Application Characterization for Near Memory Processing

Maryam S. Hosseini*, Masoumeh Ebrahimi†, Pooria Yaghini*, and Nader Bagherzadeh*

*Center for Pervasive Communications and Computing (CPCC)
*Department of Electrical Engineering and Computer Science, University of California, Irvine
*{mseyyedh, pooriam, nader}@uci.edu
†Electronics and Embedded Systems Department, KTH Royal Institute of Technology, Sweden
†mebr@kth.se

***Abstract*—Data movement between memory subsystem and processor unit is a crippling performance and energy bottleneck for data-intensive applications. Near Memory Processing (NMP) is a promising solution to alleviate the data movement bottleneck. The introduction of 3D-stacked memories and more importantly hybrid memory systems enable the long-wished NMP capability. This work explores the feasibility and efficacy of having NMP on the hybrid memory system for a given set of applications. In this paper, we first redefine a set of NMP-centric performance metrics in order to analyze the efficacy of a given processing unit. Leveraging the proposed metrics, we characterize various sets of applications to assess the suitability of a processing unit in terms of performance. Specifically, in this work we motivate the efficiency of NMP subsystems to process memory-intensive applications when 3D-NVM technologies are employed.**

***Index Terms*—Data-intensive applications, Near memory processing, 3D-stacked memory technology.**

I. Introduction

Over the years, memory technology has not been able to keep up with the improvements in processor technology in terms of latency and energy consumption, which is referred as *memory wall* [1]. The memory hierarchy is proposed to mitigate some disadvantages of the off-chip DRAM. However, the limitations in the number of pins in the memory package results in high bandwidth demands. This has become a real challenge for today's multicore processors. Also, the end of Dennard scaling and Moore's law are causing the computer performance to reach a plateau [2]. At the same time, we are witnessing another challenge in today's conventional computing systems. In the era of big data, an enormous amount of data is being generated across multiple areas such as health sciences, chemistry, physics, and IoT. Processing this huge amount of data results in frequent data movement between the memory subsystem and the processor unit, which has a severe impact on performance and energy efficiency of the whole system. So, the best way to eliminate or at least reduce the data movement overhead is to avoid it as much as possible.

With the advent of 3D-stacked memories, such as Micron's Hybrid Memory Cube (HMC) [3] and JEDEC's High Bandwidth Memory (HBM) [4], the concept of Near Memory Processing (NMP) is revolutionized. A 3D-stacked memory consists of multiple memory dies stacked vertically on top of a logic layer using short and high bandwidth Through-Silicon Vias (TSVs). The logic layer can embed the processing elements. In particular, 3D-stacking technology is endorsed as the true enabler of processing near to memory (data). Furthermore, this technology can be applied to both volatile and emerging non-volatile memory technologies, which makes it more practical and beneficial to exploit the advancements of emerging memory technologies. 3D-stacked memory system provides significantly more internal and external bandwidth than conventional DDR modules. It is reported that a single HMC offers an internal bandwidth of 160-320GB/s which is more than 10x the bandwidth offered by a single DDR3 module, while providing a high-level parallelism [3]. However, general-purpose processors have limited capabilities to take full advantage of resources offered by 3D memory modules.

In contrast to conventional memory technologies, cutting edge Non-Volatile Memory (NVM) technologies such as Phase Change Random Access Memory (PCM) and Spin-Torque Transfer Random Access Memory (STTRAM) are attracting huge attention as the promising candidates for the next-generation memory systems. Among all the emerging NVMs, PCM is considered as the most mature one with proven performance which can benefit from more reduction in the switching power [5] [6]. As process technology node shrinks, PCM becomes faster and more energy efficient compared to DRAM technology which is facing scaling challenges.

In this paper, we evaluate the suitability of different processing units (host CPU, near 3D-DRAM and near 3D-PCM processing) for various sets of applications. The contributions of this work are summarized as follows:

- Redefining a set of performance metrics with the focus of application characterization on host CPU and NMP subsystems. Specifically, in this work, we motivate the efficiency of NMP subsystems to process memory-intensive applications when PCM, as an emerging non-volatile memory technology, is employed.
- Characterizing various sets of applications (selected from several categories and domains), from Rodinia [7], Parboil [8], and Parsec [9] benchmark suites (summarized in

TABLE I), exploring the proposed metrics.

- Analyzing and suggesting the suitability of different processing units (host CPU processing and NMP) for the set of characterized applications.

II. Subsystem Selection and Analysis

This section first describes the simulation configuration of the modeled host CPU processing unit. Further, a comprehensive characterization is conducted on the various applications sets. The characterization (temporal locality, OI (Roofline analysis), R-to-W ratio, and RBL) illustrates the key behaviour of the applications and helps to suggest the most suitable processing unit for each application.

A. Simulation Configuration

We evaluate our conventional base system using Gem5-NVMain hybrid simulator [10]. The simulator integrates the full system simulator, Gem5 [11], with NVMain 2.0 [12] which is a main memory simulator for emerging NVM technologies. We model the base host processor with 16 ALPHA cores running at 1 GHz frequency. DRAM memory is modeled with DDR4-2400 memory configuration. The architectural details for the host CPU are summarized in TABLE II. For the NMP subsystems, we assume two different memory technologies (3D-DRAM and 3D-PCM). For the evaluation, we consider applications from different benchmark suites (shown in TABLE I) to cover a wide range of application domain.

B. Application Characterization

*1) **Host CPU versus NMP**:* We use *temporal locality* and *operational intensity* metrics to evaluate the amenability of various applications to host CPU processing and near memory processing. Temporal locality metric helps to categorize applications into: (a) applications with poor or no temporal locality which cannot benefit from host caches and (b) applications with high temporal locality, which are more suitable for host processing. One way to estimate the temporal locality of an application is to analyze how the second-level (L2) cache miss rate of a processor changes as we increase the cache capacity [13]. We analyze the L2 cache miss rate over different cache sizes (16MB to 128MB) to estimate temporal locality of an application. We examine the results for 8-way set associative cache with 64-byte line size and various cache sizes.

Temporal Locality: Fig. 1 shows temporal data locality sweeping shared L2 cache (all power of two shared L2 cache sizes from 16MB to 128MB) with fixed cache-line size of 64B. Based on results shown in Fig. 1, applications such as BP and MO have a very large cache miss rates, and increasing the cache size does not improve the miss rate. Based on the results, we can infer that irregular memory access patterns (limited or no temporal locality) and/or large working sets (which cannot fit even in large caches) with a large reuse distance (larger than the cache size) cause excessive cache misses for such applications. So, the major performance bottleneck comes from moving data between the processor and off-chip memory. These applications cannot benefit from large cache hierarchies, and it would be more beneficial (in terms of access latency) to process them in the NMP subsystem. Applications such as HS, BFS, SC, Fr, SpMV, BS, BT, X264, and SW can be categorized into limited or no temporal locality group since there is no or a very small change in their miss rate when the cache size increases. The important working sets of these applications can fit in realistic size of L2 (1MB per core), but they suffer from irregular memory access patterns which limits their temporal locality. Thereby, such applications cannot exploit host caches, and it is more efficient to process them in the NMP subsystem.

TABLE I
List of Evaluated Applications

Application	Acronym	Domain
Back Propagation	BP	Pattern Recognition
Myocyte	MO	Biological Simulation
HotSpot	HS	Physics Simulation
Breath-First Search	BFS	Graph Algorithms
Heart Wall	HW	Medical Imaging
Sparse Matrix Vector Mult.	SpMV	Graph Analytics, ML
Black Scholes	BS	Linear Algebra
Body Track	BT	Computer Vision
Ferret	Fr	Similarity Search
Stream Cluster	SC	Data Mining
VASARI Image Processing Sys.	VIPS	Media Processing
X264	X264	Media Processing
Swaptions	Sw	Financial Analysis

TABLE II
System Parameters and Configurations

Host CPU System	
Processor	16 ALPHA cores, 1 GHz frequency
Caches	line-size: 64 B per-core L1 (I): 32 KB, 2-way set associative per-core L1 (D): 32 KB, 2-way set associative shared L2: 32 MB, 8-way set associative
DRAM Memory (DDR4-2400MHz)	
Organization	32 GB, 4 channels × 4 rank, 2 Gb, x8
Bandwidth	62.1 GB/s (empirical)

HW and VIPS are applications with considerable change in their miss rate when the cache size increases. This change explains memory access pattern and working set size for such applications. High temporal locality, large working sets (larger than the realistic size for L2), and large reuse distances decrease the L2 miss rate when there is enough space in L2 cache to hold the important working sets of these applications.

Operational Intensity (OI): Based on this metric, we can characterize applications into CPU-bound and memory-bound. Applying the Roofline model [14], we relate the performance and OI of an application to the peak performance of the underlying processor and memory bandwidth. Roofline model shows the peak performance, peak memory bandwidth, and operational intensity all together in a 2-dimensional graph. Fig. 2 shows the constructed Roofline model for our 1 GHz 16-core ALPHA processor in a single-socket system with the empirical DRAM bandwidth of 62.1 GB/s (1.79 GB/s in $\log_{10}$ scale). The y-axis represents the attainable performance and the x-axis shows the OI (both in $\log_{10}$ scale).

Based on the Roofline graph, the modeled host system has a peak performance of 2.4 GFlops/s (in $\log_{10}$ scale) and peak

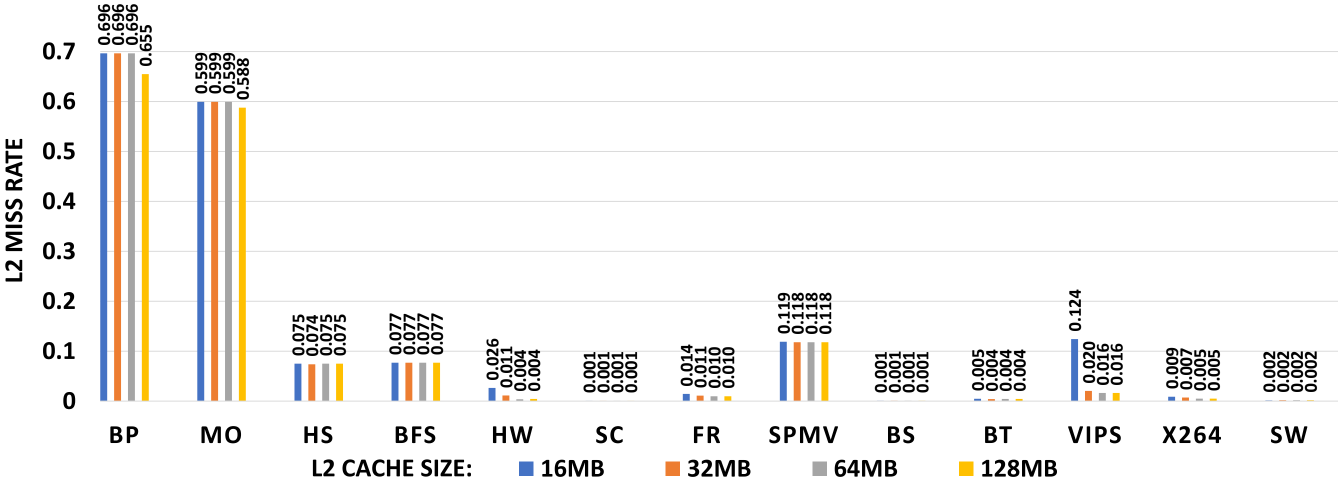

Fig. 1. Temporal data locality sweeping L2 cache size (16MB-128MB). L2 miss rate data assumes 16 processors with 64B cache-line size.

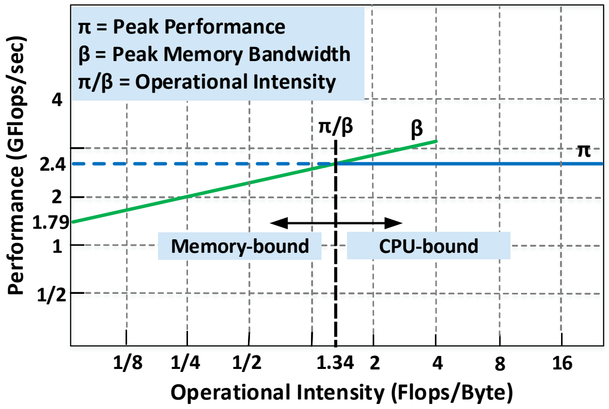

Fig. 2. Roofline model for our 16-core ALPHA processor with 1 GHz frequency, peak floating point performance of 2.4 GFlops/sec (in $\log_{10}$ scale) and peak memory bandwidth of 1.79 GB/s (in $\log_{10}$ scale). The minimum OI to get the maximum performance is π/β = 1.34 Flops/byte.

memory bandwidth of 1.79 GB/s (in $\log_{10}$ scale). For a given application, a point can be found on the x-axis based on its OI. The minimum OI to get the maximum performance is π/β which is 1.34 Flops/byte for our host processor. As it is illustrated in Fig. 2, applications with OI greater than or equal to 1.34 Flops/byte are CPU-bound, and applications with OI less than 1.34 are memory-bound.

Fig. 3 shows the OI for all the evaluated applications. Application such as BFS, HS, X264, MO, BP, and SpMV are characterized by extremely low OI (the byte ratio is less than 1) which makes them inherently memory-bound based on the Roofline model. Fr is also considered as a memory-bound application with OI greater than 1.0. For such applications, data movement is the major performance bottleneck which causes excessive cache misses. VIPS, HW, BT, BS, SC, and Sw have OI greater than 1.34. These applications are categorized into CPU-bound that can fully utilize the host processing power and cache hierarchies (in case of having high temporal locality).

Based on temporal locality and OI metrics, we can conclude that applications with high OI and high temporal locality (HW and VIPS) are CPU-bound which can benefit from host processing power and large cache hierarchies. Applications with low temporal locality and low OI (BP, MO, Fr, SpMV, HS, BFS, and X264) are memory-bound that can benefit from NMP systems.

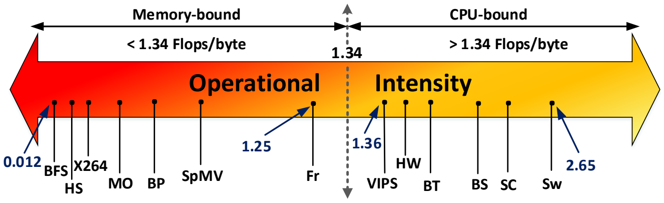

Fig. 3. Application categorization based on operational intensity and Roofline model of the simulated processor analyzed in this work. As it is depicted, applications with operational intensity less (more) than 1.34 (in $\log_{10}$ scale) are categorized as memory-bound (CPU-bound).

Applications such as SC, BT, BS, and SW show a mixed behavior of no/low temporal locality and high OI. For this set of applications, we use memory bandwidth utilization and L2 Miss Per Kilo Instructions (L2 MPKI) metrics to determine their memory intensity. L2 MPKI determines memory intensity which denotes the number of DRAM requests per kilo instructions. L2 MPKI can be defined as:

$$\frac{Number\ of\ miss\ memory\ accesses}{(Total\ number\ of\ committed\ instructions\ /\ 1000)} \tag{1}$$

Fig. 4 shows memory bandwidth utilization (in GB/s) and L2 MPKI for applications with no temporal locality and high OI. A very low memory bandwidth utilization (less than 0.02 GB/s) and a very low L2 MPKI (less than 0.2) determine that such applications have a negligible memory footprint which categorize them into CPU-bound (CPU-friendly) that can be processed efficiently on host CPU.

*2) **Near 3D-DRAM versus Near 3D-PCM Processing**:* After categorizing applications into CPU-bound and memory-bound, we use Read-to-Write ratio (R-to-W) and Row Buffer Locality (RBL) [15] as NMP-centric metrics to evaluate the

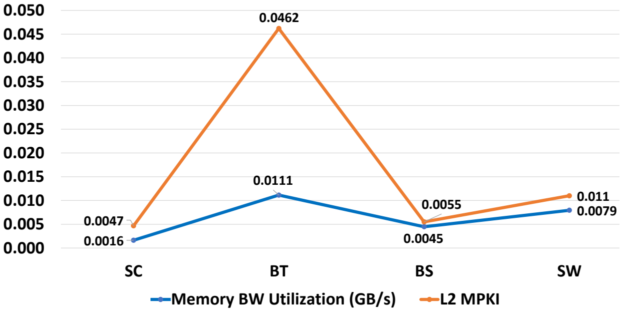

Fig. 4. Memory bandwidth (BW) utilization and L2 MPKI for mixed class of instructions (instructions with no/low temporal locality and high OI). The less DRAM accesses and memory BW utilization an application has, the more CPU-friendly the application is.

amenability of memory-intensive applications to near 3D-DRAM and near 3D-PCM processing. TABLE III shows the device level properties of PCM and DRAM technologies. PCM as an emerging memory technology suffers from a very high write latency comparing to DRAM technology. It provides desirable properties such as satisfactory read latency (comparable to DRAM), zero standby power, no refresh power, superior scalability, CMOS process compatibility, higher memory capacity for the same chip area, 3D die-stacking capability, and more benefit from TSVs in terms of power saving in the 3D structure [15] [16]. Because of these attractive properties, memory-read-intensive applications can be efficiently processed close to 3D-PCM.

R-to-W: The right Y-axis in Fig. 5 shows the average R-to-W ratio for all memory-intensive applications. A very high R-to-W ratio indicates that read bandwidth has the major impact on the system performance. So, for memory-bound applications with a very high R-to-W ratio (memory-read-intensive) near 3D-PCM processing would potentially outperform near 3D-DRAM processing because of power saving in the 3D-structure and a very low standby power offered by PCM technology [15] [16]. Based on multiple research work, PCM write latency varies from 150ns to 500ns (shown in TABLE III). We consider average R-to-W ratio higher than 3 (PCM to DRAM write latency ratio) to characterize memory-bound applications into read-intensive and write-intensive. This approach helps to differentiate between NMP units based on the average R-to-W ratio of each application. As a result, for memory-bound applications with average R-to-W ratio higher than 3 (memory-read-intensive applications), read latency is the dominant factor in determining the performance. Therefore, such applications are potentially good candidates for near 3D-PCM processing. Based on results shown in Fig. 5, BP, MO, Fr, and SpMV can be categorized into memory-write-intensive applications with average R-to-W ratio less than 3. For such applications, near 3D-DRAM outperforms near 3D-PCM in terms of write latency and potentially write energy. HS, BFS, and X264 can be characterized into memory-read-intensive with average R-to-W ratio higher than 3.

RBL: It is also used as an NMP-centric metric to examine the possibility of processing memory-read-intensive applications close to 3D-DRAM or 3D-PCM. Choosing this metric, we assume a same style buffering and size for row buffer in both DRAM and PCM.

TABLE III
DEVICE LEVEL COMPARISON OF DRAM AND PCM [16] [17] [18]

Characteristics	DRAM	PCM
Standby Power	Refresh Power	Very Low
Access Granularity	64 Byte	64 Byte
Read Latency	20 ∼ 50ns	∼ 50ns
Write Latency	20 ∼ 50ns	150 ∼ 500ns
3D-stacking	Yes	Yes
TSV Power Saving	Less	More
Row Buffer Hit/Miss Latency	40ns/40ns	40ns/128ns

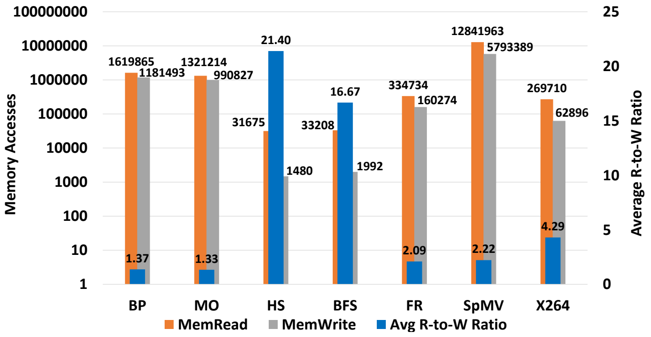

Fig. 5. Application characterization based on average R-to-W ratio (right Y-axis). Memory-intensive applications can be categorized into read-intensive (R-to-W ratio >= 3) and write-intensive (R-to-W ratio < 3).

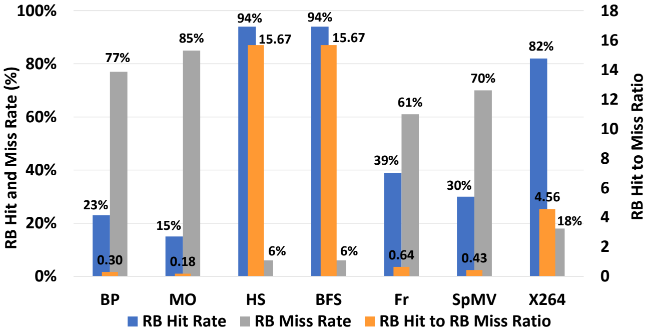

Fig. 6. Left Y-axis shows row buffer hit and miss rate (%) and the right Y-axis shows the ratio of row buffer hit to row buffer miss for memory-intensive applications across all memory channels.

First, we used R-to-W ratio metric to characterize memory-intensive applications into memory-read-intensive and memory-write-intensive. Then, RBL is used to categorize memory-read-intensive applications into: (1) applications with high RBL, and (2) applications with low RBL. As it is shown in TABLE III, Row Buffer (RB) hits are producing same latency in both memory technologies, but RB misses incur higher (3x more) latency in PCM than DRAM. Considering this attribute in both PCM and DRAM, we can conclude that memory-read-intensive applications with high RB hit rate (at least 3x higher than row buffer miss rate) are good candidates for near 3D-PCM processing and memory-read-intensive applications with low RB hit rate are more suitable for near 3D-DRAM processing.

Fig. 6 shows RB hit and miss rate for all memory-intensive applications. Using RBL metric, we characterize memory-read-intensive applications into groups with high RBL and low RBL. Based on the ratio of PCM and DRAM row buffer miss latency ($\frac{PCM\ RB\ miss\ latency}{DRAM\ RB\ miss\ latency}$), see TABLE III), and application's RB hit to RB miss ratio (see Fig. 6), we can conclude that in all three memory-read-intensive applications (HS, BFS, and X264), RB hit to miss ratio is higher than 3. This illustrates that for such applications, near 3D-PCM processing would potentially outperform near 3D-DRAM processing.

III. Insights

In order to suggest the most suitable processing unit (host CPU processing or NMP (3D-DRAM or 3D-PCM)), we redefined four performance metrics to characterize various sets of applications. The first two metrics (temporal locality and OI) categorize applications into CPU-bound and memory-bound and the next two metrics (R-to-W ratio and RBL) classify memory-bound applications into read-intensive and write-intensive groups. Application characterizations are summarized as follows:

I **CPU-intensive (CPU-bound) class with high temporal locality and high OI (OI >= 1.34 Flops/byte):** This class includes applications that can benefit from large cache hierarchies (cache-friendly applications that leverage from temporal locality) and host processing power (OI >= 1.34 Flops/byte or attainable performance = 2.4 GFlops/s which is the peak performance of the host processor). HW and VIPS are examples of **`CPU-friendly`** applications with both high temporal locality and high OI. Host CPU processing is the most suitable processing unit for this class of applications.

II **Memory-intensive (memory-bound) class with low/no temporal locality and low OI (OI < 1.34 Flops/byte):** Characterization results reveal that this class of applications show a significant memory footprint with host CPU under-utilization (OI < 1.34 Flops/byte or attainable performance < peak performance). This class includes **`NMP-friendly`** applications that can benefit from NMP systems. Based on application's R-to-W ratio and RBL, NMP (3D-DRAM) or NMP (3D-PCM) is suggested as the most suitable processing unit. BP, MO, Fr, and SpMV are memory-write-intensive applications (R-to-W ratio < 3) that can be processed efficiently on the NMP (3D-DRAM) unit. HS, BFS, and X264 are memory-read-intensive applications (R-to-W ratio >= 3) with high RBL (RB hit to RB miss ratio > 3) that can be processed efficiently on the NMP (3D-PCM) unit.

III **Mixed class with no/low temporal locality and high OI (OI >= 1.34 Flops/byte):** Characterization results show a mixed behaviour of no/low temporal locality and high OI. For this class of applications, two more memory metrics (memory bandwidth utilization and L2 MPKI) are used to estimate the memory footprint. Based on the results, low memory bandwidth utilization (< 0.02 GB/s) and low L2 MPKI (< 0.2) illustrate that these applications are more **`CPU-friendly`** (high OI and negligible memory footprint) that can utilize host CPU power and cache hierarchies (leveraging spatial locality). SC, BT, BS, and Sw are applications in this class with host CPU processing as the most suitable processing unit.

IV. Conclusion

This paper conducts a thorough performance analysis for various sets of applications and examines aptness of NMP systems (3D-DRAM and 3D-PCM) versus host CPU processing for each class of applications. To find out the best processing unit, a set of performance metrics (temporal locality, OI by Roofline analysis, R-to-W ratio, and RBL) are redefined and utilized for application characterization. Considering the extracted metrics, this paper discusses and proposes the best viable processing unit for each application. Specifically, this work motivates the efficiency of NMP subsystems to process memory-intensive applications when 3D-PCM as an emerging non-volatile memory technology is employed.

Acknowledgement

This article is partially supported by "Tutkijat Maailmalle" grant.

References

[1] W. A. Wulf and S. A. Mckee, "Hitting the memory wall: Implications of the obvious," *Computer Architecture News*, vol. 23, pp. 20–24, 1995.

[2] H. Esmaeilzadeh, E. Blem, R. S. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," in *ISCA*, June 2011, pp. 365–376.

[3] "HMC. Consortium, Hybrid Memory Cube Specification 1.0, 2013."

[4] "HBM. JEDEC Standard, High Bandwidth Memory JESD235B, 2018."

[5] B. C. Lee, P. Zhou, J. Yang, Y. Zhang, B. Zhao, E. Ipek, O. Mutlu, and D. Burger, "Phase-change technology and the future of main memory," *IEEE Micro*, vol. 30, no. 1, pp. 143–143, Jan 2010.

[6] A. Chen, "A review of emerging non-volatile memory (nvm) technologies and applications," *Solid-State Electronics*, vol. 125, 07 2016.

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

[8] "Parboil benchmarks:," http://impact.crhc.illinois.edu/parboil/parboil.aspx.

[9] C. Bienia, S. Kumar, J. P. Singh, and K. Li, "The parsec benchmark suite: Characterization and architectural implications," in *PACT 2008*. New York, NY, USA: ACM, 2008, pp. 72–81.

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

[11] N. Binkert, B. Beckmann, G. Black, S. K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D. R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewell, M. Shoaib, N. Vaish, M. D. Hill, and D. A. Wood, "The gem5 simulator," *SIGARCH Comput. Archit. News*, vol. 39, no. 2, pp. 1–7, Aug. 2011.

[12] M. Poremba, T. Zhang, and Y. Xie, "Nvmain 2.0: A user-friendly memory simulator to model (non-)volatile memory systems," *IEEE Computer Architecture Letters*, vol. 14, no. 2, pp. 140–143, July 2015.

[13] S. C. Woo, M. Ohara, E. Torrie, J. P. Singh, and A. Gupta, "The splash-2 programs: characterization and methodological considerations," in *ISCA*, June 1995.

[14] S. Williams, A. Waterman, and D. Patterson, "Roofline: An insightful visual performance model for multicore architectures," *ACM 2009*.

[15] J. Meza, J. Li, and O. Mutlu, "Evaluating row buffer locality in future non-volatile main memories," 12 2018.

[16] W. Zhang and T. Li, "Exploring phase change memory and 3d die-stacking for power/thermal friendly, fast and durable memory architectures," in *PACT*, Sep. 2009, pp. 101–112.

[17] S. Mittal and J. S. Vetter, "A survey of software techniques for using non-volatile memories for storage and main memory systems," *IEEE TPDS*, vol. 27, no. 5, pp. 1537–1550, May 2016.

[18] S. Chen, P. B. Gibbons, and S. Nath, "Rethinking database algorithms for phase change memory," in *CIDR*, January 2011.