Evaluation of power counters and controls on general-purpose GPUs

[1]Ghazanfar Ali, [2]Sridutt Bhalachandra, [2]Nicholas Wright, [1]Alan Sill, [1]Yong Chen

[1]Texas Tech University, [2]Lawrence Berkeley National Laboratory

U.S. DEPARTMENT OF ENERGY
Office of Science

Abstract

General-purpose graphic processing units (GPUs) become increasingly important in high-performance computing (HPC) systems due to their massive computational performance. Although GPUs are attractive, modern GPU architectures consume a considerable amount of power, making it imperative to improve their energy-efficiency. This research focuses on understanding the power consumption and performance of various GPU architectures under different operating conditions and workloads. The investigation result provides insights for future predictive models and informed procurement designs and decisions.

Objective

The major factors complicating power management in the GPU-enabled HPC systems are the ever-changing GPU architectures and design space along with the complexity of HPC and emergent artificial intelligence (AI) workloads that run on them. This research study focuses on profiling different GPU architectures and workloads to understand the power consumption behavior and impact of power controls on performance.

GPU Interfaces

Nvidia GPUs support multiple interfaces to expose counters and perform control:

- Nvidia Management Library (NVML) is API for monitoring and managing various states of the Nvidia GPU
- Nvidia System Management Interface (nvidia-smi) is a command line utility on top of the NVML API.

```
nvidia-smi --querygpu=timestamp,power.draw,pstate,utilization.gpu,memory.total,
utilization.memory,memory.free,memory.used,clocks.sm,clocks.mem --loop-ms=1
```

```
timestamp, index, power.draw [W], pstate, temperature.gpu, utilization.gpu [%], utilization.memory [%], memory.total [MiB], memory.free
 [MiB], memory.used [MiB], clocks.current.sm [MHz], clocks.current.memory [MHz], clocks.current.graphics [MHz]
2020/06/26 13:09:10.682, 0, 252.79 W, P0, 28, 100 %, 56 %, 16280 MiB, 1261 MiB, 15019 MiB, 1316 MHz, 715 MHz, 1316 MHz
2020/06/26 13:09:10.687, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.697, 0, 252.79 W, P0, 28, 100 %, 56 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
2020/06/26 13:09:10.698, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.700, 0, 252.79 W, P0, 28, 100 %, 56 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
2020/06/26 13:09:10.701, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.703, 0, 244.57 W, P0, 28, 100 %, 56 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
2020/06/26 13:09:10.710, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.714, 0, 244.57 W, P0, 28, 100 %, 68 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
2020/06/26 13:09:10.715, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.717, 0, 244.57 W, P0, 28, 100 %, 68 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
2020/06/26 13:09:10.718, 1, 29.50 W, P0, 23, 0 %, 0 %, 16280 MiB, 16270 MiB, 10 MiB, 1189 MHz, 715 MHz, 1189 MHz
2020/06/26 13:09:10.720, 0, 244.57 W, P0, 28, 100 %, 68 %, 16280 MiB, 1261 MiB, 15019 MiB, 1303 MHz, 715 MHz, 1303 MHz
```

Figure 1: nvidia-smi output showing power, frequency among other metrics.

GPU Power Control and Metric Collection Framework

To use `nvidia-smi`, a transparent framework is developed:

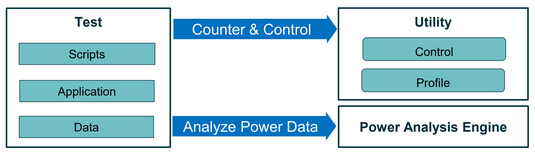

Figure 2: GPU power control and metric collection framework.

The framework consists of three modules:

- **Utility** module includes functions to launch application, applying power control and collection of GPU metrics
- **Power Analysis Engine** module provides an extensible interface to analyze the collected GPU metrics
- **Test** module invokes functions of the Utility and Power Analysis Engine modules

GPU Power and Performance Analysis

GPU Architectures:
1) Nvidia Pascal 100 (P100)
2) Nvidia Volta 100 (V100)

Benchmarks:
The power consumption and impact of power controls on three benchmarks and *idle* state are investigated and compared on both GPU architectures:
a) IDLE (power consumption while idle)
b) DGEMM (compute-bound)
c) STREAM (memory-bound)
d) FIRESTARTER (stress test)

Power Control Configurations:
All benchmarks are evaluated using the following power control configurations:

- Performance uses maximum available power up to the thermal design power (TDP) i.e. 250W
- Power cap uses the lowest supported power limit i.e. 125W
- Dynamic Voltage & Frequency Scaling (DVFS) uses the lowest supported frequency:
  - 544 MHz (P100)
  - 135 MHz (V100)

Power Stabilization Analysis

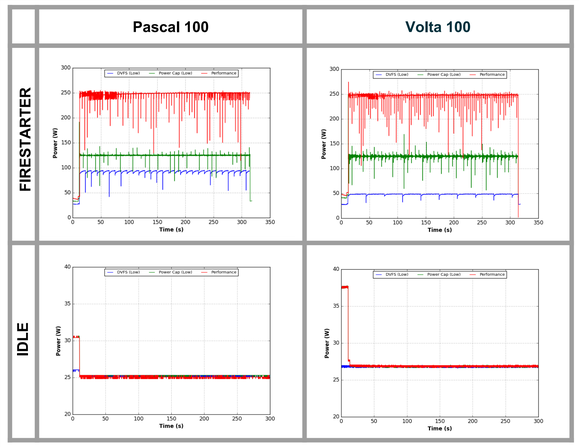

Figure 3: Power stabilization for Firestarter and IDLE

- Firestarter on V100 consumes ≈50W with DVFS (Low) in contrast to P100's ≈100W (differences in their lowest supported frequencies)

- The idle power usage on V100 is slightly higher (≈2W) than P100 on average
- *Overall, FIRESTARTER using DVFS (Low) shows the lowest power consumption and variation while taking the longest time for stabilization.*

Impact of Input Size

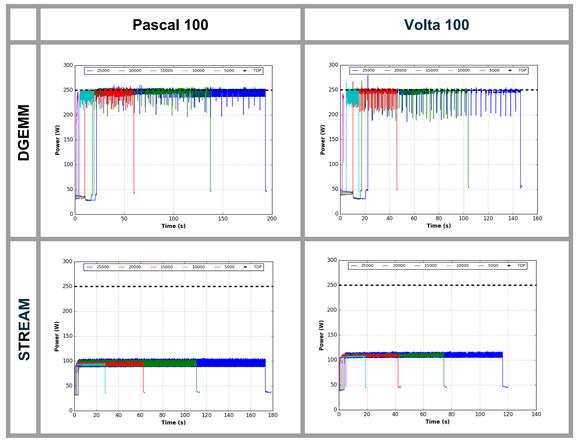

Figure 4: Power usage with varying matrix sizes - 5K, 10K, 15K, 20K, and 25K in performance configuration

- DGEMM power usage is near TDP on both architectures. DGEMM performance, however, is 25% faster on V100
- STREAM uses ≈100W on both architectures. STREAM throughput is 33% faster on V100
- *Overall, one can observe that the power profile is not affected by input size.*

Performance Comparison Analysis

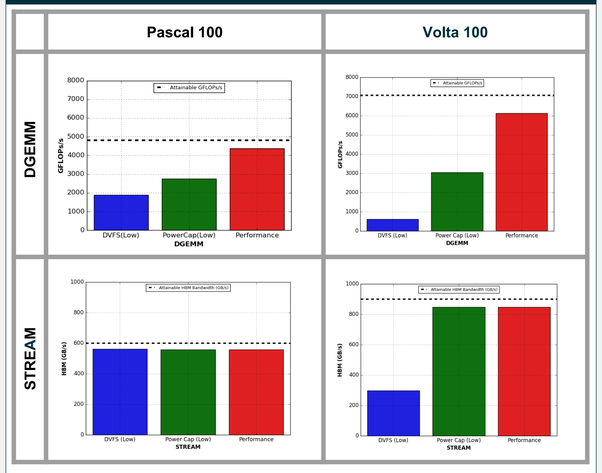

Figure 5: Performance comparison between P100 and V100

- DGEMM on V100 is 33% more performant than P100 using the same amount of power. With the Power Cap (Low), P100 marginally performs better. Using DVFS (Low), performance is the lowest on V100 due to comparatively its lower frequency.
- Interestingly, the throughput of STREAM is unaffected by the Power Cap (Low) as its peak power is lower than the minimum supported power limits on both GPUs.
- *Overall, with Performance configuration, V100 is faster than P100 mainly due to V100 higher SM count & frequency.*

Power Comparison Analysis

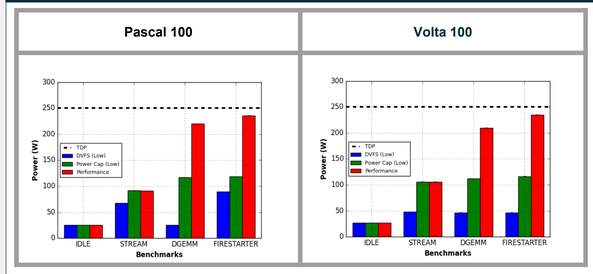

Figure 6: Power comparison between P100 and V100

- DVFS (Low) consumes the lowest power across all benchmarks on both architectures. Thus, DVFS can be used to extend the operational range of power limiting to support even lower power caps
  - On P100, it has more effect on the compute-bound DGEMM than the memory-bound STREAM benchmark
  - On V100, all benchmarks except IDLE consume similar power with DVFS (Low)
- *Power consumption in all modes is consistent across all benchmarks on both architectures (no noticeable variation)*

Conclusions and Future Work

- This research study has helped us to design and develop a preliminary framework to control and collect GPU metrics
- The evaluation of benchmarks on both GPU architectures has shown interesting insights related to power consumption behavior as well as impact of different power controls on the performance of benchmarks
- The power stabilization analysis, input size scaling, and performance-power comparisons across both GPU architectures show the GPU power characteristics for diverse computation patterns

In the future, we hope to explore further the behavior of each power control on power and performance to better understand some of the idiosyncrasies observed leading to a deterministic power consumption and control model for GPUs.

Acknowledgement

The National Energy Research Scientific Computing Center (NERSC) is a U.S. Department of Energy Office of Science User Facility operated under Contract No. DEAC02-05CH11231. Results presented in this paper were obtained using the Chameleon testbed supported by the National Science Foundation.