# Characterizing Memory Throttling Using Processor and Memory Performance

Bo Li
Virginia Tech
bxl4074@vt.edu

Edgar A. León
Lawrence Livermore National Laboratory
leon@llnl.gov

Kirk W. Cameron
Virginia Tech
cameron@vt.edu

## 1. INTRODUCTION

Memory bandwidth is one of the most important factors contributing to the performance of many HPC applications. Characterizing their sensitivity to this resource may help application and system developers understand the performance tradeoffs when running on multiple systems with different memory characteristics.

In this work, we emulate systems with different levels of memory bandwidth through memory throttling. The memory throttling approach adjusts attainable memory bandwidth of supercomputers by inserting a specific number of DDR Idle cycles between each DDR read and write operation. The higher number of DDR idle cycles we insert, the lower memory bandwidth we can get. Our previous work [3] has explored the memory throttling with the purpose of reducing power consumption. There are also some other simulation based approaches for power shifting and memory throttling [1, 2].

The contributions of our work include: we depict the change of performance roofline of systems as we use memory throttling to emulate different systems. We analyze the impact of memory bandwidth on performance using the Roofline model [5]. Meanwhile, we identify the pattern between memory bandwidth utilization and the maximum number of idle cycles inserted. Accordingly, we suggest an estimation approach for predicting the optimal memory throttling (no performance loss) for a given application or code region. This contributes to the goal of shifting power to other critical components (e.g. CPU) for improving the performance on a power-limited system. Compared with our previous prediction model [3], our new estimation approach can gain a higher $R^2$ value of 0.865. It is close to the best approximation with $R^2 = 1$.

## 2. EXPERIMENTAL SETUP

We use the BG/Q system on Vulcan cluster at Lawrence Livermore National Laboratory. Each node has 16 PowerPC A2 cores under 1.6 $GHz$, and 16 $GB$ 1.33 $GHz$ memory. Memory throttling interface can clock gating the peak memory bandwidth in the range from 0 to 126 idle cycles.

The application we used was the Livermore Unstructured Lagrange Explicit Shock Hydrodynamics (LULESH) mini-app. It is divided into five code regions from an application developer's perspective. We mainly focused on code regions 1, 2, and 4 due to their appropriate length of execution time. Based on our previous work [3], code region 1 and 2 are memory intensive and region 4 is classified as compute bound. We also use STREAM benchmark [4] to obtain the peak memory bandwidth under throttling.

## 3. RESULTS

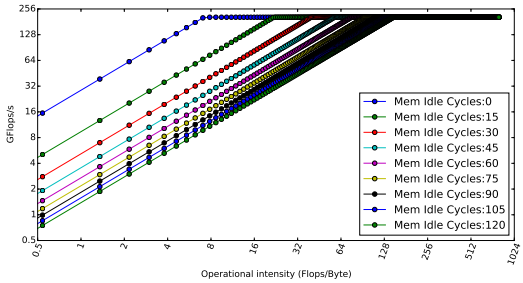


Figure 1: Impact of memory throttling on performance roofline.

The Roofline model [5] profiles the theoretical peak computation and memory performance ($GFlops/s$ and $GB/s$) of a given system. Figure 1 shows the change of performance roofline with memory throttling. The inflection point tells the operational intensity of an application who can achieve the best computation and memory performance. As we throttle the memory to reduce the peak memory bandwidth, the inflection point move towards the right side indicating a less amount of data transferred between CPU and memory. For an given application, the change of performance roofline on different systems could make its bottleneck change, for example, from CPU to memory.

Figure 2 shows the impact of memory throttling on memory bandwidth utilization (percentage of the peak bandwidth). The X-axis is the different configurations of LULESH (p is problem size, t is number of threads). The Y-axis is the percentage of peak memory bandwidth that has been utilized. The number on each bar is the maximum number of idle cycles inserted without hurting performance. We collected those numbers through an exhaustive search and captured the number where performance starts getting worse. Memory throttling can reduce the slack in memory bandwidth and help gain an extremely high memory bandwidth utilization, while the FP performance utilization is very low and far from being the performance bottleneck. While achieving a high memory bandwidth utilization through memory throttling, we can shift the saved power budget to other

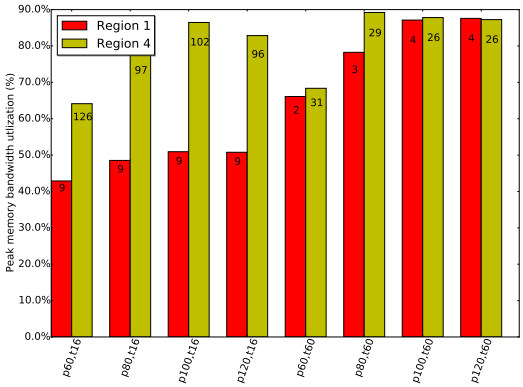


Figure 2: Percentage of peak memory bandwidth utilization under optimal memory throttling.

components like CPU for a better performance.

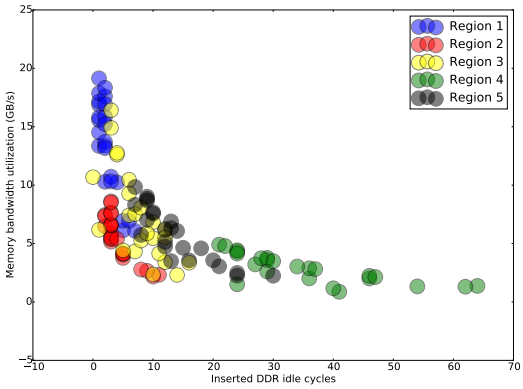


Figure 3: Memory bandwidth utilization versus optimal memory throttling.

To look into the relationship between memory bandwidth utilization ($GB/s$) and the optimal number of idle cycles, we collected some optimal memory throttling configurations of LULESH.The result is shown in Figure 3. All 5 code regions are included. The higher level of utilization an application can achieve, the less number of idle cycles we can insert without hurting performance.

The relationship between memory bandwidth utilization and optimal idle cycles can be approximated by a power function. In Figure 4, we approximated such a relationship using code region 1, 2, and 4, and achieved a high $R^2$ value as 0.8654.

## 4. CONCLUSIONS AND FUTURE WORK

In this work, we characterized the impact of memory throttling on the performance roofline using the Roofline model. We applied non-linear regression to approximate the relationship between the optimal memory throttling and bandwidth. The high value of $R^2$ shows the accuracy of our approximation. Future work includes proposing a prediction model for runtime memory throttling, validating our prediction model on other HPC applications, and dynamically shifting power by incorporating memory throttling with CPU DVFS.

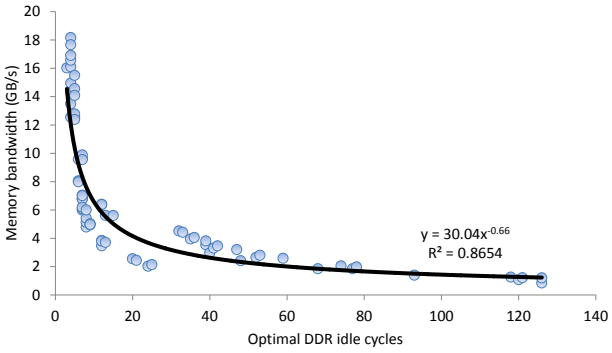


Figure 4: Approximation using power function. The performance loss threshold is 10%.

## 5. ACKNOWLEDGMENTS

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-ABS-676150.

## 6. REFERENCES

[1] Deng, Q., Meisner, D., Ramos, L., Wenisch, T. F., and Bianchini, R. Memscale: Active low-power modes for main memory. In *Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems* (New York, NY, USA, 2011), ASPLOS XVI, ACM, pp. 225–238.

[2] Felter, W., Rajamani, K., Keller, T., and Rusu, C. A performance-conserving approach for reducing peak power consumption in server systems. In *Proceedings of the 19th Annual International Conference on Supercomputing* (New York, NY, USA, 2005), ICS '05, ACM, pp. 293–302.

[3] Li, B., and León, E. A. Memory throttling on BG/Q: A case study with explicit hydrodynamics. In *6th Workshop on Power-Aware Computing and Systems (HotPower 14)* (Broomfield, CO, Oct. 2014), USENIX Association.

[4] McCalpin, J. D. Memory bandwidth and machine balance in current high performance computers. *IEEE Computer Society Technical Committee on Computer Architecture (TCCA) Newsletter* (Dec. 1995), 19–25.

[5] Williams, S., Waterman, A., and Patterson, D. Roofline: An insightful visual performance model for multicore architectures. *Commun. ACM 52*, 4 (Apr. 2009), 65–76.