# Empirical Characterization of Power Efficiency for Large Scale Data Processing

Yongbin LEE *, Sungchan KIM *

* Division of Computer Science and Engineering, Chonbuk National University, Korea

**forever1363@chonbuk.ac.kr, sungchan.kim@chonbuk.ac.kr**

*Abstract*— **It becomes popular to equip CPU and GPU on a single computer system because of its performance and energy benefits, constituting a heterogeneous system for processing big data workloads. However, the optimal exploitation of such a heterogeneous system requires us to know the power consumption characteristics of the applications for difference processing units. To this end, this paper aims at characterizing the power efficiency of CPUs and GPUs for big data processing through empirical measurements. We take three recent computing units, high-end CPU, and GPU, and mobile embedded GPU as target platforms. We first show the performance and power consumption measurements on each computing platform using the Rodinia benchmarks as representative big data workloads. Then, we discuss how performance-per-watt of each computing platform is associated with different characteristics of the workloads.**

*Keywords*— **Performance-per-watt, big data workload, measurement, Rodinia benchmark**

## I. INTRODUCTION

Data centers are one of the largest and fastest growing consumers of electricity in the United States. In 2013, U.S. data centers consumed an estimated 91 billion kilowatt-hours of electricity, which is large enough to power all the households in New York City twice over, and are on-track to reach 140 billion kilowatt-hours by 2020 [1]. It tends to increase rapidly related to process ever-growing large scale data-intensive applications, namely big data. Such workloads often possess a high degree of data-level parallelism. Recent improvement in computing capability of modern CPUs is mainly due to the employment of multiple computing units in a single chip, which is called a multi-core processor. A typical multi-core processor exploits both SIMD (Single-Instruction-Multiple-Data) and MIMD (Multiple-Instruction-Multiple-Data) types of parallelisms. A core is designed to have a special set of instructions to support SIMD operation. The Intel SSE or AVX instructions are such examples [2][3]. Then, more than a cores in a single chip enable MIMD operations, for example, at thread-level. While such an architectural innovation of processors scales well in performance, its energy efficiency is limited to cope with big data workloads that are growing at an astounding rate. As a result, a power-aware metric, so called performance-per-watt, becomes the first-class citizen when designing a computing platform for big data processing.

Regarding such a challenge, GPUs (Graphics Processing Units) are attracting considerable attention. Highly parallel small computing cores compared to CPU is a key enabler for yielding high performance-per-watt for big data workloads. Besides its conventional form, i.e., discrete card for HPC (High Performance Computing), power-efficient mobile GPUs are available on the market, being actively used in many domains, such as mobile systems, robots, and automotive.

Even though there is a general agreement in the advantage of GPU over CPU in performance-per-watt, to our best knowledge, no quantitative comparisons of the recent CPUs and (mobile and discrete) GPUs with suitable workloads exist. To this end, in this paper, we perform the comparison of the recent CPU and GPU quantitatively in terms of performance-per-watt. Particularly, we perform extensive empirical measurements of performance and energy consumption for three computing platforms, high-end server CPU and discrete GPU, and mobile GPU by using the Rodinia benchmark suites [4] as big data workloads. Then, we provide several key observations on the characteristics of performance-per-watt of each computing platform for the different big data workloads.

The rest of this paper is as follows. Section II summaries the related work. Section III explains on the experimental methodology including the computing platforms for comparisons and the benchmarks to run. Section IV provides the experimental results and related observations. Section V concludes this paper.

## II. RELATED WORK

The several benchmarks have been introduced for multi-core CPU and GPU [4][5][6]. Among them, the Rodinia benchmark suite provides a collection of parallel programs for the study of heterogeneous systems [4]. They analysed characterization of diversity of the benchmarks, and using CUDA and OpenMP, confirmed the speedup following parallelization toward each application. They also showed that the advantage of accelerator-based computing is its potential to achieve better power efficiency than CPU-based computing. Even though the previous works have performed the comparisons of GPUs and multi-core CPUs [4][7], the GPUs and CPUs used in the literatures are out-dated. The recent commodities have improved significantly especially in terms of power efficiency,

and the analysis with the recent platforms has not performed yet.

## III. Background and Evaluation Methodology

### A. GPGPU

A GPGPU (General-Purpose Graphics Processing Units), or GPU in short, utilizes a GPU to perform general purpose computation in applications that are traditionally done by a CPU. In this study, we used a Kepler architecture-based GPU [8]. A plurality of parallelism is present in the GPU, which in turn is also a multi-core processor. The GPU has 13 to 15 cores, each of which is called an SMX (Next Generation Streaming Multiprocessor). Each of the SMX units features 192 single-precision CUDA cores and has fully pipelined floating-point and integer arithmetic logic units. The SMX schedules threads in groups of 32 parallel threads, called warps. Each SMX has four warp schedulers and eight instruction dispatch units, allowing four warps to be issued and executed concurrently.

GPUs are programmed using CUDA [9], which is an extension of C with slightly syntactical additions and run-time API and library.

### B. Rodinia Benchmark Suite

We use the Rodinia benchmark suite [4], which is a collection of benchmarks for parallel processing on heterogeneous computing platforms. The latest version of Rodinia contains 31 parallel applications from various domains such as medical imaging, bioinformatics, data mining, and scientific computing. Parallelizable part of each application is implemented using both multi-core CPU and GPU. The multi-core implementation employs OpenMP (Open Multi-Processing) [10], which is an API that supports multi-platform shared memory multiprocessing programming in C, C++, and Fortran used on multicore CPUs. On the other hand, the GPU implementation is available in CUDA and OpenCL (Open Computing Language) [11], which is a programming language for heterogeneous systems including GPUs.

### C. Experiment Setups

Table 1 shows the architectural parameters for the three computing platforms under consideration in this study. We use a quad-core Intel i7 processor with the simultaneous hardware multithreading enabled, thus eight cores are seen by users. As a discrete GPU, we use NVIDIA GTX Titan Black that is one of high-end GPUs available on the market. For a mobile GPU, we chose NVIDIA Tegra K1 [12], which is a System-on-chip that has a quad-core ARM Cortex A15 CPU and a Kepler GPU with 192 CUDA cores. In the experiment, we used the Jetson TK1 development board [13] for the Tegra K1 that runs CUDA programs on a complete Linux distribution.

Table 2 provides the descriptions on the application domain, input data size, and parallelism exhibited on GPU of the benchmarks chosen from the Rodinia suites for the evaluation. Note that some benchmarks are excluded from the study as the input data of the benchmarks is too small to reasonably measure the performance of the high-end CPU and GPU.

The power consumptions of the benchmarks were obtained from instrumental measurement using a digital multimeter. Because it is difficult to specifically measure the power consumption of processing units and memory from the target systems, we use a simple scheme for measuring power consumption. The measurement was performed at the power supply, meaning that we obtain the power consumption of the entire system. To consider the power consumption due to the execution of the benchmarks only, we first measure the power consumption of the system when it is idling, and then subtract the idle power from the power consumption measured while the system is busy for running the benchmarks.

TABLE 1. PARAMETERS FOR THREE COMPUTING PLATFORMS.

| Platforms | Parameter | Value |
|---|---|---|
| CPU (Intel i7) | # cores | 8 |
| | Core clock | 3.20GHz |
| | Peak throughput (single-precision) | 102.4GFLOPS |
| | Cache size (L1/L2/L3) | (32KB/256KB/8,192KB) |
| | Main memory size | 12,295MB |
| | Main memory bandwidth | 25.6GB/s |
| | PCI-e version | 2.0 |
| Discrete GPU (NVIDIA GTX Titan Black) | # SMXs | 15 |
| | # CUDA cores | 2,688 |
| | Core clock | 889MHz |
| | Peak throughput (single-precision) | 4,494GFLOPS |
| | Main memory size | 6,291MB |
| | Main memory bandwidth | 336GB/s |
| Mobile GPU (NVIDIA Tegra K1) | # SMXs | 1 |
| | # CUDA cores | 192 |
| | Core clock | 852MHz |
| | Peak throughput (single-precision) | 365GFLOPS |
| | Main memory size | 1,048MB |
| | Main memory bandwidth | 17GB/s |

TABLE 2. SELECTED BENCHMARKS FROM THE RODINIA SUITES.

| Benchmark (Abbreviation) | Dwarves | Domains | Input data size (KB) | # threads per GPU run |
|---|---|---|---|---|
| B+ Tree (BT) | Graph Traversal | Search | 6889 | 1,536,000 |
| Breadth-First Search (BFS) | Graph Traversal | Graph Algorithms | 262 | 1,000,448 |
| K-means (KM) | Dense Linear Algebra | Data Mining | 61223 | 65,536 |
| Needleman-Wunsch (NW) | Dynamic Programming | Bioinformatics | 16777 | 2,048 |
| Particle Filter (PF) | Structured Grid | Medical Imaging | 104858 | 1,024 |
| Back Propagation (BP) | Unstructured Grid | Pattern Recognition | 2000 | 1,048,576 |
| Heart Wall (HW) | Structured Grid | Medical Imaging | 50762 | 13,056 |
| Streamcluster (SC) | Dense Linear Algebra | Data Mining | 524 | 65,536 |
| LU Decomposition (LUD) | Dense Linear Algebra | Linear Algebra | 262 | 976 |
| LavaMD (MD) | N-Body | Molecular Dynamics | 8 | 128,000 |
| Myocyte (MC) | Structured Grid | Biological Simulation | 3 | 64 |
| SRAD_v1 (SR_v1) | Structured Grid | Image Processing | 91966 | 230,400 |
| SRAD_v2 (SR_v2) | | | 134218 | 4,194,304 |

## IV. Evaluation Results

In this section, we provide the performance and power consumption of the benchmarks on the computing platforms under consideration and several observations to draw characteristics on performance-per-watt for the different platforms and workloads.

Figure 1(a) and (b) show the power consumptions and data processing rate of the benchmarks on four different ways, CPU with a single thread and 8 threads, discrete GPU, and mobile GPU, respectively. First of all, the mobile GPU consumes less

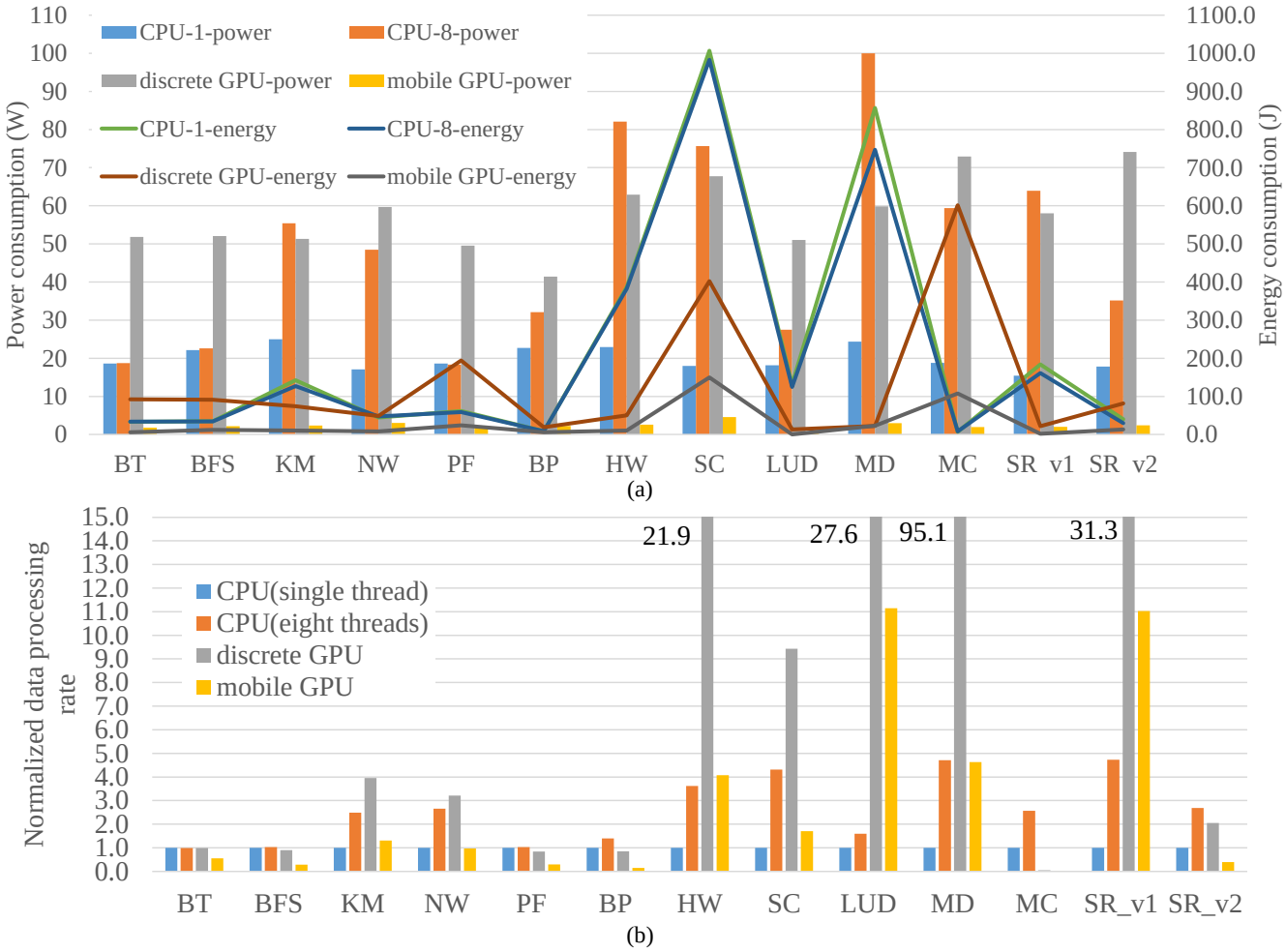


Figure 1. Comparison of the three platforms for running the benchmarks in terms of (a) power and (b) data processing rate normalized to the case of the CPU with a single thread.

energy than the CPUs and discrete GPU for all the benchmarks except for MC, which will be explained later. On the other hand, the discrete GPU consumes energy the most over the CPU and the mobile GPU. Note that the power consumption of the mobile GPU is less than 5W throughout all the benchmarks. Overall, the mobile GPU is up to 30x and 7x power-efficient compared to the discrete GPU and the CPUs.

Data processing rates normalized to that of the CPU with a single thread are shown in Figure 1(b). Interestingly, not all benchmarks are best performed with the discrete GPU. PF, BP, SR_v2, and MC are such examples. On the other hand, the performance of the mobile GPU is comparable with that of the 8-threaded CPU in several benchmarks such as BT, HW, LUD, MD, and SR_v1.

Performance-per-watt, i.e., data processing rate per watt, of each benchmark is depicted in Figure 2. The mobile GPU outperforms others except for the MC and MD benchmarks. There is, however, no consistent tendency on the power efficiencies of the CPU and the discrete GPU, which depends on the benchmarks. The BT benchmark is a typcial case where the mobile GPU is the most energy-efficient. Performance-per-watt of the discrete GPU is only 75 KB/s per W while it is 1206 KB/s per W with the mobile GPU consuming 5.34W on average.

On other hand, the MC benchmark is best performed with the CPU unlike other benchmarks. This is due to limited parallelism in the benchmark. In particular, up to 64 threads are created for running the benchmarks, providing enough parallelism with the CPU. This degree of parallelism, however, remains the same when applied to the GPUs. Given that even the mobile GPU is capable of processing 192 simultaneous threads at least, such a limitation in parallelism severely restricts the exploitation of the compute capability of the GPUs. The problem becomes worse with the discrete GPU that is designed to execute 2K threads at the same time.

Another exceptional observation is found in the MD benchmark, where the GPUs are significantly energy-efficient compared to the CPU. This is because the benchmark has abundant parallelism in opposite to the MC benchmark. Furthermore, the input data of the benchmark is very small so that the overhead for transferring input data to the GPU devices is almost negligible. As a result, the most of execution time is consumed by the computation using the GPUs.

## V. Conclusions

The study provides the performance and power characterization of recent multi-core CPUs and GPUs using the Rodinia benchmarks as representative workloads of parallel programs. Several key observations were made through the extensive measurements. Overall, using GPUs promises better performance-per-watt compared to the case of CPUs as more

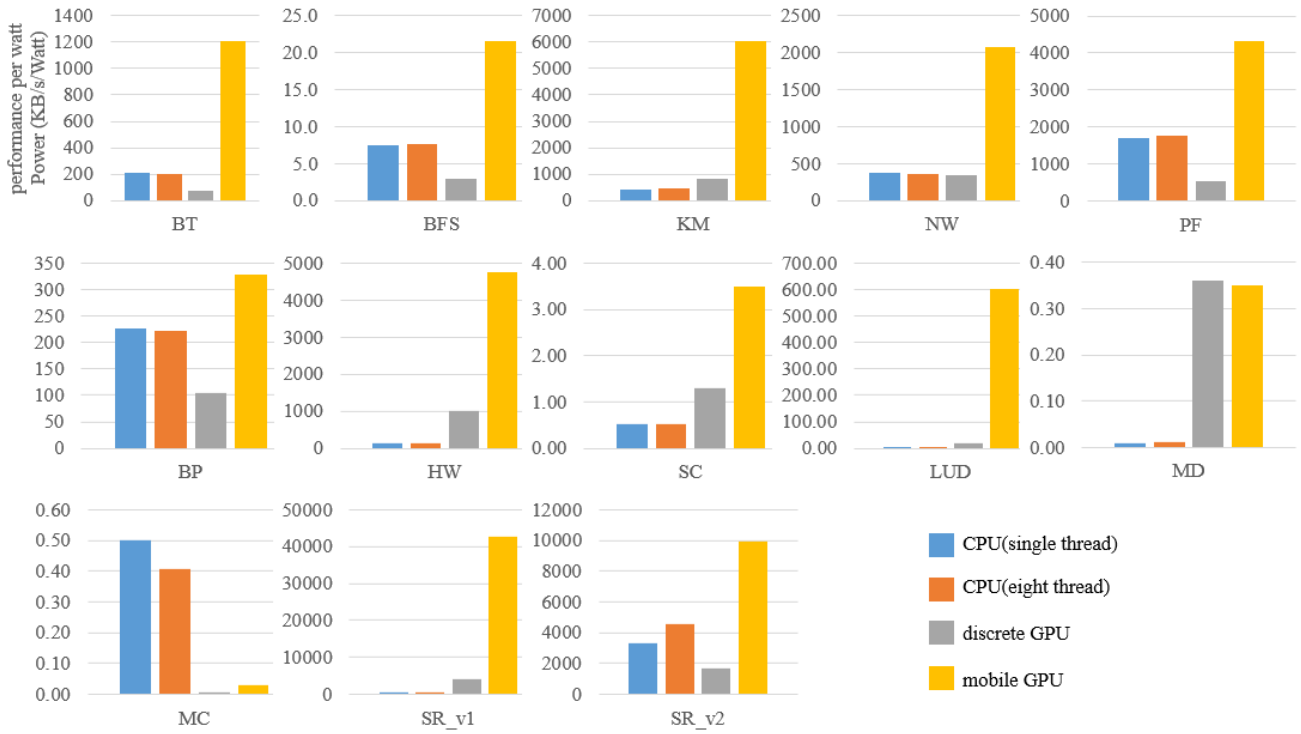


Figure 2. Performance per watt of the benchmarks on the three computing platforms.

parallelism exists in an application. The experiments show that an application with limited parallelism is advantageous of using CPU instead of GPU. The mobile GPU shows the significant improvement in performance-per-watt over the CPU and discrete GPU due to its outstanding power efficiency. Hence, beyond the typical applications such as battery-powered hand-held devices, it may be considered for large scale data processing in place of conventional multi-core CPUs and discrete GPUs in the near future.

Future work will focus on several in-depth comparisons to find different characteristics of the parallel benchmarks from those observed in the previous studies.

## Acknowledgements

This work was supported by ICT R&D program of MSIP/IITP (B0101-15-0661, the research and development of the self-adaptive software framework for various IoT devices).

## REFERENCES

[1] Natural Resources Defense Council. 2015. “America's Data Centers Consuming and Wasting Growing Amounts of Energy”, Feb. 06 (http://www.nrdc.org/energy/data-center-efficiency-assessment.asp)

[2] Intel streaming SIMD extension http://www.intel.com/support/processors/sb/CS-030123.htm?wapkw=sse (modified Feb., 09, 2015)

[3] Intel Advanced Vector Extensions, https://software.intel.com/en-us/isa-extensions/intel-avx

[4] Che, Shuai, et al. "Rodinia: A benchmark suite for heterogeneous computing." Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on. IEEE, 2009.

[5] C. Bienia, S. Kumar, J. P. Singh, and Kai Li, “The PARSEC benchmark suite: characterization and architectural implications, “ in Proc. International conf. Parallel Architectures and Compilation Techniques, pp. 72-81, 2008.

[6] J. A. Stratton, C. Rodrigues, I.-J. Sung, N. Obeid, L.-W. Chang, N. Anssari, G. D. Liu, W. W. Hwu, “Parboil: A Revised Benchmark Suite for Scientific and Commercial Throughput Computing, IMPACT Technical Report, IMPACT-12-01, University of Illinois, at Urbana-Champaign, 2012.

[7] Yuki Abe, Hiroshi Sasaki, Martin Peres, Koji Inoue, Kazuaki Murakami, Shinpei Kato, “Power and Performance Analysis of GPU-Accelerated Systems,” In Proc. USENIX Workshop on Power-Aware Computing and Systems, 2012.

[8] NVIDIA CUDA(Kepler Architecture), http://www.nvidia.com/object/-nvidia-kepler.html

[9] J. Nickolls, I. Buck, M. Garland, and K. Skadron. Scalable parallel programming with CUDA. ACM Queue, 6(2):40–53, 2008.

[10] OpenMP API Specification for parallel programming, http://www.openmp.org

[11] John E. Stone, David Gohara, and Guochin Shi, “OpenCL: a parallel programming standard for heterogeneous computing systems,” Computing in Science & Engineering, vol 12, no.3, 2010.

[12] NVIDIA Tegra K1, http://www.nvidia.com/object/tegra-k1-processor.html

[13] NVIDIA Jetson TK1, http://www.nvidia.com/object/jetson-tk1-embedded-dev-kit.html

**Yongbin Lee** received the B.S. degree in 2014, and now has been studying for a master's degree since March, 2014 in computer science and engineering from Chonbuk National University, Korea. His research interests include embedded system, big data computing, and GPU.

**Sungchan Kim** received the B.S. degree in material science and engineering, the M.S. degree in computer engineering, and the Ph.D. degree in electrical engineering and computer science from Seoul National University, Seoul Korea, in 1998, 2000, and 2005, respectively. He is currently Associated professor at Chonbuk National University, Korea. His research interests include various aspects and emerging technologies for parallel computing such as reliable multiprocessor system, non-volatile memory-based storage, and Big Data processing.