Evaluation of power controls and counters on general-purpose Graphics Processing Units (GPUs)

Ghazanfar Ali*, Sridutt Bhalachandra†, Nicholas Wright†, Alan Sill*, and Yong Chen*

* Texas Tech University, Lubbock, TX
Email: {ghazanfar.ali, christopher.turner, alan.sill, yong.chen}@ttu.edu
† Lawrence Berkeley National Laboratory, Berkeley, CA
Email:{sriduttb,njwright}@lbl.gov

***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/decisions.**

I. INTRODUCTION

A single advanced graphics processing unit (GPU) today consumes power (≈300W) equivalent to traditional high-performance computing (HPC) nodes [1]. A multi-GPU HPC node with current support of up to eight GPUs per node can consume several kWs of power per node. Consequently, this influx of these power-hungry GPUs has brought new power management challenges [2] in GPU-enabled HPC systems. The major factors complicating power management in the GPU-enabled HPC systems are the ever-changing GPU architectures and design space [3] along with the complexity of HPC and emergent artificial intelligence (AI) workloads that run on them. This research study investigates the power consumption and performance trade-off of diverse workload patterns on various GPU architectures. Based on these investigation findings, the long-term goal of this project is to develop power-performance prediction models and control strategies for GPUs in HPC systems.

II. GPU MEASUREMENT AND CONTROL

Nvidia GPUs support multiple interfaces to expose counters and enable system admins to adjust configurations to control GPU power in an efficient manner. The widely used GPU interfaces include *Nvidia Management Library (NVML)* and *Nvidia System Management Interface (nvidia-smi)*. In this work, *nvidia-smi* is used to control and collect GPU power and other metrics.

A. GPU Power Control and Metric Collection Framework

To generate a power/performance profile of a given workload efficiently using *nvidia-smi*, we have designed and developed a transparent framework (no re-compiling or linking) in this research, as shown in Fig. 1. This framework consists of three modules:

1) A *Utility* module includes functions to launch an application, apply power control and collect metrics.
2) A *Power Analysis Engine* module provides an extensible interface to analyze the collected GPU metrics data.
3) A *Test* module invokes functions of the *Utility* and *Power Analysis Engine* modules.

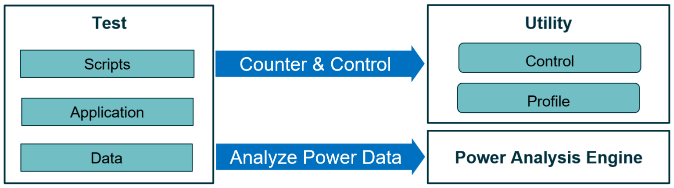

Fig. 1: GPU power control and metric collection framework.

III. GPU POWER AND PERFORMANCE ANALYSIS

The current evaluation tested two Nvidia GPU architectures – Pascal 100 (P100) and Volta 100 (V100). The GPU power consumption and impact of power controls is obtained using three benchmarks – DGEMM [4] (compute-bound), STREAM (memory-bound), and FIRESTARTER (stress test), as well as in the *idle* state. We evaluated these benchmarks using the following powe r control configurations:

- *Performance* mode uses power up to the thermal design power (TDP), i.e. 250W in our implementation.
- *Power cap (Low)* mode uses the lowest supported power limit (i.e. 125 W)
- *Dynamic Voltage and Frequency Scaling (DVFS) Low* mode uses the lowest supported frequency - 544 MHz (P100) and 135 MHz (V100)

A. Power Stabilization

Power stabilization [5] analysis shows the time taken by the GPU power to stabilize and the variations afterward. As shown in Fig. 2, FIRESTARTER on V100 consumes 50W roughly at DVFS (Low) in contrast to P100's about 100W consumption, mostly due to differences in their lowest supported frequencies. The power control has nominal impact on the idle power consumption. However, V100 power consumption 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.*

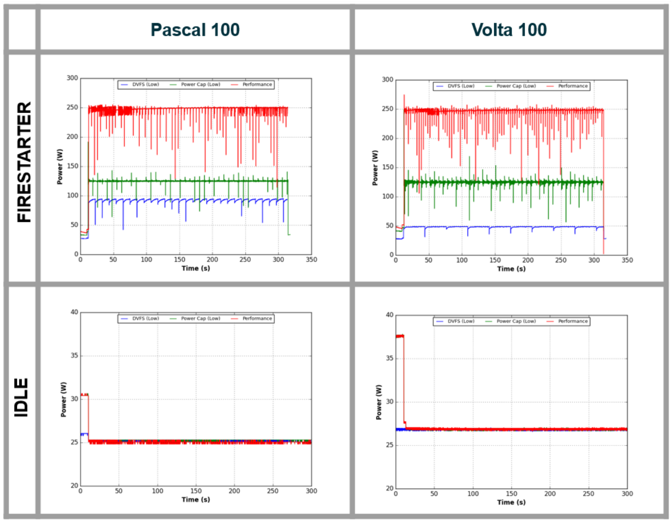

Fig. 2: Power stabilization for FIRESTARTER and IDLE

B. Impact of Input Size

Fig. 3 analyzes the impact of input size on the power profile. DGEMM power usage is close to the TDP on both architectures, but the execution time is roughly 25% less on V100. STREAM uses ≈100W on both architectures and is about 33% faster on V100. *Overall, the analysis shows that power profile is not affected by the input size.*

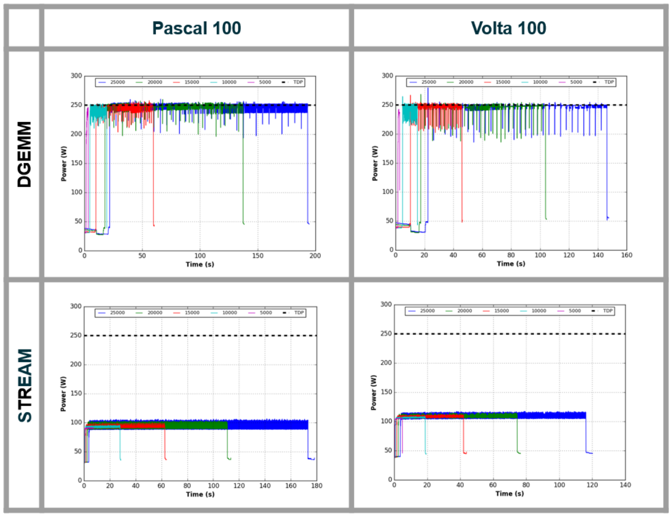

Fig. 3: Power usage with varying matrix sizes - 5K, 10K, 15K, 20K, and 25K in performance configuration.

C. Performance Comparison

Fig. 4 shows comparative analysis of the performance of DGEMM and STREAM on both architectures with all the power controls. In Performance mode, DGEMM on V100 is around 30% faster than P100; whereas in the Power Cap (Low) mode, V100 marginally performs better than P100. Using DVFS (Low), performance is lower on V100 due to its comparatively lower frequency. STREAM on V100 is faster than P100 with Performance and Power Cap (Low) configurations. These performance gains on V100 are mainly due to its higher streaming multiprocessors (SMs) count and frequency ranges [6]. The STREAM performance is not affected by the Power Cap (Low) as the peak power consumption of STREAM is lower than the minimum supported power limits on the both GPU architectures.

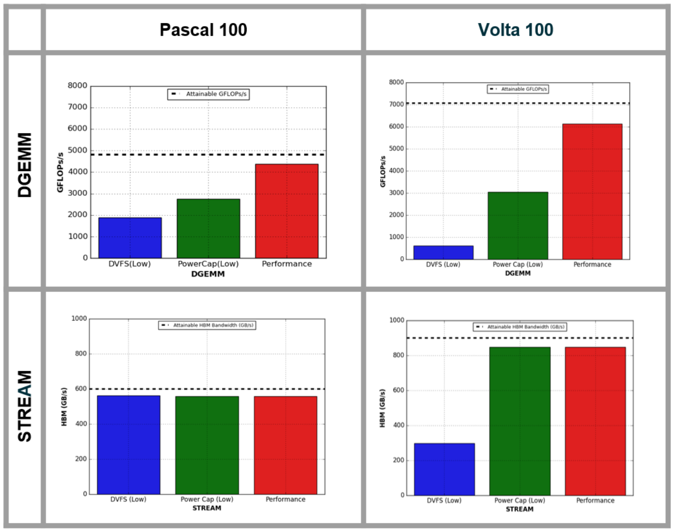

Fig. 4: Performance comparison on P100 and V100

D. Power Comparison

Fig. 5 shows the power consumption for all benchmarks on both architectures in different configurations. DVFS (Low) consumes the lowest power across all benchmarks on both architectures in all modes indicating DVFS may be used to effectively 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).

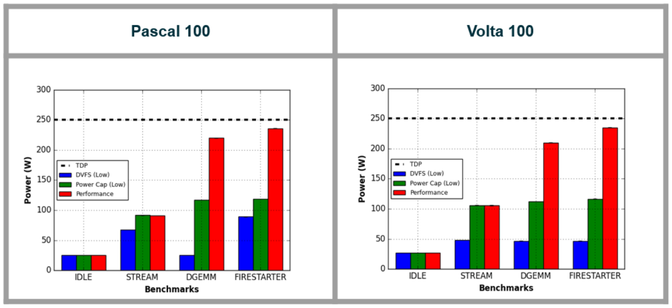

Fig. 5: Power comparison on P100 and V100

IV. Conclusions and Future Work

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 predictive model for GPU-enabled HPC systems.

ACKNOWLEDGMENT

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.

REFERENCES

[1] Y. Jiao, H. Lin, P. Balaji, and W. Feng, "Power and performance characterization of computational kernels on the gpu," in *2010 IEEE/ACM Int'l Conference on Green Computing and Communications Int'l Conference on Cyber, Physical and Social Computing*, 2010, pp. 221–228.

[2] O. Khodorkovsky, "Power management in multi-gpu systems," Sep. 16 2014, uS Patent 8,839,012.

[3] C. McClanahan, "History and evolution of gpu architecture," *A Survey Paper*, vol. 9, 2010.

[4] V. Volkov and J. W. Demmel, "Benchmarking gpus to tune dense linear algebra," in *SC'08: Proceedings of the 2008 ACM/IEEE conference on Supercomputing*. IEEE, 2008, pp. 1–11.

[5] M. Arora, S. Manne, I. Paul, N. Jayasena, and D. M. Tullsen, "Understanding idle behavior and power gating mechanisms in the context of modern benchmarks on cpu-gpu integrated systems," in *2015 IEEE 21st international symposium on high performance computer architecture (HPCA)*. IEEE, 2015, pp. 366–377.

[6] Z. Jia, M. Maggioni, B. Staiger, and D. P. Scarpazza, "Dissecting the nvidia volta gpu architecture via microbenchmarking," *arXiv preprint arXiv:1804.06826*, 2018.