3D DRAM and PCMs in Processor Memory Hierarchy

Krishna Kavi[1], Stefano Pianelli[2], Giandomenico Pisano[2], Giuseppe Regina[2] and Mike Ignatowski[3]

[1]University of North Texas, Denton, Texas, USA
[2]University of Pisa, Italy
[3] Advanced Micro Devices, Austin, Texas, USA

Abstract— *In this paper we describe and evaluate two possible architectures using 3D DRAMs and PCMs in the processor memory hierarchy. We explore using (a) 3D DRAM as main memory with PCM as backing store and (b) 3D DRAM as the Last Level Cache and PCM as the main memory. In each of these configurations, since the proposed main memories are significantly faster than today's off-chip 2D DRAMs for main memory and either flash memory based SSDs or magnetic hard drives for secondary storage, we will introduce hardware assistance for virtual to physical address translation and to speed up page-fault handling.*

We use Simics, a full system simulator and benchmarks from both SPEC 2006 and OLTP suites to evaluate our designs. Our experiments measure energy consumed and execution performance; we use CACTI for obtaining energy and latency values for our memory configurations.

Index Terms—Memory hierarchy, 3D DRAMs, PCM, set-associate addressing, energy modeling, memory latency modeling.

1. Introduction

The purpose of this paper is to investigate different alternatives for using 3D DRAMs and PCMs in the memory hierarchy. More specifically, we will explore the following organizations:

a). 3D DRAM as main memory (we call this CMM) and PCM as secondary memory

b). 3D DRAM as Last Level Cache and PCM as main memory (we call this LLC).

Since 3D stacked DRAMs offer much lower access latencies (and higher bandwidths) than off-chip 2D DRAMs, and PCMs offer similar advantages over other technologies for secondary memory, we will assume hardware assistance for virtual to physical address translation, as well as different ways of viewing pages and how page faults are handled. Our feeling is that traditional memory management that relies on several levels of page tables for translating virtual addresses to physical addresses will effectively defeat the advantages of the new technologies. Moreover, since the time needed to transfer pages between PCM and 3D DRAM will be significantly less than that for transferring pages between magnetic disk drives and 2D DRAMs, kernel intervention leading to process context switches on page faults should be minimized.

In this paper we will evaluate our memory organizations and associated hardware needed to achieve our objective; we use execution performance and energy consumption as evaluation metrics. We use several different benchmarks drawn from both SPEC 2006 and OLTP suites, and vary benchmark mixes running on different cores in a multicore

system. We use Simics, a full system simulator for our simulations, and CACTI for evaluating latencies and power requirements for our organizations.

The rest of the paper is organized as follows. In the next section we will review research that is very closely related to ours. In Section III, we will describe the underlying hardware components for our memory architectures. Section IV shows the results obtained using CACTI models for 3D DRAM memories along with the additional hardware structures (primarily SRAMs) and PCM memories. Using values for access latencies and power taken from CACTI simulations, we evaluate our memory organizations for executing various benchmarks. The experimental setup is described in Section V. Section VI analyzes the results.

II. Related Works

There are several methods used for stacking two or more dies: wafer-to-wafer bonding, die-to-die bonding and die-to-wafer bonding with different kinds of overlays. We will assume die-to-die technology with face-to-face overlays [7]. Stacking technology allows for the reduction of wire lengths by introducing vertical connections between dies called *Through Silicon Vias* (TSV) [6]. 3D stacked DRAMs appear to be an obvious way to take advantage of the new technology, and overcome memory access delays [8][9]. By using high capacity DRAM dies and using several die-to-die connections we can greatly reduce memory access latencies and increase bandwidths [10][11]. Several studies have shown that 3D DRAM memory may also reduce energy consumed by applications while improving performance, particularly when the memory layers are organized as *True 3D* [12]. In a *True 3D DRAM* organization, the $N-1$ upper layers contain only DRAM bit-cells. Layer 1 contains only the control logic such as sense amplifiers, row decoders, row buffers etc. In true 3D organization, ranks and banks of DRAM cross multiple layers to reduce the length of data paths and increase clock frequencies. In our work we assume that all the extra logic such as SRAMs needed for our cache like indexing, row buffers, and other components of a memory controller, are placed on the same logic layer (i.e. layer 1). In fact since layer 1 is dedicated to these functions we feel that it should have more than adequate area to accommodate our requirements. We use CACTI to model True-3D organization and obtain latencies and energy values for our memory organizations.

Qureshi et. al., [1] have studied the use of PCMs as main memory with a small 2D DRAM as a buffer to both speedup accesses and reduce write-backs to PCM. In particular, they focused their work to study the effect of overall system performance by adding PCM as a complement to the DRAM memory. The DRAM buffer is organized similarly to a hardware cache that is not visible to the OS, and is managed by the DRAM controller. In our study, however, we evaluate different memory organizing using 3D DRAM and PCM in the memory hierarchy. The study by Lee, et. al., [13] is similar to that of Qureshi [1], in that they also use PCM as a replacement to DRAM as main memory. Like Qureshi, Lee uses small DRAM based buffers between last level caches and PCM, to reduce the amount of data written back to PCM. However, Lee studies the use of multiple DRAM buffers instead of a single DRAM based cache.

Our previous studies [16] have provided an organization and named it Cache Main Memory (CMM). The idea behind the CMM organization is that, since 3D DRAMs have lower latencies and higher bandwidths, allowing them to appear both as cache and as main memory can be advantageous. This duality makes the memory perfect either for operations

that are more efficient if they use cache like addressing (fast address translation) or for operations that require main memory like organization (DMA, shared pages, OS management of memory). Our previous studies were limited since: (a) they did not provide details on the hardware needed, (b) they did not provide results on energy consumption, (c) they used access latencies for 3D DRAM and PCM memories that were simplistic and rely on average values, and (d) benchmarks did not seem to fully stress the memory architectures. In this paper we have addressed these limitations.

There have been other studies that are aimed at improving performance of PCM-based memory system and reduce the amount of data written back to PCMs [2][3] [4], [13]. These approaches are orthogonal to our studies since they can be applied within our organizations.

III. FOUNDATION OF THE ARCHITECTURE

The cache like indexing mentioned above with CMM [16] designs allows us to minimize the number of levels of page tables needed by the OS for translating virtual addresses to physical addresses. We assume that OS will use one or two levels of page tables and map a *large* virtually addressed region (or segment) to a *smaller* physical region (segment) in main memory. We assume that a virtual segment has k pages but physical segments contain fewer than k pages – pages in a virtual segment compete for pages in a physical segment similar to cache lines (see Figure 1).

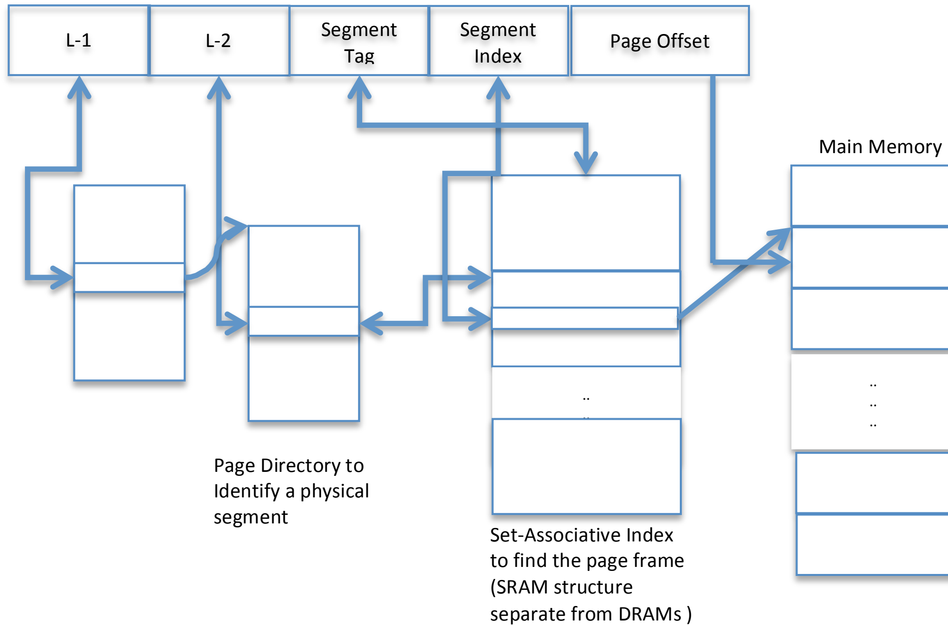

Figure 1: Cache Like Indexing for Virtual Address Translation

In this paper we assumed 1024 pages per virtual segment, and use 64 pages per physical segment. The sizes of virtual and physical segments can be varied based on the size of the main memory, the number of page tables that must be looked up

during translation and the number of tag bits needed for cache-like addressing of pages in a segment. SRAM structures store the virtual page numbers associated with pages that are currently in these physical pages. We use set associative search through the sets belonging to a physical segment to find the desired page. Once found, the newly obtained physical address is stored in TLB for future accesses; to further speedup the translation TLBs are used. Using larger virtual segments will require more tag bits.

3D-DRAM as LLC

In a configuration reported in this paper we explore using 3D DRAM as LLC (instead of SRAM based caches), and PCM as main memory. 3D DRAM as last level cache should be distinguished from traditional SRAM based caches. 3D LLC are divided into two components:

1. One component built with SRAM logic, which implements cache-like indexing and holds tags (usually on the logic layer of the 3D organizations).
2. The other component is implemented with DRAM logic and stores the actual data contained in LLC

For very large DRAMs when used as LLC, the number of lines of data in LLC will be large and thus the number of SRAM entries will also be very large. However, note that most SRAM based level 3 caches in current processors require a large portion of a chip (as much as 50%). Since we eliminate the traditional SRAM based level 3 cache, we feel that the saved area can be used to build a SRAM to hold the tags for DRAM based LLCs.

The actual data contained in the LLC will be located in the 3D DRAM. The SRAM location with matching tags will be used as an index into DRAM to find the desired data. When using 3D DRAM as LLC, the size of a single memory line is set to 1024 bytes, or 8 times larger than a typical cache line. The underlying memory controller will transfer data equivalent to a cache line to $L2$ caches.

PCM

When 3D DRAM is used as LLC, PCM will be used as main memory and we propose to use the same virtual to physical address translation described previously with the CMM organization for accessing PCM pages (Figure 1). This requires SRAM based tag structure (as with CMM organization described previously). For PCM as main memory configuration, the internal organization of PCM is similar to a DRAM organization using ranks, banks and row buffers.

IV. CACTI MODELS

We modeled SRAM, PCM and 3D DRAM memories using CACTI (in particular we used CACTI-3DD [15] to simulate 3D DRAMs), in order to obtain very accurate design parameters for estimating delays and power requirements of the components used in our architectures. More specifically we obtain values for: *memory access times, cycle times, area and dynamic power*. We used all these parameters taken from CACTI within our simulations for obtaining execution performance results and overall energy consumed by benchmark applications. We explored different sizes and associatives for TLBs and as can be seen from Figure 2, a 2048 entry 8-way associative TLB provides a good compromise between performance and energy consumed by TLB hardware.

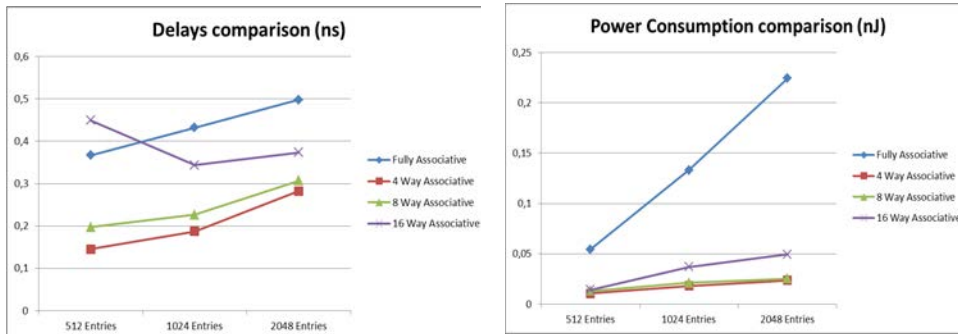

Figure 2. TLB latencies and Power requirements

In addition, we use SRAM structure to contain tags representing (partial) physical addresses of currently resident physical pages (see Figure 1). The size of the SRAM depends on the size of the main memory, since a tag is stored in SRAM for each main memory page. SRAM is also used when 3D DRAM is used as the Last Level Cache. The size of the SRAM depends on the number of cache line in 3D DRAM. We use 8-way associativity. Figure 3 shows the results for different SRAM sizes -- x-axis represents DRAM sizes for which SRAMs contain tags.

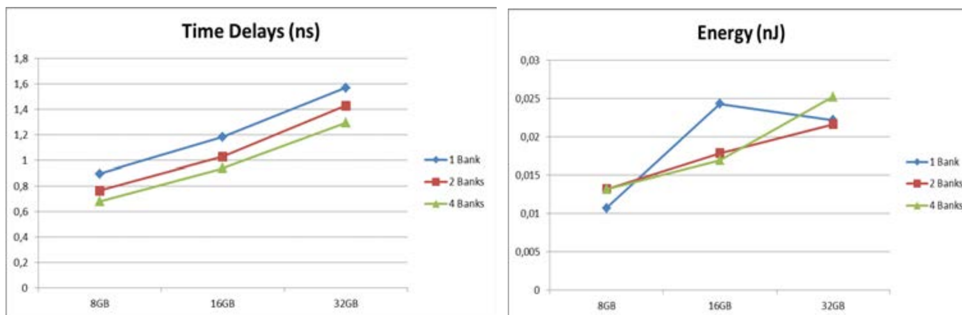

Figure 3. Evaluation of SRAM latencies and Energy

We also used CACTI to model 3D DRAM. We chose 8, 16 and 32 GB for our DRAMs, but varied the number of ranks and banks. The number of channels has been fixed to 1. This choice actually penalizes the memory parallelism but it also provide a very simple entry-level 3D-DRAM for the *True-3D* configuration. We decided to use 4 ranks similar to the research described in [8] and [12]. The number of banks and the number of dies have been varied in our experiments (see Figure 4). All the results indicate designs with 8 dies (8 layers of DRAM cells) are not the best choice for our organization: they consume more energy and cause longer latencies. We notice that among 4 die alternatives, larger memories perform better with more banks: 16 banks for 8 GB, 32 banks for 16 GB and 64 banks for 32 GB.

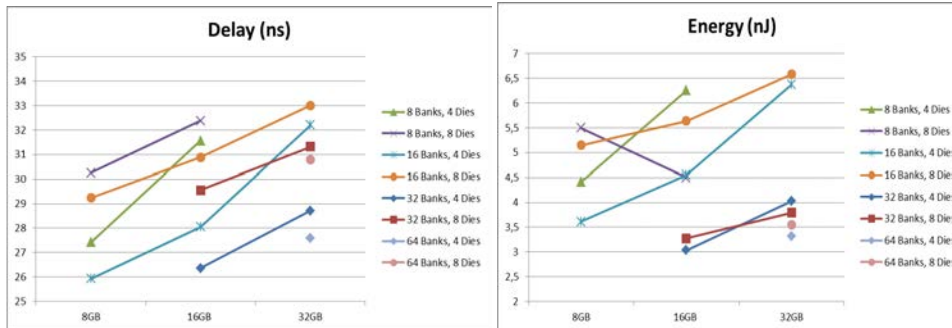

Figure 4. CACTI models for 3D DRAM

Initially we explored available CACTI extensions for modeling PCM, such as the NVSim [17]. However this tool proved to be not useful for our study because it only simulates PCM at a bank level while we needed to simulate a complete PCM memory device with multiple banks. So we followed the work of Qureshi [1]. Basically, if we use a PCM with x GB, its delays and energies will be 4 times those of a 2D-DRAM with x/4 GB capacity

V. EXPERIMENTAL SETUP

To simulate the different memory architectures we described in this paper we used Wind River Simics, a full system simulator. Simics includes several tools and modules that can be used to model user-defined architectures and components. Since we are only studying memory subsystem, the module of interest to us is the G-Cache. This module was originally designed to simulate simple caches but can easily be expanded to simulate any memory hierarchy.

We used benchmarks from SPEC 2006 and OLTP suites. Since we used a 4-core $x86-64$ “Hammer” system in our simulations, we created several benchmark mixes (mixes of 4 benchmarks each) to test our architecture as shown in Table 1 and Table 2. We will refer to each mix of 4 benchmarks by the name given in the first column of the table[1].

For L-1 and L-2 caches we used the same configurations as those of [20]. Per core L-1 caches are 32KB, 128 byte lines and use 4 way associative; per cores L-2 caches are 256KB, 8-way associative and 128byte lines.

Baseline

In order to evaluate the efficiencies of our proposed organizations we defined a generous baseline system. The system includes an *infinite* 2D DRAM for main memory. Thus it does not encounter page faults. However the system relies on slower 2D technology. The latencies and the energies modeled are taken from commercially available DDR3 DRAMs with 1GB for each bank. Also the baseline uses traditional 4K pages (unlike 32KB pages used for 3D DRAM organizations of our work) and relies on multiple

[1] Thus when we say ‘Gobmk’ or any other benchmark name, are actually referring to corresponding mix and not a single benchmark application.

levels of page tables for virtual to physical address translation. It uses a finite sized TLB and thus can encounter penalties on TLB misses. We modeled the baseline with TLB miss penalties using data for commercial systems using AMD processors. We felt that using a very generous baseline allows us to see the true benefits of new memory technologies.

Table 1: SPEC 2006 Benchmark Mixes

Mix Name	Bench 1	Bench 2	Bench 3	Bench 4	Total (GB)
Gobmk	Gobmk	Hmmer	H264Ref	Gromacs	0.046
Gamess	Gamess	Sphinx3	Tonto	Namd	0.027
Sjeng	Sjeng	Libquantum	Leslie3d	Astar	0.192
Omnetpp	Omnetpp	Astar	Calculix	Gcc	0.140
Milc	Milc	Wrf	Zeusmp	Soplex	0.866
Zeusmp	Zeusmp	Leslie3d	Gcc	CactusADM	0.718
GemsFTD	GemsFDTD	Mcf	Bwaves	CactusADM	2.262
Mcf	Mcf	Zeusmp	Milc	Bwaves	1.656

Table 2: OLTP Benchmark Mixes

Mix Name	Bench1	Bench2	Bench3[2]	Bench4	Total (GB)
Auction Mark	Auction Mark	Auction Mark	Sjeng	Stream	20 ÷ 25
Seats	Seats	Seats	Sjeng	Stream	20 ÷ 25
Tatp	Tatp	Tatp	Sjeng	Stream	20 ÷ 25
Epinions	Epinions	Epinions	Sjeng	Stream	20 ÷ 25

VI. Results and Analysis

A). CMM (3D DRAM as main memory)

In the first memory organization (or CMM that uses 3D DRAM as main memory) we used different TLB configurations (capacities and associativity). Charts 1 and 2 depict results obtained with the tested configurations, compared to the baseline, using SPEC2006 benchmark mixes. *For these experiments, we used a 8GB 3D DRAM since the memory footprints for SPEC2006 benchmarks is relatively small. We explored larger 3D DRAM sizes for OLTP benchmarks.* Although the baseline consists of an infinite 2D DRAM, the baseline does not always outperform our CMM. There are several reasons for this. First, the benchmarks used have finite memory footprints, often smaller than the 3D DRAM configurations we used - thus infinite 2D DRAM offers no special advantage. Second, conventional off-chip 2D DRAMs are significantly slower than 3D DRAMs. And baseline uses 4 KB pages (compared to 32 KB in 3D DRAM). This requires more frequent accesses to TLB and page tables for address translations.

[2] We used Sjeng and Stream benchmarks along with OLTP in these mixes to represent server environments that may be presented with large footprint applications along with heavy processing load benchmarks.

It appears that for CMM (3D DRAM as main memory), 8-way 2048 entry TLB performs better than other configurations. *In subsequent experiments we will use this TLB configuration.* The chart shows that on average, this configuration performs 18% *better than the baseline*.

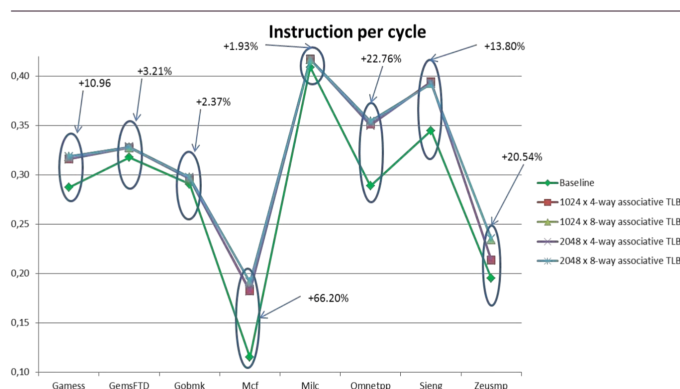

Chart 1: IPC - SPEC 2006 for CMM architecture (using different TLB configurations)

Let's now look at energy consumed. Although the baseline configuration contains infinite DRAM, in order to estimate power values for the baseline (infinite 2D DRAM) we used sizes that are comparable to the 3D DRAM used in CMM organization. Looking at the Chart 2, it should be noted that even under this assumption (finite energy consumption), our CMM system has comparable performance in terms of energy requirements for SPEC2006 mixes. *And most interestingly, TLBs and SRAM structures needed for CMM consume less than* 1% *of the energy used by the CMM memory system (detailed data not included in this paper, but similar observation can be made from Chart 8).*

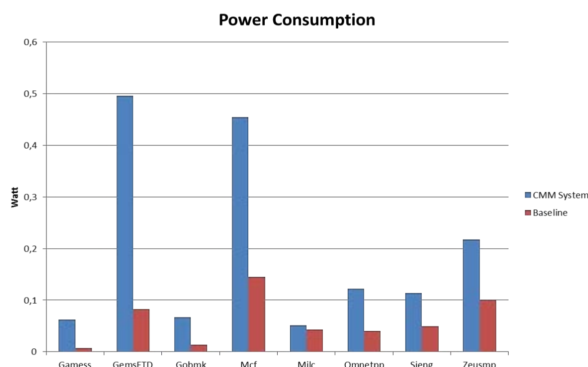

Chart 2: Energy Consumption - SPEC 2006 for CMM architecture (using 2048 8-way TLB)

OLTP benchmarks are characterized by a large memory footprint, in the order of $20 - 25\ GB$, and represent Cloud applications. As can be expected, for these applications the baseline's infinite DRAM becomes advantageous and outperforms our CMM (see Chart 3). And within CMM, lager memory footprint applications perform better with

larger TLBs (Chart 3 shows data for different TLB sizes). Note that having a large 3D DRAM is not always beneficial - in some cases the longer latencies associated with larger 3D DRAMs can defeat the larger capacity (unless more than 4 dies are used; we used 4 dies). This can be seen when 32 GB 3D DRAM is used (which is more than sufficient to fully contain the OLTP benchmarks) the performance, in terms of IPC is worse than the baseline. It should be noted that while baseline does better than CMM, the performance differences are not significant. In reality a practical 2D DRAM based system will face several additional delays due to page faults.

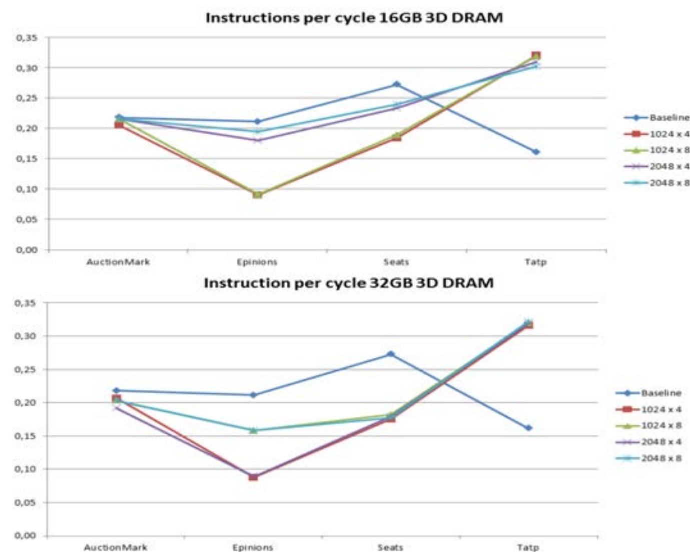

Chart 3: IPC - OLTP for CMM architecture

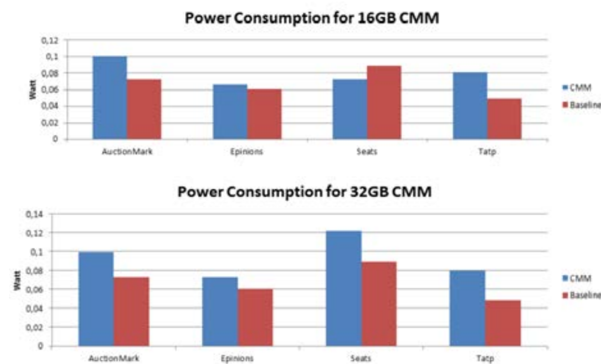

Chart 4: Energy consumption – OLTP for CMM architecture

Let's now consider energy performance for OLTP benchmarks. Chart 4 confirms what was stated before: even though CMM power consumption is still greater than the baseline, the values are within comparable range; in some cases, CMM actually has lower energy values than the baseline. Note that the baseline energy values are based on 2D DRAM

sizes that are comparable to our 3D DRAM sizes used in CMM. If the baseline included a magnetic disk as a backing store, the power requirements for that configuration would be significantly greater than that for our organizations.

B). 3D-DRAM as Last Level Cache (LLC)

Charts 5 and 6 clearly show that our configuration that uses 3D DRAM as LLC and PCM as main memory outperforms the baseline configuration (with infinite 2D DRAM).

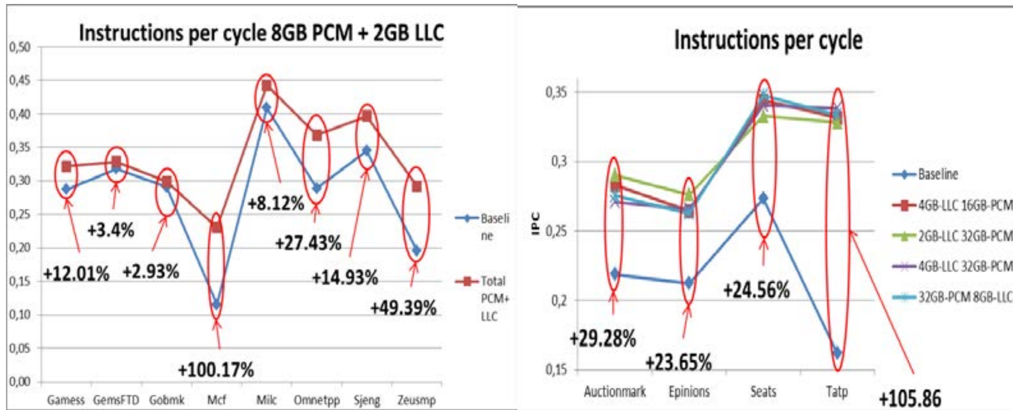

Chart 5: IPC for SPEC 2006 for LLC

Chart 6: IPC for OLTP for LLC

The execution performance gains are particularly impressive for mcf (SPEC2006) and Tatp (OLTP)[3]. The average performance gains for SPEC2006 benchmark mixes are $+27.30\%$ and $+45\%$ for the OLTP mixes. An observation that should be emphasized from the data shown above is that using larger than $\mathbf{2}$ GB 3D DRAM for LLC shows insignificant performance improvements, regardless how large the PCM is. This may be in part because of our system, which has only 4 cores and the nature of these benchmark programs. Our experiments show that $\mathbf{2}$ GB 3D DRAM as LLC and $\mathbf{32}$ GB PCM as main memory is the best choice.

Looking at Charts 7 and 8, the significant energy consumed in this configuration is due to the 3D DRAM (as LLC), but the energy consumed increases only marginally when the size of the DRAM is doubled. This allows us to choose, depending on our needs, the best alternative: for instance a greater last level cache may be more useful when the application can use more cache capacity, or use smaller caches to save energy and cost of the system.

[3] Note these names refer to benchmark mixes and not single benchmark

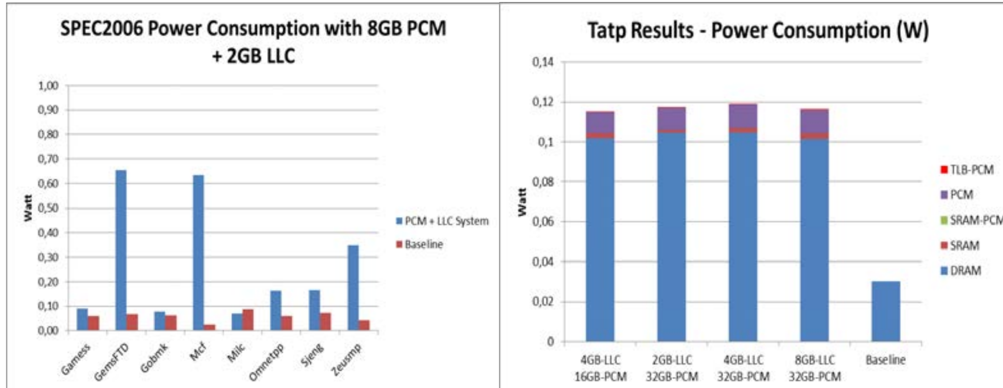

Chart 7: Power consumption for SPEC 2006 **Chart 8: Power consumption for OLTP**

VII. CONCLUSIONS

Memory wall [8], which refers to the disparity of speeds between processors and memories, is still a major problem limiting the performance that can be achieved with modern processor technologies. Some new memory technologies may alleviate this problem to some extent. They include 3D DRAM memories and Phase Change Memories. These technologies present opportunities and challenges when they are included in processor memory hierarchy.

In this paper we explored two different memory organizations for using 3D DRAMs and PCMs. Each configuration has associated advantages and disadvantages, differing in execution performance, energy consumed and cost. Our goal is to provide initial data that may guide choices on how these new technologies can be used.

3D DRAM as LLC configuration, with PCM as the main memory, achieves best results in terms of execution times, but consume more energy than the other configuration. This is due to larger SRAMs needed to store tags for a large LLC (we separate the tags form DRAM and store them in SRAM for fast access to tags).

The CMM (3D DRAM as main memory and PCM as secondary memory) configuration require higher execution times than the previous case, but require lower energies. This is expected because CMM uses a longer path to its data using larger pages; at the same time, we needed smaller SRAMs. In this configuration, SPEC2006 benchmarks mixes achieve reasonably comparable results as the baseline, since they exhibit smaller memory footprints, unlike OLTP benchmarks where CMM performs worse than the baseline.

Acknowledgement: This work is supported in part by the NSF Net-Centric I/UCRC, AMD and other industrial members

VIII. References

[1] M. Qureshi, S. Gurumurthi and B. Rajendran. *Phase Change Memory – from Devices to Systems*, M. &. C. Publishers, Ed., Morgan & Claypool Publishers, 2011.

[2] M. Qureshi, V. Srinivasan and J.A. Rivers. "Scalable high performance main memory system using phase change memory tecnology," in *Proceedings of ISCA-2009, pp 24-33*, 2009.

[3] M. Quereshi, "Improving read performance of phase change memories via write cancellation and write pausing".

[4] M. Quereshi, M. Franceschini, A. Jagmohan and L. Lastras. "PreSET: Improving performance of phase change memories by exploiting asymmetry in write times," in *ISCA-2012, pp 380-391*, 2012.

[5] W. A. Wulf and S. A. McKee, "Hitting the Memory Wall: Implications of the Obvious," *Computer Architecture News,* p. 23(1), 20-24 March 1995.

[9] J. Lau, Through-Silicon Vias for 3D Integration, McGraw-Hill Professional, 2012.

[7] G. Lecarpentier and J. D. Vos., *Die 2 Die Bonding,* SET S.A.S. (Smart Equipment Technology), 131 Impasse Barteudet, 74490 Saint Jeoire, France & IMEC, Kapeldreef 75, Leuven B-3001, Belgium, 2012.

[8] G. Loh, "Computer Architecture for Die Stacking," *International Symposium on VLSI Technology, Systems, and A6plications (VLSI-TSA),* vol. N/A, p. N/A, 2012.

[9] C. Liu, "Bridging the processor-memory gap with 3D IC technologgy," *IEEE Design and Test,* pp. 564-565, 2005.

[10] H. Sun, "3D DRAM design and application to 3D multicore systems," *IEEE Design & Test of Computers,* pp. 36-46, 2009.

[11] C. Weis, "Design space exploration for 3D-stacked DRAMs," in *Proceedings of DATE-11*, 2001.

[12] G. Loh, "3D-Stacked Memory Architectures for Multi-core Processors," in *Computer Architecture, 2008. ISCA '08. 35th International Symposium on*, 2008.

[13] B. C. Lee, E. Ipek, O. Mutlu and D. Burger, "Architecting phase phange memory as a scalable dram alternative.," *Sigarc Comput. Archit. News ,* vol. 3, no. 37, pp. 2-13, 2009.

[14] T. Barr, A. Cox and S. Rixner, "Translation caching: skip, don't walk (the page table)," in *ISCA-2010*, Saint-Melo, 2010.

[15] S. J. E. Wilton and N. Jouppi, "CACTI: an enhanced cache access and cycle time model," *Solid-State Circuits, IEEE Journal of,* vol. 31, no. 5, pp. 677-688, 1996.

[16] J. Sherman, K. Kavi, B. Potter and M. Ignatowski, "A Multi-core Memory Organization for 3-D DRAM as Main Memory," in *Architecture of Computing Systems ARCS 2013*, vol. 7767, H. Kubaitova¡, C. Hochberger, M. Danak and B. Sick, Eds., Springer Berlin Heidelberg, 2013, pp. 62-73.

[17] D. Xiangyu, X. Cong, X. Yuan and J. Norman P., "NVSim: A Circuit-Level Performance, Energy, and Area Model for Emerging Nonvolatile Memory," *IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS,* vol. 31, no. 7, pp. 994-1007, July 2012.