Models and Techniques for Green High-Performance Computing

Vignesh Adhinarayanan
Department of Computer Science, Virginia Tech
Blacksburg, Virginia 24061
avignesh@vt.edu

ABSTRACT

Modern high-performance computing (HPC) systems are power-limited. For instance, the U.S. Department of Energy has set a power envelope of 20MW for the exascale supercomputer expected to arrive in 2021-22. Achieving this target requires a 10.8-fold increase in performance over today's fastest supercomputer with only a 1.3-fold increase in power consumption. As a consequence, the architecture of an HPC system is changing rapidly—e.g., via heterogeneity and hardware overprovisioning. In my thesis, I address (i) modeling, (ii) management, and (iii) evaluation challenges concerning power and energy in this changing landscape of HPC systems. In this extended abstract, I focus on my work on modeling data movement power over interconnect wires.

1 INTRODUCTION

Historically, the high-performance computing (HPC) community has considered performance as the only primary design criterion [6]. In fact, even the notion of efficient supercomputing was viewed as a controversial topic when it was proposed in 2003 [5]. However, with the power consumption of the fastest supercomputers rapidly increasing [1], power and energy have emerged as first-order design criteria alongside performance. For instance, the U.S. Department of Energy has set a power envelope of 20MW for the exascale supercomputer expected to arrive in 2021-22. Achieving this target requires a 10.8-fold increase in performance over today's fastest supercomputer with only a 1.3-fold increase in power consumption. As such, we are now in the era of *power-constrained* supercomputing, where we seek to maximize the performance obtained by a supercomputer under some strict power constraint. To do so, we need to: (i) understand where power is being spent in today's hardware and (ii) manage the power consumed efficiently. In my thesis, I make the following contributions to support *power-constrained* supercomputing:

- **Accurate online power estimation for graphics processors.** In this work [4], we develop performance counter-based power models which are then refined at runtime using data from a low-resolution power meter in order to achieve high accuracy and high resolution for GPU power measurement. While adopting the best practices from CPU power modeling results in an average error of 6%, our online power modeling approach reduces the error to nearly 1%.
- **Targeted microbenchmarking to measure data movement power.** Past research on measuring data movement power on real hardware [8] failed to distinguish data movement power from data access power. In our work[3], we developed a novel approach based on the physical distance of data movement to measure interconnect power accurately and study its characteristics. Our evaluation shows that up to 14% of the dynamic power is consumed by the interconnect (which is less than what previous studies have suggested).
- **Power sloshing to maximize performance of a power-capped system.** In this proposed work, we seek to improve the performance of a power-capped system by (i) identifying the architectural component acting as the power (or energy) bottleneck and (ii) alleviating the bottleneck by making more power (or energy) available to these components. We propose to perform this re-allocation of power budgets dynamically at runtime aided by application phase prediction [7].
- **Principled evaluation of proposed techniques via PCA and clustering** To evaluate any idea in the systems area, we need to choose test cases carefully. To systematically select a concise, but useful, set of applications for evaluating the techniques we employ statistical methods such as principal component analysis (PCA) and hierarchical clustering [2]. Our evaluation shows that the four benchmark suites we studied (namely, Parboil, SHOC, Rodinia, and SPEC ACCEL) contains significant redundancy, and we can perform a thorough evaluation using only a fraction of the applications in these benchmark suites.

The rest of this extended abstract focuses on my work on measuring interconnect power on real hardware.

2 METHODOLOGY

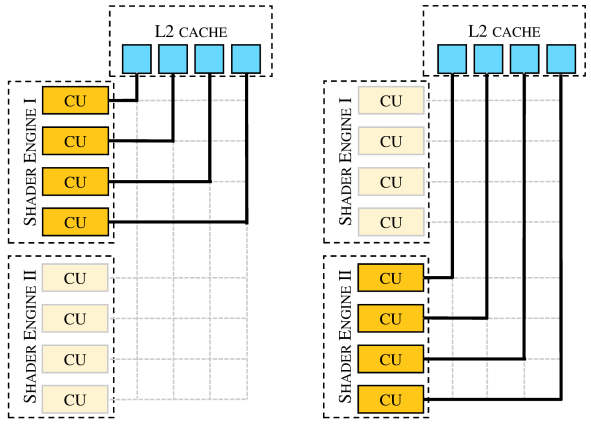

Figure 1: Design of our interconnect power microbenchmarks

Our microbenchmarking methodology is based on the observation that longer wires consume more energy than shorter wires

[1] The mean power consumption of the Top 500 supercomputers nearly trebled over a five-year period from 2008 to 2013 [1]

while carrying the same current. Therefore data that travels a longer physical distance within the chip consumes more energy than the same amount of data moving a shorter distance.

Our conjecture based on the above observation is that when we continuously move data from a partition of the L2 cache to the various L1 caches that are located in the different parts of the chip, we should observe a difference in power consumption. To test this conjecture, we design two microbenchmarks, illustrated in Figure 1. The first (referred to as **short-path**) continuously moves data between compute units (CUs) in shader engine I and the L2 quadrant closest to it. The second (referred to as **long-path**) moves the data between shader engine II and the same L2 quadrant, thereby moving the data through a longer physical distance.

Verification. When the above microbenchmarks were implemented on a AMD FirePro W9100 GPU, we found that long-path consumes 5% more chip-wide dynamic power than short-path, thereby confirming our conjecture.

Interconnect Power Model. Using our microbenchmark design approach, we characterized the impact of several important parameters—e.g., data movement distance, toggle rate, voltage, frequency, and interconnect bandwidth–on interconnect power. The characterization results were then combined into a parameterized equation which naturally lends itself to model interconnect power of larger applications, different chips, and different technology nodes. The general form of the parameterized equation can be expressed as follows:

Interconnect Power = Constant × % Peak Bandwidth × Toggle Rate × Distance × Scaled Frequency × Scaled Voltage^2

where *Constant* refers to the maximum power consumed by the interconnect for a given chip and a reference DVFS state.

3 RESULTS

In this section, we present the interconnect power of 22 OpenCL applications obtained from various sources. The results are presented for the 28 nm AMD FirePro W9100 GPU architecture and a hypothetical 7 nm shrink of the same die.

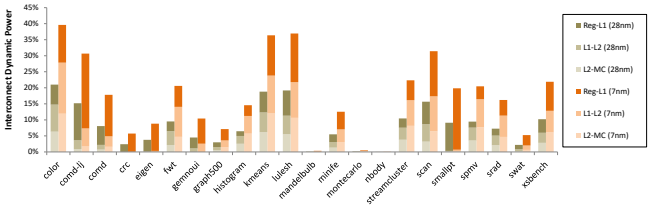

Figure 2: Percentage of the total dynamic power spent on the interconnect.

Figure 2 shows the power spent on the different parts of the interconnect, expressed as a percentage of overall dynamic power. Across applications, the on-chip interconnect consumes 5.6% of the total dynamic power on our GPU on an average. Within the interconnect, *register to L1* consumes the most power, using over 45% of the total interconnect power. The crossbar between L1 and L2 consumes 30% of the total interconnect power and the rest is consumed by *MC to L2*.

Among all applications, *color* shows the highest percentage of 14.3% for interconnect power. This is due to the fact that *color* is an irregular application with many branch and memory divergence, causing large amount of data accesses at different levels of the memory hierarchy. *Comd-lj*, *kmeans*, *lulesh*, and *scan* also consume significant amount of interconnect power, with over 10% of the overall dynamic power going towards the interconnect. Of these, *kmeans*, *lulesh*, and *scan* are either memory-bound or partially memory-bound, and understandably consume a greater amount of interconnect power as data has to be frequently fetched from the distant memory. *Comd-lj* is largely compute-bounded with most data accesses either going to register file or L1. Although the distance between the SIMD units and L1 is relatively small, it still has a significant amount of power spent in data movement because of the high data access counts to L1.

On the other extreme, applications such as *mandelbulb*, *montecarlo*, and *nbody* all consume nearly *zero* interconnect power. These are all compute-bounded, but unlike *comd*, the working set for these applications fits within the register files and therefore doesn't access L1 much. Therefore, they avoid short distance accesses as well and see a lower data movement power.

On the 7 nm architecture, the trends remain the same. But, the interconnect consumes 8.9% of the total dynamic power across applications. Individually, we see up to 21.9% for interconnect power as in the case of *color*. These values correspond to nearly 59% increase in the interconnect power for real applications. This highlights that data movement is going to be an even more significant problem in future GPUs.

4 CONCLUSION

In this work, we described a novel methodology to measure the interconnect power in *real* processors. We also characterized the interconnect power of 22 applications both in 28 nm technology and in a hypothetical 7 nm node. We showed that up to 14% of the dynamic power in these applications is spent on the interconnect which may increase up to 22% in the 7 nm node.

REFERENCES

[1] The Green500 List. http://www.green500.org/greenlists.
[2] Adhinarayanan, V., and Feng, W. An automated framework for characterizing and subsetting GPGPU workloads. In *2016 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)* (2016), pp. 307–317.
[3] Adhinarayanan, V., Paul, I., Greathouse, J. L., Huang, W., Pattnaik, A., and Feng, W. Measuring and modeling on-chip interconnect power on real hardware. In *2016 IEEE International Symposium on Workload Characterization (IISWC)* (2016), pp. 1–11. Best Paper Award.
[4] Adhinarayanan, V., Subramaniam, B., and Feng, W. Online Power Estimation of Graphics Processing Units. In *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)* (2016), pp. 245–254.
[5] Feng, W. Making a case for efficient supercomputing. *Queue 1*, 7 (2003), 54.
[6] Hsu, C., and Feng, W. The right metric for efficient supercomputing: A ten-year retrospective. In *Parallel and Distributed Processing Symposium Workshops, 2016 IEEE International* (2016), IEEE, pp. 1090–1093.
[7] Isci, C., Contreras, G., and Martonosi, M. Live, Runtime Phase Monitoring and Prediction on Real Systems with Application to Dynamic Power Management. In *Proceedings of the 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (Dec. 2006), pp. 359–370.
[8] Kestor, G., Gioiosa, R., Kerbyson, D. J., and Hoisie, A. Quantifying the Energy Cost of Data Movement in Scientific Applications. In *Proc. of the IEEE Int'l Symp. on Workload Characterization (IISWC)* (2013).