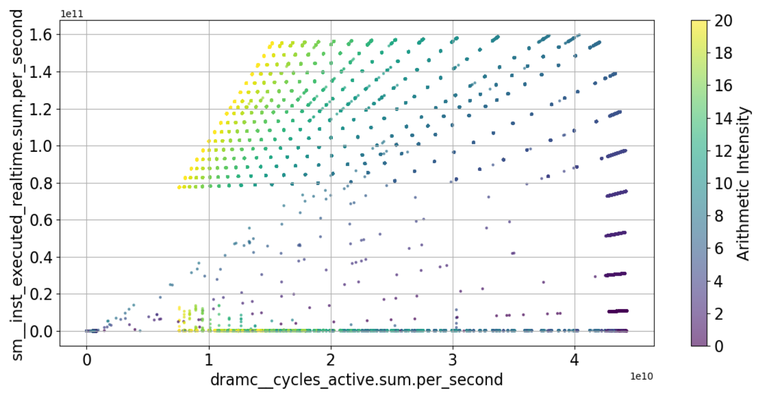
Real-world CUDA applications, aside artificial benchmarks, often include a diverse set of CUDA kernels. Their execution times can range from a few microseconds to several tens of milliseconds. The LATEST research conducted earlier discovered, that the SM frequency switching latency often spans longer (up to 25 ms in A100 SXM-4). Additionally, tracing of the start and the end of each CUDA kernel appears to be challenging, especially in applications using multiple CUDA streams. Thus, an interval-based approach was adopted for both performance assessment and SM frequency tuning.



CUDA ENRUNTIME workflow loop

Arithmetic Intensity Model

Exploratory analysis of the artifical benchmark performance metrics revealed compact clusters of the metrics samples from the same arithmetic intensity benchmarks. We use this structure to create a lightweight classification model for the arithmetic intensity based on the performance metrics.



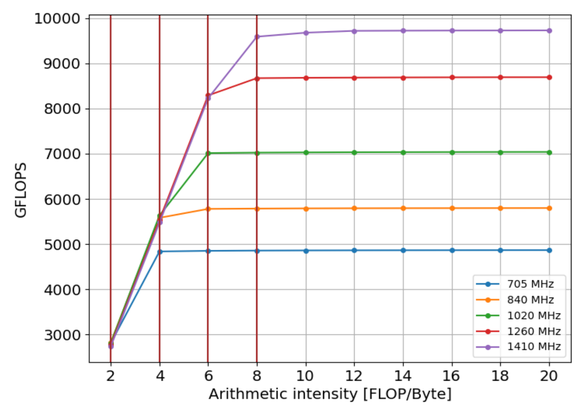
Observed arithmetic intensity patterns in the highest-correlation metrics

Roofline-aware SM Frequency tuning

Roofline-Aware SM frequency tuning aims to achieve energy savings while mitigating the performance penalty as well.

Arithmetic intensity (x) [FLOP/B]	SM freq. optimum [MHz]
x < 2.0	705
2.0 <= x < 4.0	840
4.0 <= x < 6.0	1020
6.0 <= x < 8.0	1260
8.0 <= x	1410

Frequency levels used on A100 SXM-4 according to workload arithmetic intensity



The rooflines (A100) for the frequency levels utilized in the runtime system - vertical lines indicate the arithmetic intensity thresholds

Results (Nvidia A100 SXM-4)

Mandelbrot (2 AIs, 2 s period)	Energy [kJ]	Execution time [s]
A100 default - 1410 MHz	10.4	30.0
CUDA ENRUNTIME dynamic tuning	8.6 (-17.1%)	30.7 (+2.2%)
Karolina default – 1290 MHz	8.3 (-20.3%)	31.4 (+4.4%)

Mandelbrot (2 AIs, 500 ms period)	Energy [kJ]	Execution time [s]
A100 default - 1410 MHz	10.3	30.0
CUDA ENRUNTIME dynamic tuning	8.8 (-14.6%)	30.8 (+2.5%)
Karolina default – 1290 MHz	8.3 (-16.7%)	31.7 (+5.5%)

ESPRESO FEM	Energy [kJ]	Execution time [s]
A100 default - 1410 MHz	13.6	53.7
CUDA ENRUNTIME dynamic tuning	12.6 (-7.4%)	57.0 (+6.1%)
Karolina default – 1290 MHz	11.4 (-16.1%)	58.9 (+9.6%)

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic through the e-INFRA CZ (ID:90254).
This work was supported by the POP3 project under grant agreement No 101143931. The project is supported by the European High-Performance Computing Joint Undertaking and its members (including top-up funding by the Ministry of Education, Youth and Sports of the Czech Republic (ID:MC2401)).

EUROPEAN UNION
European Structural and Investment Funds
Operational Programme Research, Development and Education

MINISTRY OF EDUCATION, YOUTH AND SPORTS