Roofline-aware DVFS for GPUs

Cedric Nugteren Gert-Jan van den Braak Henk Corporaal
Eindhoven University of Technology, The Netherlands
{c.nugteren, g.j.w.v.d.braak, h.corporaal}@tue.nl

ABSTRACT

Graphics processing units (GPUs) are becoming increasingly popular for *compute* workloads, mainly because of their large number of processing elements and high-bandwidth to off-chip memory. The *roofline model* captures the ratio between the two (the *compute-memory* ratio), an important architectural parameter. This work proposes to change the compute-memory ratio dynamically, scaling the voltage and frequency (DVFS) of 1) memory for compute-intensive workloads and 2) processing elements for memory-intensive workloads. The result is an adaptive *roofline-aware* GPU that increases energy efficiency (up to 58%) while maintaining performance.

Categories and Subject Descriptors

C.1.4 [**Processor Architectures**]: Parallel Architectures; C.4 [**Performance of Systems**]: Modeling Techniques

General Terms

Performance

Keywords

Parallel Computing, GPU, DVFS, The Roofline Model

1. INTRODUCTION

In the past decade, graphics processing units (GPUs) have emerged as a popular platform for non-graphics computations: programmers now use these massively parallel accelerators for computational problems in domains such as image processing and molecular science. In particular, GPUs are well-suited for throughput-oriented applications, because of their large number of processing elements and their high off-chip memory bandwidth [3]. While the first enables a high instruction throughput, typically measured in floating point operations per second (FLOPS), the second enables a high data throughput, measured in bytes per second (B/s).

The ratio between the instruction and data throughput is the *compute-memory ratio*, an important design parameter for GPUs. This is visualised in the *roofline model* [9], an abstract analytical model based on a kernel's *operational intensity*. This metric (measured in operations per byte), determines which of the two limits apply: kernels can either be *compute-bound* (limited by the processing elements) or *memory-bound* (limited by memory bandwidth).

This work explores adapting the compute-memory ratio of GPUs for a specific workload. Dynamic voltage and frequency scaling (DVFS) is applied to either the GPU core or its memory, saving power while maintaining performance. This idea was proposed as part of earlier work [8], but is now also validated experimentally. Although DVFS for GPUs is not new [4, 5, 7], this is the first work to combine it with the roofline model and the operational intensity metric.

ADAPT '14, Jan 22 2014, Vienna, Austria
ACM 978-1-4503-2514-1/14/01.
http://dx.doi.org/10.1145/2553062.2553067

2. ROOFLINE-AWARE DVFS

The concept of this work is illustrated by the roofline model. Figure 3 gives an example roofline model of a GeForce GTX470 GPU (Fermi architecture) with a single-precision peak instruction throughput of 538 GFLOPS (counting `fmad` as one operation) and an off-chip memory throughput of 144GB/s. On the left hand side, a memory-intensive GPU kernel is shown: it accesses an average of one off-chip byte per instruction. According to the roofline model abstractions, the instruction throughput can now be halved (or more) without loosing performance. The right hand side shows the dual: a compute-intensive kernel that allows memory throughput to be reduced while maintaining performance.

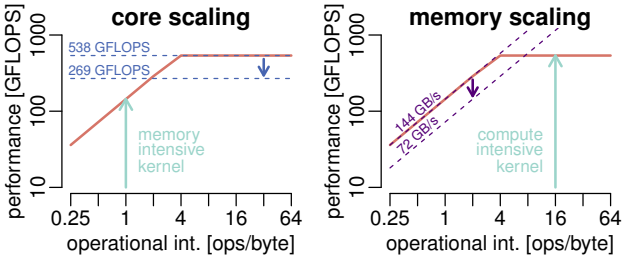

Figure 3: A roofline model for a GTX470 GPU, illustrating the scaling of core performance for memory-intensive workloads (left) and memory scaling for compute-intensive workloads (right).

Two techniques can be applied to obtain a scaling of core and memory performance on a per-kernel basis: 1) dynamic frequency scaling (DFS) of the core or memory clock, or 2) dynamic disabling of processing elements or memory banks.

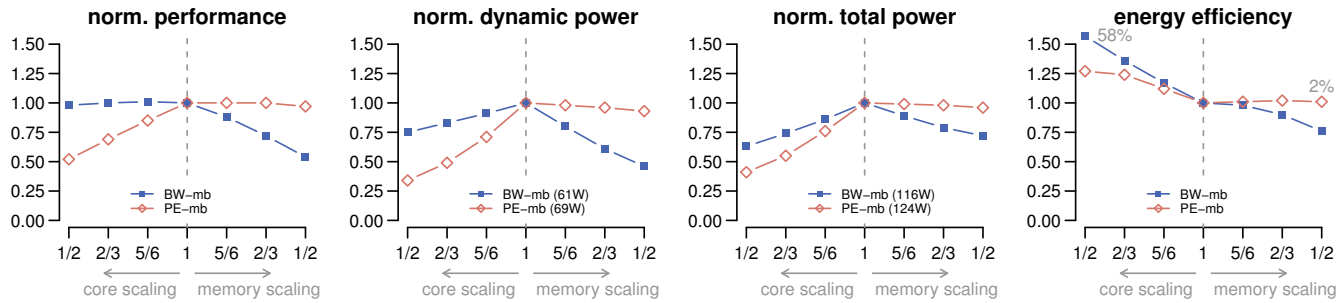

Figure 1: Normalised performance (left), normalised dynamic and total power (middle), and energy efficiency (right) for a memory-intensive (`BW-mb`) and a compute-intensive micro-benchmark (`PE-mb`).

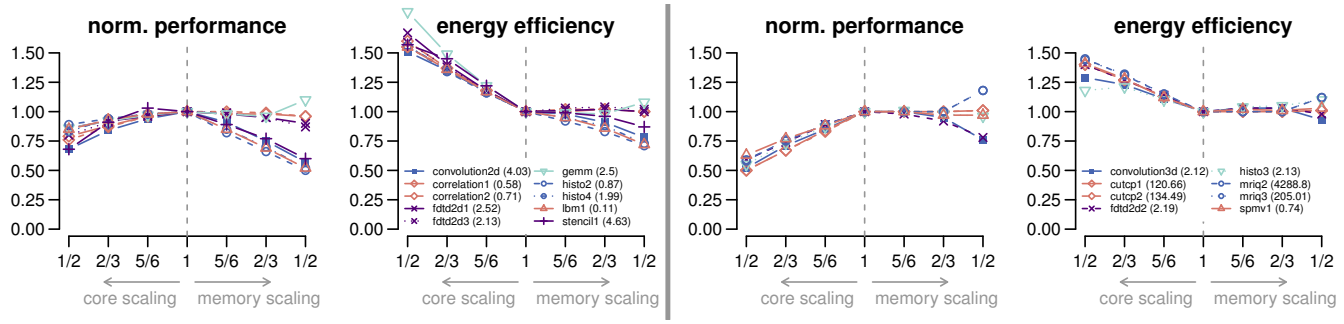

Figure 2: Sorted results for benchmarks with a low core scaling sensitivity (left) and a high core scaling sensitivity (right). The legends show the simulated operational intensities in operations per off-chip byte.

The latter technique reduces static (leakage) and dynamic power linearly, but complicates data storage due to powered down memory banks. DFS only reduces dynamic power linearly, but can be extended with voltage scaling (DVFS) to obtain better scaling of dynamic power ($P = \alpha \cdot f \cdot C \cdot V^2$) and additional scaling of static power. We therefore propose to apply DVFS based on a kernel's operational intensity, thus obtaining a roofline-aware GPU.

Earlier work [2, 6] focused on *phases*: portions of the execution in which specific properties (e.g. the operational intensity) remain constant. Although such phases might exist, it is typically valid to consider the average operational intensity: GPUs execute many threads independently on multiple cores, averaging work from different phases.

3. EXPERIMENTAL RESULTS

To quantify the potential of roofline-aware DVFS, experiments are performed on 1) two micro-benchmarks, and 2) the Parboil and PolyBench/GPU benchmark suites. We use version 3.2.1 of GPGPU-Sim [1] and the GPUWattch power model [6]. The simulator is configured to match a GTX470 with (rounded) nominal frequencies of 1200MHz and 900MHz for the core and memory (quad-pumped) respectively. The frequencies are halved in steps: 1200–1000–800–600MHz (core) and 900–750–600–450MHz (memory) for a total of 1 nominal and 6 scaled operating points. The amount of voltage scaling applied in three steps for both core and memory is 0.9^2, 0.8^2 and 0.7^2 [6]. The estimated leakage power is 55W based on GTX480 data ($\frac{14\text{cores}}{15\text{cores}} \cdot 59W$ [6]) and scales with core voltage but not with frequency. The configuration files, the benchmarks, and the full results are available at `http://github.com/cnugteren/rooflineDVFS`.

Figure 1 shows the micro-benchmark results, constructed to be either memory-intensive (`BW-mb`) or compute-intensive (`PE-mb`). The results are as expected: core scaling halves the performance (and reduces the power) for the compute-intensive benchmark but maintains nominal performance for the memory-intensive benchmark. Vice versa for memory scaling. At nominal performance, energy savings are 58% for `BW-mb` (core scaling to $\frac{1}{2}$) and 2% for `PE-mb` (memory scaling to $\frac{1}{2}$). The reason the compute-intensive benchmark does not benefit is the relatively low memory power compared to the GPU core, in particular when it is idle most of the time.

Figure 2 shows results for the benchmarks grouped by their sensitivity to core scaling. The two left graphs show results with a low operational intensity and significant energy gains for core scaling. Although performance is affected in most cases, it does not drop linearly with the core frequency. The benchmarks sensitive to core scaling (right hand side of figure 2) show almost no change in performance nor energy efficiency for memory scaling. Most of these have a high operational intensity, although there are exceptions.

4. CONCLUSIONS

This work has shown the potential of roofline-aware DVFS for GPUs. The theory of scaling the roofline model based on the operational intensity is valid in practice for the micro-benchmarks, although memory scaling does not achieve as much energy gains as core scaling. Real benchmarks show promising results for core scaling, although further work and experiments are required to take advantage of the potential.

5. REFERENCES

[1] A. Bakhoda, G. Yuan, W. Fung, H. Wong, and T. Aamodt. Analyzing CUDA Workloads using a Detailed GPU Simulator. In *ISPASS: International Symposium on Performance Analysis of Systems and Software*. IEEE, 2009.

[2] K. Berry, F. Navarro, and C. Liu. Application-level Voltage and Frequency Tuning of Multi-Phase Program on the SCC. In *ADAPT-3: International Workshop on Adaptive Self-Tuning Computing Systems*. ACM, 2013.

[3] S. Fuller and L. Millett. Computing Performance: Game Over or Next Level? *IEEE Computer*, 44, 2011.

[4] R. Ge, R. Vogt, J. Majumder, A. Alam, M. Burtscher, and Z. Zong. Effects of Dynamic Voltage and Frequency Scaling on a K20 GPU. In *PASA-2: Workshop on Power-aware Algorithms, Systems, and Architectures*, 2013.

[5] J. Lee, V. Sathisha, M. Schulte, K. Compton, and N. S. Kim. Improving Throughput of Power-Constrained GPUs Using Dynamic Voltage/Frequency and Core Scaling. In *PACT-20: International Conference on Parallel Architectures and Compilation Techniques*. IEEE, 2011.

[6] J. Leng, T. Hetherington, A. ElTantawy, S. Gilani, N. Kim, T. Aamodt, and V. Reddi. GPUWattch: Enabling Energy Optimizations in GPGPUs. In *ISCA-40: International Symposium on Computer Architecture*. ACM, 2013.

[7] X. Mei, L. S. Yung, K. Zhao, and X. Chu. A Measurement Study of GPU DVFS on Energy Conservation. In *HotPower: Workshop on Power-Aware Computing and Systems*. ACM, 2013.

[8] C. Nugteren, G.-J. v. d. Braak, and H. Corporaal. Future of GPGPU Micro-Architectural Parameters. In *DATE: Design Automation and Test in Europe*, 2013.

[9] S. Williams, A. Waterman, and D. Patterson. Roofline: An Insightful Visual Performance Model for Multicore Architectures. *Communications of the ACM*, 52:65–76, Apr 2009.