Near Volatile and Non-Volatile Memory Processing in 3D Systems

Maryam S. Hosseini, *Student Member, IEEE,* Masoumeh Ebrahimi, *Senior Member, IEEE,* Pooria Yaghini, *Member, IEEE,* and Nader Bagherzadeh, *Fellow, IEEE*

Abstract—The cost of transferring data between the off-chip memory system and compute unit is the fundamental energy and performance bottleneck in modern computing systems. Furthermore, with the advent of emerging data-intensive applications and technology scaling, this bottleneck has continuously increased. To overcome these difficulties, Near Memory Processing (NMP) based on 3D die stacking becomes a potential technology to transform the *computation-centric* system towards *memory-centric* system. In this work, we explore the feasibility and efficacy of a NMP architecture based on an emerging Non-Volatile Memory technology (NVM) for data-intensive applications and compare it with the conventional 3D-stacked NMP architecture based on DRAM. We demonstrate the effectiveness of our approach with experimental results.

Index Terms—Near Memory Processing, Data-Intensive Applications, 3D-Stacked Memory Technologies, Emerging Non-Volatile Memory Technologies.

1 Introduction

3D-stacked memory technology is one of the most promising solutions to address the *memory wall* problem in modern computing systems [1] [2]. Micron's Hybrid Memory Cube (HMC) [3] [4] and JEDEC's High Bandwidth Memory (HBM) [5] are examples of this memory technology. This technology enables stacking multiple high capacity memory layers vertically on top of a logic tier using short and fast Through-Silicon Vias (TSVs) bus within one package and provides a massive internal memory bandwidth with lower power consumption and latency [21] [22] [6]. It has been reported that an HMC package offers an internal bandwidth of 160GB/s to 320GB/s while providing a high-level vault (vertical partition composed of multiple memory banks) parallelism [3].

In the era of big data and with the advent of emerging data-intensive applications, researchers have witnessed the inefficiency of conventional CPU-centric processing systems when running large data sets. Data-intensive applications increase power and bandwidth pressures to the memory system. Tackling these challenges, researchers have proposed *Near Memory Processing (NMP)* architecture based on 3D-stacked memory technology that integrates processing units within memory package to offer higher memory bandwidth to the processing units. NMP architecture exhibits a significant potential for performance and energy efficiency, since it reduces the aggregate need for transferring data within large memory hierarchy.

The next promising innovation for the next generation memory systems is the use of byte-addressable cutting edge Non-Volatile Memories (NVMs). Phase Change Memory (PCM) [7], Spin-Torque Transfer Random Access Memory (STTRAM) [8], and Resistive Random Access Memory (ReRAM) [9] are examples of emerging NVMs (with different characteristics) explored by researchers and manufactures for replacing DRAM at the main memory layer. The expectation from NVM types is to provide larger capacity per chip, memory access latency and energy consumption (low-power) competitive to the DRAM technology, and better technology scaling. Across emerging NVM technologies, PCM is considered as the most mature one that can benefit from more reduction in the switching power and can scale better than DRAM technology [7] [10]. It has been reported that the PCM is expected to scale to 9nm in the near future which introduces memories with higher density that can meet the capacity requirements of many-core computing systems [11].

The goal of this paper is to motivate the efficiency of NMP subsystems to process data-intensive applications when 3D-NVM technology is employed. In this paper we make the following contributions:

- We perform a detailed characterization (Roofline, data locality, and memory access behavior analysis) for various set of applications as a case study. The analysis evaluates the potential benefits of NMP architecture over conventional Host CPU processing to accelerate processing data-intensive applications in an efficient way.
- We explore two NMP architectures based on different 3D-stacked memory technologies (3D-DRAM and 3D-PCM) and analyze the impact of constructing NMP architecture based on emerging non-volatile memory technology (PCM).
- We show that executing certain data-intensive applications on NMP architecture based on 3D-NVM can improve performance and reduce power consumption versus executing on a conventional Host CPU

- *M S. Hosseini, P. Yaghini and N. Bagherzadeh are with the Department of Electrical Engineering and Computer Science, University of California, Irvine, CA, 92697.*
  E-mail: {mseyyedh, pooriam, nader}@uci.edu
- *M. Ebrahimi is with the Department of Electronics and Embedded Systems, KTH Royal Institute of Technology, Sweden.*
  E-mail: mebr@kth.se

and 3D-DRAM based NMP.

The rest of this paper is organized as follows: Section 2 provides a brief background related to the conventional Host CPU and near memory processing, and then describes different architectural techniques proposed to tackle performance and energy problems in processing data-intensive applications. Section 3 describes the methodology used for application characterization and how to leverage the proposed metrics. Section 4 explains the evaluation methodology and the experimental platforms. Section 5 presents the analysis and results. Finally, Section 6 concludes the paper.

2 Background and Motivation

In this section, we provide an overview of conventional Host CPU and NMP architecture. Then, we explain prior research works relevant to architectural techniques proposed to tackle performance and energy problems when dealing with processing data-intensive applications.

2.1 Conventional Host CPU and NMP Subsystem

In conventional Host CPU processing (shown in Fig. 1.a), data shuttles back and forth between the off-chip DRAM and the processing unit. This data movement is a major performance bottleneck when dealing with data-intensive applications such as media processing, data mining, computer vision, machine learning, computational biology, and speech recognition. In this section, as the background, we describe different architectural techniques which are proposed to tackle latency and energy problem when processing data-intensive applications.

Fig. 1 depicts an abstract view of a system that is capable of processing close to memory in which the NMP subsystem is connected to Host CPU through high-speed links. Host CPU can offload kernel to the NMP subsystem. NMP transfers data through high-bandwidth and low-energy 3D interconnects between memory layers and cores in the logic layer. The NMP subsystem (Fig. 1.b) consists of a 3D processor-memory architecture, in which processing cores are embedded in the logic layer and memory layers are stacked vertically on top of it. Fig. 2 illustrates a conceptual view of a NMP architecture. The logic layer composed of multiple vault logics, connected to each other through an interconnect network such as Network-on-Chip (NoC). NoC is the dominant communication infrastructure which provides a scalable efficiency in hardware area and power [12] [13]. The memory is divided into multiple vertical partitions called vaults in which each vault has its own memory controller in the logic layer. Each memory layer consists of multiple independent vaults. Each of these vertical vaults can be accessed in parallel as they have independent processing cores and memory controllers in the logic layer.

In this work, the modeled NMP subsystems are based on 3D-DRAM and 3D-PCM technologies. One of the main potential applications of NMP architectures is deep learning applications, where they involve a massive data movement between the processors and off-chip DRAM. The NMP architectures facilitate these data movement through short-length and fast TSV connections between memory and logic

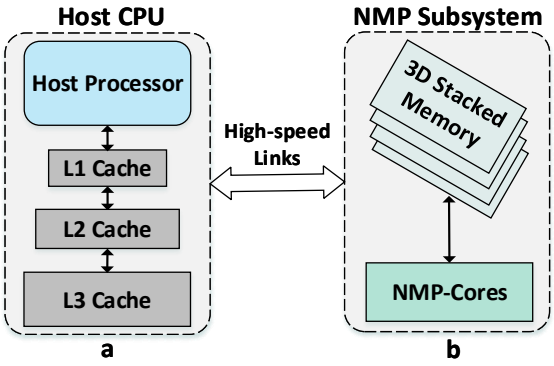

Fig. 1: The overall architecture of a system with NMP capability. An application can run on the Host CPU as in the conventional manner, or it can be offloaded to the NMP subsystem in which data can be accessed more efficiently.

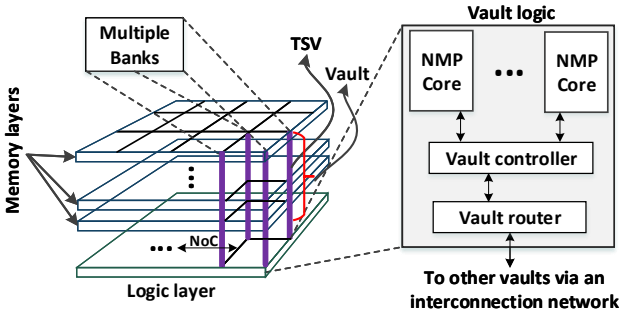

Fig. 2: Conceptual view of a NMP architecture. Each processing unit in the logic layer can utilize high-bandwidth, low-latency and low-power TSV connection to access data in memory with higher internal bandwidth.

layers. This leads to significant power saving and latency improvement.

2.2 Processing In Memory

The cost of moving data in an application continue to increase significantly as applications process larger amount of data. *Processing In Memory (PIM)* provides an opportunity to eliminate unnecessary data movement by bringing part of the computation into the memory. Two decades ago (late 1990s and early 2000s), several research studies investigated the integration of processing logic, which ranges from simple cores to accelerators and FPGAs, and DRAM (or embedded DRAM) modules on a single chip [14] [15] [16] [17] [18] [19] [20]. In this architecture, a host processor was connected to the PIM chip with a custom interconnect. Although it was reported that there was potential for a significant speedup in some classes of applications (e.g, image processing, machine learning, and graph processing). There was a limited success on the past PIM projects and the major reason comes from additional cost (integrating logic and DRAM module) and density shortcoming of 2D chips.

2.3 Near Memory Processing based on 3D Stacking

The most recent and promising innovation that can provide continued scaling of performance is the ability to stack

multiple memory layers on a multi-core processor die. In 3D-stacked memory (e.g., HMC and HBM), a logic layer and multiple memory layers are stacked vertically on top of each other using short and high bandwidth TSVs. TSV-based interconnection provides a low latency and energy efficient data transfer between logic layer and memory layers. Currently, this memory technology provides an opportunity to architects to embed a wide range of computational logic in the logic layer considering the area, energy, and thermal dissipation constraints. These benefits can potentially improve system performance and energy efficiency in a practical manner, but only with careful design of NMP architectures.

It is reported that the 3D-stacked package can communicate with a maximum bandwidth up to 320GB/s with internal memory layers through TSVs and external units through high bandwidth links [3]. Unfortunately, today's processors are not capable of taking full advantage of the improvements offered by the 3D memory technology. NMP systems enabled by 3D-stacking can address one of the major reasons for the limited success of previous PIM projects. This technique avoids additional cost of integrating processing cores with DRAM on the same chip.

NMP systems are the biggest opportunity for emerging data-intensive applications. Such applications scan through massive datasets with a very low temporal locality. As a result, they cannot benefit from large and multi-level cache hierarchies and thus waste memory bandwidth and energy.

There are several research work on integration of the computation unit to the logic layer of 3D-stacked DRAM. Zhang et al. [23] proposed to integrate programmable GPUs to the logic die of 3D-DRAM to offer high throughput. Pugslet et al. [24] created a near data computing architecture for MapReduce workloads. In this work, a host processor is connected to many daisy-chained 3D-stacked DRAM devices with energy-efficient processor cores in their logic layer. Gao et al. [25] proposed a practical near-data processing architecture for in-memory analytics frameworks where a high-end host processor with out-of-order cores is attached to multiple 3D-stacked memory devices (e.g., HMC). In this work, near-data processing cores are responsible for executing the portions of applications with a very low temporal locality, and host processor is responsible for executing the portions of applications with a significant temporal locality. However, to the best of the our knowledge, this work is the first to study a 3D-stacked NMP architecture based on an emerging non-volatile memory technology (PCM).

3 Characterization Methodology

Several data-intensive applications are selected to evaluate performance and power consumption of the proposed NMP architectures (discussed in Section 4). We choose several multi-threaded applications from different benchmark suites to cover a wide range of computation and memory patterns [26] [27] [28] [29]. Table 1 summarises all the evaluated applications and their description. We used full system gem5-NVMain hybrid simulator [33] to evaluate Host CPU performance. Key parameters of the Host CPU system are shown in Table 3.

In this section, we describe the general behaviour of the studied applications and their performance bottleneck.

TABLE 1: Evaluated applications and their description.

Application	Name	Description
Back Propagation	BP	Pattern Recognition
Breadth-First Search	BFS	Graph Analysis
HotSpot 3D	HS-3D	Physics Simulation
Sparse Matrix Vector Multiplication	SpMV	Graph Analysis Machine Learning
HeartWall	HW	Medical Imaging
Stream Cluster	SC	Data Mining
Kmeans Clustering	Kmeans	Artificial Intelligence
Ray Tracing	C-ray	Computer Graphics
Image Rotation	Rotate	Image Processing
Stencil	Stencil	Physics Simulation Machine Learning

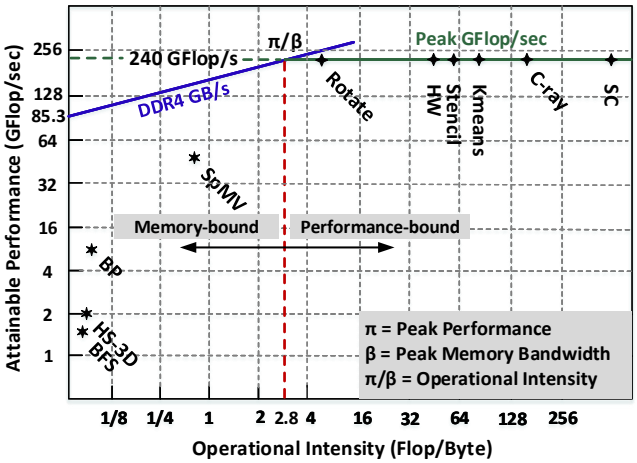

Fig. 3: Constructed Roofline model for our 8-core ALPHA processor with 2 GHz frequency, peak floating point performance of 240 GFlops/sec and peak memory bandwidth of 85.3 GB/s (theoretical). For each application, Roofline data point is shown on the graph based on its operational intensity and attainable performance. The minimum operational intensity to get the maximum performance is π/β = 2.81 Flops/Byte.

As a case study, we conduct a thorough characterization (Roofline analysis, temporal and spatial data locality analysis, and memory access behavior) to illustrate the unique behaviour (memory requirement and access behaviour) of the studied applications and to justify the use of NMP architecture.

3.1 Roofline Analysis

By applying the *Roofline model* which is a throughput oriented performance model, we can find if an application lies in the *memory-bandwidth bound* region or *performance bound* region of the underlying hardware [30]. A Roofline model is constructed which describes the theoretical limits of the modeled Host CPU system described in Section 4.1. Fig. 3 presents the constructed Roofline model along with the Roofline data points for the evaluated applications. These data points represent the operational intensity (Flops per Byte ratio) of each application. In this model, Host CPU system has the theoretical performance limit of 240 GFlops/sec and peak memory bandwidth of 85.3 GB/s (21.3 GB/s per channel).

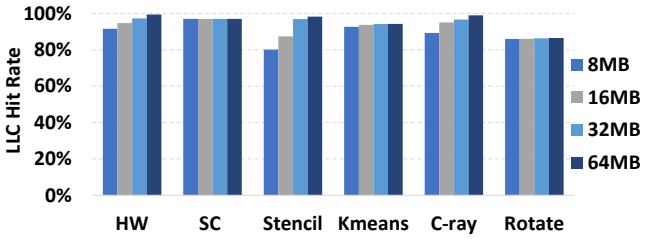

Fig. 4: Temporal data locality sweeping LLC capacity 8-64MB with fixed cache-line size of 64B across all compute-bound applications.

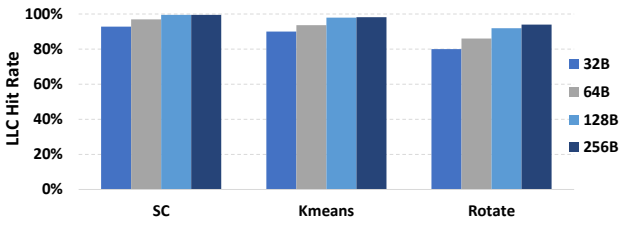

Fig. 5: Spatial data locality sweeping cache-line (LLC) size 32-256B with fixed cache capacity of 16MB across all compute-bound applications with poor/no temporal data locality.

Our Roofline analysis (see Fig. 3) shows that: 1) the attainable performance of Rotate, HW, Stencil, Kmeans, C-ray, and SC applications is approaching the theoretical performance bound of the Host CPU which categories them into applications with high compute bound. These applications have a high computation to communication ratio, 2) BFS, HS-3D, BP, and SpMV applications have a very low compute bound. These applications are bounded by memory bandwidth and they cannot fully utilize the Host CPU processing power.

3.2 Temporal and Spatial Data Locality Analysis

In order to confirm the results obtained by Roofline analysis, we estimate the amount of data locality in compute-bound applications (Rotate, HW, Stencil, Kmeans, C-ray, and SC). Applications with high Flops/Byte and high data locality can exploit benefits offered by Host processing power and large cache hierarchies.

Temporal data locality is the measure of how likely a data is to appear again in a sequence of requests after being requested within a time span. One way to estimate the temporal data locality of an application is to analyze how the cache hit rate of a processor changes as we increase the last-level cache (LLC) capacity with a fixed cache-line size. Fig. 4 shows temporal data locality sweeping cache size from 8MB to 64MB with fixed cache-line size of 64B for applications with high compute bound (see Fig. 3). *We observe that HW, Stencil, and C-ray have enough temporal data locality to leverage from cache hierarchies in Host CPU system. Kmeans and Rotate exhibit a very small improvement in their cache hit rate which implies a very poor temporal locality. SC application with no improvement in the cache hit rate shows no temporal locality.*

TABLE 2: List of all evaluated applications and their memory access behavior. The reported numbers are measured from Host CPU execution. In this table, "Mid" stands for Middle. Applications with "High" and "Mid" memory intensity are classified into memory-intensive (highlighted in gray in the table) and other applications are labeled as memory-non-intensive.

Application	Memory Access Behavior		
	Memory intensity	RBL	R-to-W ratio
BP	21.5 (High)	24.51% (Low)	1.63
BFS	2.4 (Mid)	76.5% (High)	3.93
HS-3D	2.3 (Mid)	75.61% (High)	4.26
SpMV	2.1 (Mid)	16.24% (Low)	4.48
Rotate	0.4 (Low)	28.80%	1.72
HW	0.13 (Low)	5.94%	1.08
Kmeans	0.085 (Low)	75.53%	2.17
C-ray	0.07 (Low)	21.10%	1.41
Stencil	0.03 (Low)	58.91%	1.63
SC	0.006 (Low)	58.47%	3.47

Spatial data locality is the phenomenon that if a program references a particular data, then it is extremely likely that the program will also reference other data that are nearby to the referenced data. This data locality determines sensitivity to the cache-line size and can be estimated by sweeping the cache-line size with a fixed cache capacity [31]. Fig. 5 illustrates spatial data locality for three compute-bound applications with poor/no temporal locality (see Fig. 4) by sweeping cache-line size from 32B to 256B with a fixed cache size of 16MB. *We conclude that all of three compute bound applications with poor/no temporal locality (SC, Kmeans, and Rotate) have enough spatial locality. These applications can utilize the benefits of large cache hierarchies provided by the Host CPU system.*

3.3 Memory Access Behavior

Memory intensity, row buffer locality (RBL), and read-to-write ratio (R-to-W) are three components which we use to estimate the memory access behavior of each application. Table 2 lists all applications used in our study (both memory-intensive and compute-intensive) and their memory characteristics. The focus of this section is on applications with high and middle memory intensity (memory intensity > 2).

Memory intensity is the frequency at which a request misses in the last-level cache which is determined in the unit of misses per kilo instructions (MPKI) of LLC. Applications with memory intensity greater than two (LLC MPKI > 2) are classified into memory-intensive and other applications (which have LLC MPKI < 2) are labeled as non-memory intensive (compute-intensive). *Data shown in Table 2 confirms the results obtained by Roofline analysis and data locality analysis. Rotate, HW, Kmeans, C-ray, Stencil, and SC have a very low memory intensity which lies them in the non-memory-intensive category. BP, BFS, HS-3D, and SpMV applications (highlighted in gray in Table 2) have memory intensity greater than two which categorize them into memory-intensive class.*

Memory device organization includes a peripheral storage known as *row buffer (RB)* which acts as a cache for memory array rows and is independent from the memory technology. This memory component is present in both

TABLE 3: The key parameters of the simulated systems

Host CPU System	
Processor	8 cores @ 2 GHz
Caches	per-core L1 (I): 32 KB, 2-way per-core L1 (D): 32 KB, 2-way per-core L2: 256 KB, 4-way shared L3: 16 MB, 8-way cache-line size: 64 B
DRAM Memory	
DDR4-2666 MHz	16 GB: 16 Gb $\times 8$ 4 channels $\times$ 4 ranks $\times$ 4 banks Row buffer size: 8 KB Bandwidth: 21.3 GB/s per channel (theoretical) 15 GB/s per channel (empirical)
Timing Parameters	$t_{ck} = 1.25ns$ $t_{RAS} = 42$, $t_{RCD} = 19$, $t_{CAS} = 10$ $t_{CCD} = 4$, $t_{RP} = 19$, $t_{WR} = 210$
NMP System	
Cores	8 cores @ 1.8 GHz
Caches	per-core L1 (I): 32 KB, 4-way per-core L1 (D): 32 KB, 4-way cache-line size: 64 B
3D-stacked Memory	
3D-DRAM (HMC)	16 GB: 2 layers $\times$ 4 vaults $\times$ 4 rank 4 ranks/vault 2 cores per vault logic Row buffer size: 256 B
3D-PCM	16 GB: 2 layers $\times$ 4 vaults $\times$ 4 rank 4 ranks/vault 2 cores per vault logic Row buffer size: 256 B

DRAM and PCM. When content of a memory array's row is placed in the row buffer, successive memory requests to the same row are served immediately from the row buffer. These memory accesses are called row buffer hits. If a memory request refers to a row which is different from the latched one in the row buffer, then this request causes row buffer miss. Examining this metric, a same style buffering and size is assumed for row buffer in both DRAM and PCM technologies. *The RBL of an application is the average hit rate of the row buffer across all memory channels. Based on Table 2, BFS and HS-3D have high RBL (RB hit rate $>$ 75%). BP and SpMV with RB hit rate less than 25% are considered as applications with low RBL.*

Based on different characteristics of DRAM and PCM such as read energy, write energy, and power consumption, *R-to-W* ratio metric is used to provide a detailed information of each application. Since PCM suffers from a high write energy/power, this metric is used to find the dominant operation (read or write) in memory-intensive applications.

4 Evaluation Methodology

In this section, we discuss the evaluation methodology and simulation configurations for three different processing units (conventional Host CPU, 3D-DRAM based NMP, and 3D-PCM based NMP).

Our study of identifying the potential of NMP system to boost the performance and power consumption of the memory-intensive applications (discussed in Section 3) is based on matching the characteristics of these applications to NMP systems (3D-DRAM NMP and 3D-PCM NMP). We simulated ten real-world applications presented in Section 3 for our evaluation. The application characterization (Section 3) is performed on conventional Host CPU (see Table 3 for Host CPU configuration) to define the unique behavior of each application. Roofline analysis along with data locality (temporal and spatial locality) analysis helped us to classify applications into compute-bound and memory-bound. Then by defining memory access behavior of memory-intensive applications (applications with high and middle memory intensity highlighted in gray in Table 2), we can match their memory characteristics (RBL and R-toW ratio) to the two existing NMP systems.

4.1 Simulation Models

We evaluated the conventional Host CPU system using full system gem5-NVMain hybrid simulator [33]. We use Ramulator-Pim, a processing-in-memory simulation framework to evaluate NMP subsystems [34]. This framework is based on two simulators, ZSim [35] (a fast and accurate simulator for thousand core systems) and Ramulator [36] (a fast and cycle accurate DRAM simulator).

Table 3 summarizes the key parameters and configurations of the simulated systems. The Host CPU system includes a 8-core ALPHA processor with two levels of private caches (L1 and L2) per core and a shared L3 cache. The memory subsystem is modeled using Micron DDR4 timing parameters [37] which includes four DDR4-2666 MHz memory channels with four banks per rank and four ranks per channel. Each memory channel has a theoretical bandwidth of 21.3 GB/s. The application characterization (in Section 3) is conducted on this system.

The NMP systems are based on two different memory technologies: 3D-DRAM and 3D-PCM. Both NMP systems extend the 3D memory systems by introducing a number of simple cores with caches into the logic layer. Table 3 includes more details regarding the simulated NMP systems.

5 Evaluation Results

In this section, we present the experimental results for running memory-intensive applications (BP, BFS, HS-3D, and SpMV) under three different platforms (Host CPU, 3D-DRAM NMP and 3D-PCM NMP). A summary of the experimental setup for the conventional Host CPU and NMP systems is shown in Table 3. Unless otherwise stated, all results are normalized to the Host CPU system.

5.1 Performance Comparison

We use the execution stage average Instruction Per Cycle (IPC) of each memory-intensive application as a performance metric to perform speedup comparison. Fig. 6 shows a performance comparison of memory-intensive applications under three different platforms: Host CPU with the conventional DDR4 memory and NMP systems with processing units embedded in logic layer of 3D-DRAM and 3D-PCM. Along the x-axis, the applications are sorted by memory-intensity (LLC MPKI), from highest to least (see Table 2) and average IPC results are normalized to the Host CPU system. Fig. 6 indicates that in both NMP systems the

average IPC of target applications has improved comparing to the Host CPU. Comparing two NMP systems, 3D-PCM NMP has a very negligible difference with 3D-DRAM NMP in IPC improvement.

Fig. 7 provides further information into the performance comparison using memory access latency. Along the y-axis, represented values are normalized to the Host CPU system. We draw out two findings from this figure:

1) It depicts a significant performance benefits (memory access latency reduction) for BP and SpMV applications when running them on two NMP systems (3D-PCM NMP and 3D-DRAM NMP). As it is shown in Table 2 (discussed in Section 3.3), BP and SpMV exhibit a low RBL on Host CPU with DDR4 memory. Having poor data locality at the memory array level, these applications can benefit from 3D-stacked memories (3D-PCM and 3D-DRAM) which deliver higher bandwidth and memory-level parallelism compared to DDR4 memory. Furthermore, NMP systems eliminate data movement bottleneck which greatly improves the memory access latency for these applications.
2) Though there is an improvement in IPC of BFS and HS-3D applications (see Fig. 6), we observe an increase in memory access latency of these applications when NMP systems are employed (see Fig. 7). To understand the reason, we look at RBL locality of these applications when running them on Host CPU with DDR4 memory. As it is shown in Table 2, BFS and HS-3D have a high RBL (RB hit rate $> 75\%$) which is exploited by DRR4 memory because of its large row buffer size (8KB). Lower memory access latency (see Fig. 7 for BFS and HS-3D) on Host CPU with DDR4 is the result of exploiting high data locality at memory array row. Running these applications on NMP system which is enabled by memory with very small row size (256B) increases the memory access latency, since memory row misses occur more frequently. High internal parallelism (bank-level) offered by 3D-stacked memories (which have processing cores in the logic layer) is exploited by these applications. Thus, due to the high bank-level parallelism, BFS and HS-3D exhibit a significant improvement in their IPC with NMP execution.

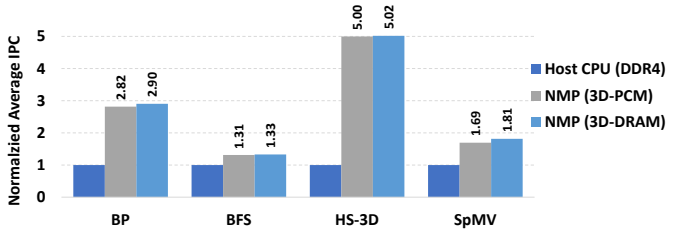

Fig. 6: Performance comparison based on average IPC between Host CPU, 3D-PCM NMP, and 3D-DRAM NMP systems across all memory-intensive applications. IPC results are normalized to the Host CPU system.

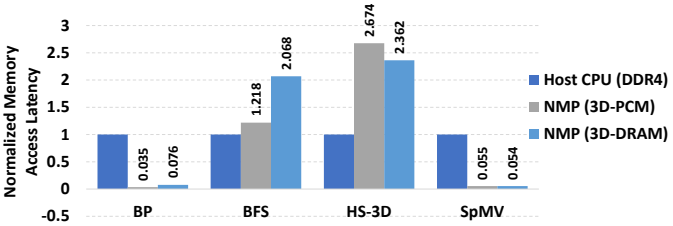

Fig. 7: Memory access latency comparison between Host CPU, 3D-PCM NMP, and 3D-DRAM NMP systems across all memory-intensive applications, normalized to the Host CPU system.

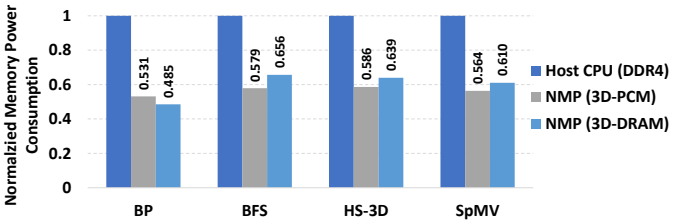

Fig. 8: Memory power consumption for Host CPU and NMP execution cases, normalized to the Host CPU system.

5.2 Memory Power Consumption

Power analysis for DDR4 memory in Host CPU system has been done using gem5-NVmain simulator [33]. We used DRAMPower [41] for evaluating power consumption of memory devices (3D-DRAM and 3D-PCM) in two NMP systems. Fig. 8 shows the memory power consumption of memory-intensive applications for Host CPU system with DDR4 and two NMP systems (3D-PCM and 3D-DRAM). The memory power consumption is normalized to the Host CPU system. The power savings are realized across all memory-intensive applications with NMP execution which are obtained having shallow cache hierarchy in NMP systems that avoids excess memory access latency.

Based on Fig. 8, we observe two interesting findings. First: for BP application that has high LLC MPKI (see Table 2), NMP systems (3D-PCM and 3D-DRAM) outperform the Host CPU execution by an average of 46.86% and 51.48% in memory power saving, respectively. Other applications (BFS, HS-3D, and SpMV) with middle MPKI (see Table 2) also experience power savings with NMP execution, still significant but lower than BP.

Second: while 3D-DRAM NMP outperforms 3D-PCM NMP for BP application, 3D-PCM NMP exhibits more power saving compared to 3D-DRAM NMP for other applications (BFS, HS-3D, and SpMV). This is due to the difference in applications' average R-to-W ratio (Table 2). Considering that PCM technology suffers from a high write energy/power, we can infer that for BFS, HS-3D, and SpMV, read operations and for BP, write operations are the dominant factor in determining the memory power consumption. This explains the reason as to why some memory-intensive applications see more power saving than others.

6 Conclusion

In this paper, we studied two NMP computing devices based on 3D stacking (3D-DRAM and 3D-PCM) to accelerate data-intensive problems caused by *memory wall* bottleneck of the conventional processing architectures. We performed a systematic characterization for a wide range of multi-threaded applications and revealed their performance bottleneck. Overall, our system-level evaluation demonstrates that the evaluated NMP systems (3D-DRAM NMP and 3D-PCM NMP) improve the performance of memory-intensive applications by 1.31x to 5x and reduce their total memory energy/power consumption by an average of 40%.

Acknowledgments

This work has been partially supported by the Center of Pervasive Communications and Computing (CPCC) at University of California, Irvine.

References

[1] Wm. A. Wulf and Sally A. McKee. 1995. *"Hitting the memory wall: implications of the obvious."* SIGARCH Comput. Archit. News 23, 1 (March 1995), 20–24.

[2] Q. Wu, K. Rose, J. Lu and T. Zhang, *"Impacts of though-DRAM vias in 3D processor-DRAM integrated systems,"* 2009 IEEE International Conference on 3D System Integration, San Francisco, CA, pp. 1-6.

[3] Hybrid Memory Cube Consortium, *"HMC Specification 2.0"*, 2014.

[4] J. Jeddeloh and B. Keeth, *"Hybrid memory cube new DRAM architecture increases density and performance,"* 2012 Symposium on VLSI Technology (VLSIT), Honolulu, HI, 2012, pp. 87-88.

[5] JEDEC Solid State Technology Association, *"JESD235B: High Bandwidth Memory (HBM) DRAM Standard"*, Nov. 2018.

[6] Eghbal, Ashkan and Yaghini, Pooria M. and Bagherzadeh, Nader. *"Capacitive Coupling Mitigation for TSV-based 3D ICs"* 2015 IEEE 33rd VLSI Test Symposium (VTS), doi=10.1109/VTS.2015.7116279.

[7] B. C. Lee et al., *"Phase-Change Technology and the Future of Main Memory,"* in IEEE Micro, vol. 30, no. 1, pp. 143-143, Jan.-Feb. 2010.

[8] E. Kültürsay, M. Kandemir, A. Sivasubramaniam and O. Mutlu, *"Evaluating STT-RAM as an energy-efficient main memory alternative,"* 2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Austin, TX, 2013, pp. 256-267.

[9] H. P. Wong et al., *"Metal–Oxide RRAM,"* in Proceedings of the IEEE, vol. 100, no. 6, pp. 1951-1970, June 2012.

[10] Chen, A. *"A review of emerging non-volatile memory (NVM) technologies and applications."* Solid-state Electronics 125 (2016): 25-38.

[11] Onur Mutlu. *"Opportunities and challenges of emerging memory technologies,"* ARM Research Summit, 2017.

[12] Maryam SeyyedHosseini. Master thesis, 2017. *"Reliability Enhancement of Many-core Processors."* ProQuest Dissertations and Theses. https://www.proquest.com/dissertations-theses/reliability-enhancement-many-core-processors/docview/1885091904/se-2?accountid=14509.

[13] Pooria M. Yaghini. PhD thesis, 2016. *"Resilient 3D Network-on-Chip Design and Analysis."* ProQuest Dissertations and Theses. https://www.proquest.com/dissertations-theses/resilient-3d-network-on-chip-design-analysis/docview/1808248453/se-2?accountid=14509.

[14] M. Gokhale, B. Holmes and K. Iobst. *"Processing in memory: the Terasys massively parallel PIM array."* Computer, vol 28, number 4, pages 23-31, 1995.

[15] M. Hall et al., *"Mapping Irregular Applications to DIVA, a PIM-based Data-Intensive Architecture."* SC '99: Proceedings of the 1999 ACM/IEEE Conference on Super computing, Portland, OR, USA, 1999, pp. 57-57.

[16] Yi Kang et al., *"FlexRAM: toward an advanced intelligent memory system,"* Proceedings 1999 IEEE International Conference on Computer Design: VLSI in Computers and Processors (Cat. No.99CB37040), Austin, TX, USA, 1999, pp. 192-201.

[17] Peter M. Kogge. *"EXECUBE-A New Architecture for Scaleable MPPs."* In Proceedings of the 1994 International Conference on Parallel Processing, Volume 01 (ICPP '94). IEEE Computer Society, USA, 77–84.

[18] D. Patterson et al., *"A case for intelligent RAM,"* in IEEE Micro, vol. 17, no. 2, pp. 34-44, March-April 1997.

[19] R. Balasubramonian et al., *"Near-Data Processing: Insights from a MICRO-46 Workshop,"* in IEEE Micro, vol. 34, no. 4, pp. 36-42, July-Aug. 2014.

[20] A. Srivastava, et al., *"Mapping Irregular Applications to DIVA, a PIM-based Data-Intensive Architecture,"* in SC Conference, Portland, Oregon, USA, 1999 pp. 57.

[21] Donghyuk Lee, Saugata Ghose, Gennady Pekhimenko, Samira Khan, and Onur Mutlu. *"Simultaneous Multi-Layer Access: Improving 3D-Stacked Memory Bandwidth at Low Cost."* ACM Trans. Archit. Code Optim. 12, 4, Article 63 (January 2016), 29 pages.

[22] W. R. Davis et al., *"Demystifying 3D ICs: the pros and cons of going vertical,"* in IEEE Design & Test of Computers, vol. 22, no. 6, pp. 498-510, Nov.-Dec. 2005.

[23] Dongping Zhang, Nuwan Jayasena, Alexander Lyashevsky, Joseph L. Greathouse, Lifan Xu, and Michael Ignatowski. *"TOP-PIM: throughput-oriented programmable processing in memory,"* In Proceedings of the 23rd international symposium on HPDC, 2014. ACM, New York, NY, USA, 85–98.

[24] S. H. Pugsley et al., *"NDC: Analyzing the impact of 3D-stacked memory+logic devices on MapReduce workloads,"* 2014 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Monterey, CA, 2014, pp. 190-200.

[25] M. Gao, G. Ayers and C. Kozyrakis, *"Practical Near-Data Processing for In-Memory Analytics Frameworks,"* 2015 International Conference on Parallel Architecture and Compilation (PACT), San Francisco, CA, 2015, pp. 113-124.

[26] S. Che and M. Boyer and J. Meng and D. Tarjan and J. W. Sheaffer and S. Lee and K. Skadron. *"Rodinia: A benchmark suite for heterogeneous computing"*. IISWC, 2009.

[27] Bienia, Christian and Kumar, Sanjeev and Singh, Jaswinder Pal and Li, Kai. *"The PARSEC Benchmark Suite: Characterization and Architectural Implications"*, PACT 2008, 72-81.

[28] *"Parboil benchmark suite:"* http://impact.crhc.illinois.edu/parboil/parboil.aspx.

[29] *"Starbench benchmark suite:"* https://www.aes.tu-berlin.de/menue/forschung/projekte/abgeschlossene_projekte/starbench_parallel_benchmark_suite/.

[30] Samuel Williams, Andrew Waterman, and David Patterson. 2009. *"Roofline: an insightful visual performance model for multicore architectures."* Commun. ACM 52, 4 (April 2009), 65–76.

[31] Steven Cameron Woo, Moriyoshi Ohara, Evan Torrie, Jaswinder Pal Singh, and Anoop Gupta. 1995. *"The SPLASH-2 programs: characterization and methodological considerations,"* SIGARCH Comput. Archit. News 23, 2 (May 1995), 24–36.

[32] L. Ke et al., "RecNMP: Accelerating Personalized Recommendation with Near-Memory Processing," 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA), Valencia, Spain, 2020, pp. 790-803.

[33] Cyjseagull. Apr. 2016 *"Gem5-NVMain,"* https://github.com/cyjseagull/gem5-nvmain-hybrid-simulator.

[34] SAFARI Research Group. *"Ramulator for Processing in Memory"*.https://github.com/CMU-SAFARI/ramulator-pim.

[35] Sanchez, Daniel and Kozyrakis, Christos,*"ZSim: fast and accurate microarchitectural simulation of thousand-core systems"*,*ACM SIGARCH Computer Architecture News*, vol. 41, p. 475, 2013.

[36] Yoongu Kim, Weikun Yang, and Onur Mutlu. *"Ramulator: A Fast and Extensible DRAM Simulator.* IEEE Comput. Archit". Lett. 15, 1, 45–49, January 2016.

[37] *"Micron DDR4 SDRAM, 16GB:"* https://www.micron.com/-/media/client/global/documents/products/data-sheet/dram/ddr4/16gb_ddr4_sdram.pdf.

[38] Jalil Boukhobza, Stéphane Rubini, Renhai Chen, and Zili Shao. *"Emerging NVM: A Survey on Architectural Integration and Research Challenges."* ACM Trans. Des. Autom. Electron. Syst. 23, 2, Article 14 (January 2018), 32 pages.

[39] W. Zhang and T. Li, *"Exploring Phase Change Memory and 3D Die-Stacking for Power/Thermal Friendly, Fast and Durable Memory Architectures,"* 18th International Conference on Parallel Architectures and Compilation Techniques, Raleigh, NC, USA, 2009, pp. 101-112.

[40] S. Yu and P. Chen, *"Emerging Memory Technologies: Recent Trends and Prospects,"* in IEEE Solid-State Circuits Magazine, vol. 8, no. 2, pp. 43-56, Spring 2016.

[41] *"DRAMPower: Open-source DRAM Power & Energy Estimation Tool"* [Online]. Available: http://www.drampower.info.

Maryam S. Hosseini (Student Member, IEEE) received her Ph.D. degree in Electrical and Computer Engineering from University of California, Irvine in July 2021. She received Ph.D. bridge fellowship from the EECS department in 2016 while pursuing her master degree. Maryam got the master degree in Computer Systems and Software from University of California, Irvine in 2017. Her research interests include computer architecture, near memory processing, high performance computing, and hybrid memory systems. She is a member of the IEEE and ACM.

Masoumeh (Azin) Ebrahimi (Senior Member, IEEE) received the Ph.D. degree (with Honors) from the University of Turku, Finland, in 2013, and the joint M.B.A. degree from the University of Turku and the EIT-ICT School in 2015. She is currently an Associate Professor with KTH Royal Institute of Technology, Sweden, and an Adjunct Professor with the University of Turku, Finland. She has led several national and international projects, such as EU-MarieCurie-Vinnova, Academy of Finland, SSF, STINT, and Vetenskapsrådet (VR). Her scientific work contains more than 100 publications, and her main areas of interests include interconnection networks, near-memory processing, and neural network accelerators. She is a member of HiPEAC. She actively acts as a Guest Editor, an Organizer, and the Program Chair in different venues and conferences.

Pooria Yaghini (Member, IEEE) received his Ph.D. degree in Computer System and Software from University of California, Irvine in 2017, where has also served as a lecturer. He received Henry Samueli Endowed Fellowship in 2011. Pooria got the master degree from Amirkabir University of Technology in 2010. His research interests include processor architectures, cache and memory systems, 3D-NoC, reliability analysis and verification, and fault-tolerant architecture design and energy-aware computing. He is also the author of featured paper of IEEE Transaction on Computers published in December 2015. He is a member of the IEEE and ACM.

Nader Bagherzadeh (Fellow, IEEE) is a professor of computer engineering in the department of electrical engineering and computer science at the University of California, Irvine, where he served as a chair from 1998 to 2003. He received a Ph.D. degree from the University of Texas at Austin in 1987. He is a Fellow of the IEEE. He is also a member of the Center for Pervasive Communications and Computing (CPCC). Professor Bagherzadeh has published more than 400 articles in peer-reviewed journals and conferences. He has been involved in research and development in the areas of: computer architecture, re-configurable computing, VLSI chip design, Network on-Chip, 3D chips, computer graphics, machine learning accelerators, memory and embedded systems.