Power and Performance of GPU-accelerated Systems: A Closer Look

Yuki Abe*, Hiroshi Sasaki*, Shinpei Kato†, Koji Inoue*, Masato Edahiro†, Martin Peres‡
*Kyushu University, Japan, †Nagoya University, Japan,
‡Laboratoire Bordelais de Recherche en Informatique, France
Email: {abe, sasaki}@soc.ait.kyushu-u.ac.jp, shinpei@is.nagoya-u.ac.jp,
inoue@ait.kyushu-u.ac.jp, eda@ertl.jp, martin.peres@labri.fr

I. Introduction

Graphics processing units (GPUs) are increasingly used as a means of powerful many-core compute devices. For example, even consumer series of NVIDIA GPUs integrate more than $1,500$ processing cores on a single chip and the peak double-precision performance exceeds 1 TFLOPS while sustaining thermal design power (TDP) in the same order of magnitude as traditional multicore CPUs. This rapid growth of GPUs is partly due to recent advances in the programming model, often referred to as general-purpose computing on GPUs (GPGPU). Current main applications of GPGPU can be found in supercomputing [8] but there are more and more emerging applications in many different fields (e.g. autonomous vehicles [5], software routers [3], and so on). This broad range of applications raises the need of understanding GPU-accelerated systems as a reliable computing infrastructure. One of the grand challenges of GPU-accelerated systems is the management of power and performance. Voltage and frequency scaling may address this problem but there is little evidence in the literature and there is no clear answer to *which GPUs benefit from voltage and frequency scaling to minimize energy*? Exploring the above issue is an essential piece of work towards the management of power and performance of GPU-accelerated systems.

This paper provides a closer look at the power and performance of GPU-accelerated systems. We conduct an evaluation using four different GPUs to quantify architectural impact of GPU voltage and frequency scaling on the execution time and power consumption. This evaluation characterizes power and performance across multiple generations of the GPU architecture.

II. Experimental setup

Our evaluation and modeling use four different NVIDIA graphics cards with an Intel Core i5 2400 processor. The operating system is Linux kernel v3.3.0 and we assume the Compute Unified Device Architecture (CUDA) as a programming model of GPUs. We consider that this is a pretty standard set of platforms for GPU-accelerated systems.

This paper focuses on the NVIDIA GPU architectures: *Tesla*, *Fermi* and *Kepler*. Among these different generations of GPU technology, we investigate four representative GPUs of the GeForce series: (i) Tesla-based GTX 285, (ii) Fermi- based GTX 460, (iii) GTX 480 and (iv) Kepler-based GTX 680. There are two Fermi-based GPUs selected because we aim to characterize a difference within the same GPU architecture as well as that among different architectures.

TABLE I
Frequency levels of the NVIDIA GPUs.

GPU		GTX 285	GTX 460	GTX 480	GTX 680
Core (MHz)	Low	600	100	100	648
	Medium	800	810	810	1080
	High	1296	1350	1400	1411
Memory (MHz)	Low	100	135	135	324
	Medium	300	324	324	810
	High	1284	1800	1848	3004

We use NVIDIA's proprietary software including the device driver, runtime library and compiler. Since this software package does not provide a system interface to scale power and performance (voltage and frequency) levels of GPUs, we modify the BIOS image of the target GPU, which is embedded in the device driver's binary code, forcing the GPU to be booted at the specified power and performance levels. This method allows us to choose a pre-defined configurable set of the GPU core and memory clocks listed in TABLE I where voltage is implicitly adjusted with frequency changes. Interested readers for this open method are encouraged to visit the software repository of Gdev [4] and find documentations about voltage and frequency scaling of NVIDIA GPUs. Note that this method requires the device driver to reload and the configuration is static while it is running. An open-source driver provided by Linux [6] provides a "`/proc`" interface to scale power and performance levels of GPUs at runtime. However, it does not currently support the memory clock and there is some performance issue [1]. We restrict our attention to the proprietary software in this paper but future work will consider the use of this open-source driver for dynamic power and performance management.

The power consumption of the entire system is measured using the Yokogawa Electric Corporation's WT1600 digital power meter. This instrument obtains voltage and electric current every $50ms$ from the power outlet of the machine. Power consumption is calculated by multiplying the voltage and current, whereas energy consumption is derived by accumulation of power consumption. The measurement is focused on power and energy of the entire system but not on those

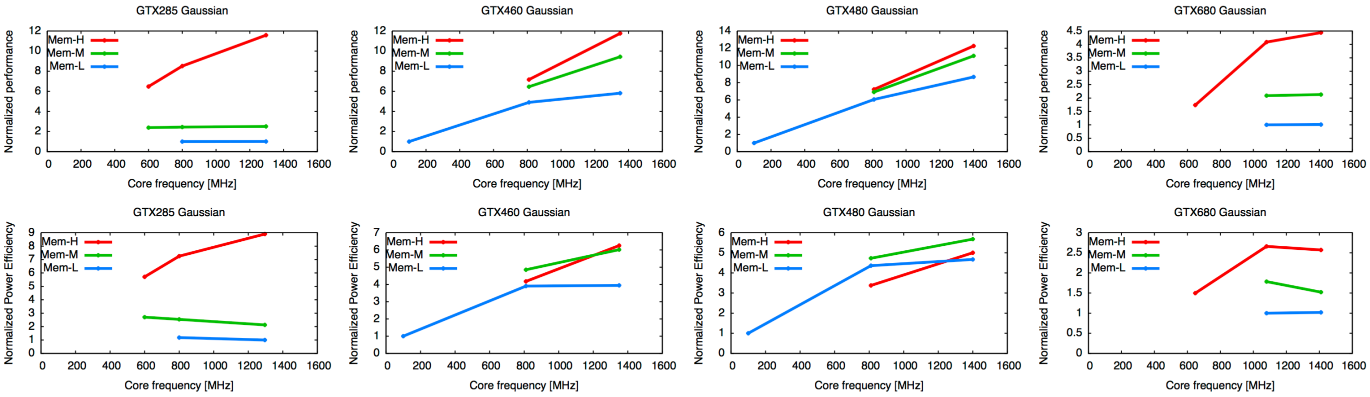

Fig. 1. Performance and power efficiency of `Gaussian`.

of individual CPUs and GPUs because our focus is on the system-wide study. There are also equipment constraints that prevent us from measuring power consumption of the GPU alone as its power is directly supplied from the power supply and also from the PCI bus.

The benchmarks used in our experiments include Rodinia [2] and Parboil [7] which are popular benchmark suites for GPGPU studies. We execute each benchmark program with the maximum feasible input data size. We also use the CUDA SDK code samples and basic matrix operation programs with large input data size. For such a program that has an execution time less than $500ms$, we modify the code to repeat the computing kernel of the program until the execution time reaches $500ms$ in order to obtain at least 10 sample points. All the test programs are compiled using the NVIDIA CUDA Compiler (NVCC) v4.2.

III. Characterization of Power and Performance

We characterize the power and performance of GPU-accelerated systems and figure out the effectiveness of GPU voltage and frequency scaling. Due to space constraints, we present only the selected result of benchmarking.

Fig. 1 shows the performance and the power efficiency, *i.e.*, reciprocal of the energy consumption of `Gaussian`. The x-axis of each figure represents the processing core frequency and different lines in the figure correspond to different memory frequencies. Mem-H, -M, and -L denote the high, medium and low frequencies of the memory corresponding to TABLE I. For example, “Mem-L” of GTX 285 sets the memory frequencies to 100MHz. The processing core and memory clocks of the GPU are scaled individually according to the configurable frequency combinations pre-defined by the NVIDIA specifications Note that frequency pairs depend on each GPU.

The best power efficiency for GTX 285 and GTX 460 is achieved by (Core-H, Mem-H), while that of GTX 480 is provided with (Core-H, Mem-M). In addition, that of GTX 680 is produced by (Core-M, Mem-H). Even if we compare the same-generation GPUs, *i.e.*, GTX 460 and GTX 480, the power efficiency characteristics differ such that the best configuration is not identical between them, which implies that it is not straightforward to predict an optimal core and memory frequency pair. This characteristic explains that processor and memory voltage and frequency scaling is a useful approach to minimizing energy of GPU-accelerated systems.

IV. Conclusion

In this paper, we have presented a characterization of power and performance for GPU-accelerated systems. We selected four different NVIDIA GPUs from three generations of the GPU architecture in order to demonstrate generality of our contribution. One of our findings is that the power efficiency characteristics differ such that the best configuration is not identical between the GPUs. This evidence encourages future work on the management of power and performance for GPU-accelerated systems to benefit from dynamic voltage and frequency scaling. In future work, we plan to develop a dynamic voltage and frequency scaling algorithm for GPU-accelerated systems.

Acknowledgement

This research was supported in part by New Energy and Industrial Technology Development Organization (NEDO) and Japan Science and Technology Agency (JST), CREST.

References

[1] Y. Abe *et al.*, “Power and performance analysis of GPU-accelerated systems,” in *Proc. of the UESNIX Workshop on Power-Aware Computing and Systems*, 2012.

[2] S. Che *et al.*, “A characterization of the Rodinia benchmark suite with comparison to contemporary CMP workloads,” in *Proc. of the IEEE International Symposium on Workload Characterization*, 2010.

[3] S. Hand *et al.*, “PacketShader: a GPU-accelerated software router,” in *Proc. of ACM SIGCOMM*, 2010.

[4] S. Kato *et al.*, “Gdev: first-class GPU resource management in the operating system,” in *Proc. of the USENIX Annual Technical Conference*, 2012.

[5] M. McNaughton *et al.*, “Motion planning for autonomous driving with a conformal spatiotemporal lattice,” in *Proc. of the IEE International Conference on Robotics and Automation*, 2011.

[6] Nouveau, “Nouveau: accelerated open-source driver for NVIDIA cards,” 2013, http://nouveau.freedesktop.org.

[7] J. Stratton *et al.*, “Parboil: a revised benchmark suite for scientific and commercial throughput computing,” IMPACT-12-01, University of Illinois at Urbana-Champaign, Tech. Rep., 2012.

[8] TOP500 Supercomputing Site, 2012, http://www.top500.org/.