Memory Access Cycle and the Measurement of Memory Systems

Xian-He Sun
Department of Computer Science
Illinois Institute of Technology, USA
sun@iit.edu

Dawei Wang
Department of Computer Science
Illinois Institute of Technology, USA
dwang31@iit.edu

ABSTRACT
Due to the infamous "memory wall" problem and a drastic increase in the number of data intensive applications, memory rather than processor has become the leading performance bottleneck of modern computing systems. Evaluating and understanding memory system performance is increasingly becoming the core of high-end computing. Conventional memory metrics, such as miss ratio, average miss latency, average memory access time, etc., are designed to measure a given memory performance parameter, and do not reflect the overall performance of a memory system. On the other hand, widely used system performance metrics, such as IPC and Flops are designed to measure CPU performance, and are not appropriate for memory performance. In this study, we propose a novel memory metric, Access Per Cycle (APC), to measure overall memory performance with consideration of the complexity of modern memory systems. A unique contribution of APC is its separation of memory evaluation from CPU evaluation. Simulation results show that APC is significantly more appropriate than existing memory metrics in evaluating modern memory systems.

Categories and Subject Descriptors
C.4 [**Performance of Systems**]: *design studies, measurement techniques, performance attributes.*

General Terms
Measurement, Performance

Keywords
Memory performance measurement; memory metric; measurement methodology

1. INTRODUCTION
Due to the unbalanced development of CPU and memory, the enlarging performance gap between processor and memory makes the memory system the dominant performance factor in high-end computing, and this gap is referred to as the "memory wall" [1] [2]. Intensive research has recently been conducted to improve the performance of memory systems. However, understanding the performance of modern hierarchical memory systems remains elusive for researchers and practitioners.

Conventionally used performance metrics, such as IPC (Instruction Per Cycle) and Flops (Floating point operations per second) are designed from a computing-centric point-of-view, and measure the overall computing performance. They are comprehensive and affected by a multitude of factors such as instruction sets, CPU micro-architecture, memory hierarchy, compiler technologies, etc., and as such are not appropriate measurements of the performance of a memory system. On the other hand, existing memory performance metrics, such as miss rate and average memory access time (AMAT), only measure a particular component of a memory system based on single memory access. They are useful in optimization and evaluation of a given component but cannot accurately characterize the performance of the memory system as a whole. Additionally, ILP technologies, such as Out-of-order, multithreading, speculation, etc., and advanced cache technologies, such as non-blocking cache, pipelined cache, and multibanked cache, are adopted in modern CPU micro-architecture to elevate the overall memory performance. These technologies make the relationship between memory access and processor execution even more complicated, since the processor could continue executing instructions or accessing memory under multiple cache misses. Thus, the influence of the improvement of one particular component in a memory system becomes increasingly tangled and elusive.

In this study, we propose a new memory metric, APC (Access Per Cycle), to evaluate memory system performance. Generally speaking, APC is measured as the number of memory accesses per cycle. More specifically, APC is the number of memory accesses requested at a certain memory level (ie: L1, L2, L3, Main Memory) divided by the number of memory access cycles at that level. Let M denote the total data access (load/store) at a certain memory level, and T denote the total cycles consumed by these accesses. According to the definition of APC,

$$APC = \frac{M}{T}$$

The term APC_D is used for L1 data cache, and APC_I is used for L1 instruction cache, which are respectively the number of L1 data or instruction cache accesses divided by the number of overall cache access cycles of their own. APC_{All}, which equals $APC_D \times APC_I$, represents a new metric to evaluate overall memory performance. Several outstanding memory accesses may co-exist in the memory system at the same time. In the APC definition, the total cycle T is measured based on the *overlapping mode*, which means when there are several memory accesses co-existing during the same cycle, T only increases by one. For memory accesses, the *non-overlapping mode* is adopted. That is, all the memory accesses issued are counted, including all successful or non-successful speculations and all concurrent accesses.

2. EXPERIMENTS AND RESULTS
The motivation for memory evaluation is the fact that the final system computing performance is heavily influenced by memory system performance. Therefore, an appropriate memory metric should reflect the system performance. The mathematical statistic variable *correlation coefficient* (CC) was used in this study to determine which memory metric most closely trends with the IPC variation. A superscalar CPU model in the M5 simulator [3] was adopted. A serial of cache configurations, which changed cache size, associativity, memory latency, or MSHR entries, were simulated. IPC and each memory metric were correlated to verify their variation similarity. The memory metrics compared include: Access per Cycle (APC), Hit Rate (HR, the counterpart of Miss rate), Hits per 1K instruction (HPKI, the counterpart of Misses per

1K instructions), average miss penalty (AMP), and Average Memory Access Time (AMAT). Each of these conventional memory metric has two measures which represent data cache performance only and comprehensive cache performance. It is expected that APC, HR, and HPKI would have a positive relation with IPC, with a CC value of (0,1]; also it is expected that AMP and AMAT would have a negative relation with IPC, with a CC value of [-1,0).

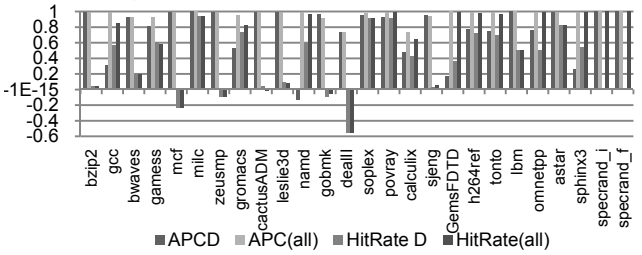

Figure 1. The Correlation Coefficients of APC and HR

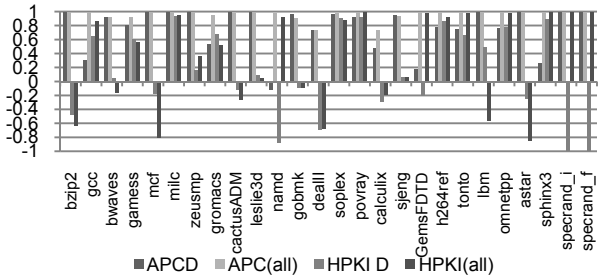

Figure 2. The Correlation Coefficients of APC and HPKI

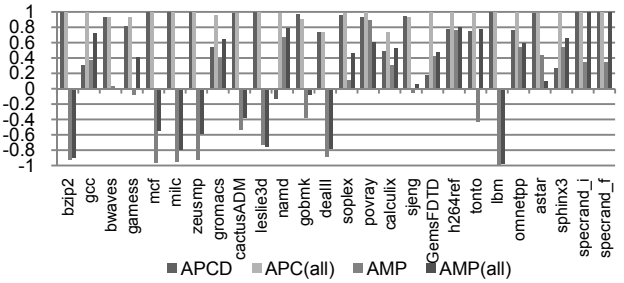

Figure 3. The Correlation Coefficients of APC and AMP

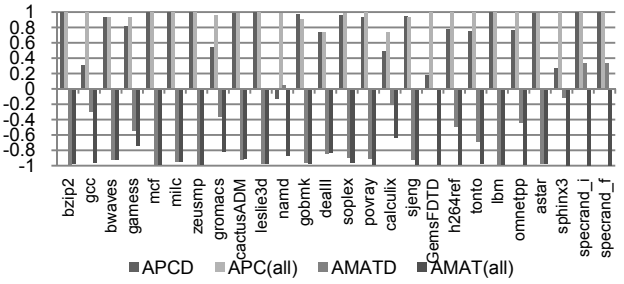

Figure 4. The Correlation Coefficients of APC and AMAT

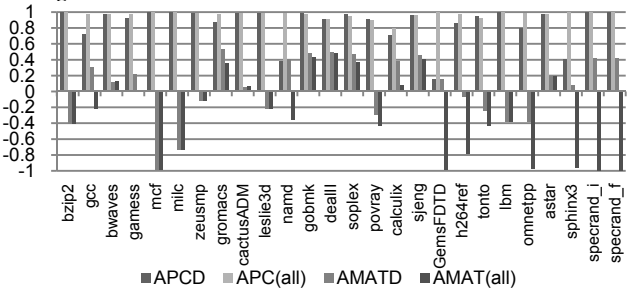

Figure 5. The Correlation Coefficients of APC and AMAT with MSHR changes

In Fig. 1 to Fig. 4, it can be seen that APC_{All} has the highest correlation coefficient value with IPC, with an average value for all applications of 0.9632, and this means that APC_{All} and IPC have a dominant relation.The $AMAT_{All}$ has the second closest relation with IPC, with an average value of -0.9393. For the other metrics, all have some misleading indications for some applications.

When changing cache access parallelism by altering the number of MSHR entries, Fig. 5 shows that APC and IPC still have the same dominant relation, with an average CC value of 0.9696. However, AMAT could not correctly reflect IPC for many applications. The reason is that AMAT does not catch the performance of advanced features such as overlapping and concurrency of modern memory systems.

3. APPLICATION OF APC

As the performance metric of memory systems, APC has many applications. Some are listed here.

According to the APC's definition, each level of memory hierarchy has its own APC values. In fact, each level's APC not only represents the performance of its memory level, but also includes all the lower levels of the memory hierarchy. For example, the value of APC_D represents the memory performance of L1 data cache, L2 cache and main memory. Therefore, by correlating IPC with APC at each level, we can find the lowest level that has a dominating correlation with IPC and can quantitatively detect the performance bottleneck inside the memory system.

As APC characterizes the overall memory performance, the IPC and APC correlation value provides a quantitative definition of data intensiveness. The idea is simple: if APC_M dominates IPC performance, then the application is data intensive. The degree of the domination provides a measurement of data intensiveness. We define an application is data intensive *if and only if* its correlation coefficient of APC_M and IPC is equal to or larger than 0.9. Therefore,

$$\text{Data Intensive Application} \equiv coe(APC_M, IPC) \geq 0.9.$$

Also, according to the mathematical definition of correlation coefficient, two variables have a dominant relation when the correlation value of two variables is equal to or larger than 0.9.

Another application of APC is that, in comparison with AMAT, it uniquely shows the performance gain of memory concurrency. In general, APC provides the means to study the matching between memory organization and microprocessor architecture, as well as a given application.

4. CONCLUSION AND FUTURE WORK

We have proposed a new memory metrics APC, given its measurement methodology, demonstrated its unique ability in measuring the overall performance of modern hierarchical memory systems, and discussed its applications. The full paper can be found at [4] [5].

5. REFERENCES

[1] X.-H. Sun, and L. Ni, Another View on Parallel Speedup, *Proc. of IEEE Supercomputing'90*, NY, Nov. 1990.

[2] W. Wulf and S. McKee. Hitting the wall: Implications of the obvious. *ACM SIGArch Computer Architecture News*, Mar. 1995.

[3] N. Binkert, R. Dreslinski, et al. The M5 simulator: Modeling networked systems. *IEEE Micro, 26(4):52*, Jul. 2006.

[4] X.-H. Sun, and D. Wang, APC: A Performance Metric for Memory Systems, *ACM SIGMETRICS Performance Evaluation Review*, 40(2) (2012).

[5] X.-H. Sun, and D. Wang, "Memory Access Cycle and the Measurement of Memory Systems", *Illinois Institute of Technology Technical Report* (IIT/CS-SCS-2011-09), 2011.