Integrated Thermal Analysis for Processing In Die-Stacking Memory

Yuxiong Zhu Borui Wang Dong Li[‡] Jishen Zhao
University of California at Santa Cruz, [‡]University of of California at Merced
{yzhu29, bwang27, jishen.zhao}@ucsc.edu, dli35@ucmerced.edu

ABSTRACT

Recent application and technology trends bring a renaissance of the processing-in-memory (PIM), which was envisioned decades ago. In particular, die-stacking and silicon interposer technologies enable the integration of memory, PIMs, and the host CPU in a single chip. Yet the integration substantially increases system power density. This can impose substantial thermal challenges to the feasibility of such systems. In this paper, we comprehensively study the thermal feasibility of integrated systems consisting of the host CPU, die-stacking DRAMs, and various types of PIMs. Compared with most previous thermal studies that only focus on the memory stack, we investigate the thermal distribution of the whole processor-memory system. Furthermore, we examine the feasibility of various cooling solutions and feasible scale of various PIM designs under given thermal and area constraints. Finally, we demonstrate system run-time thermal feasibility by executing two high-performance computing applications with PIM-based systems. Based on our experimental studies, we reveal a set of thermal implications for PIM-based system design and configuration.

CCS Concepts

•**Computer systems organization** → **Data flow architectures; Heterogeneous (hybrid) systems;** *Special purpose systems;*

Keywords

Processing-in-memory; Die stacking; Interposer; Thermal; High-performance computing

1. INTRODUCTION

Processing-in-memory (PIM), also known as near-memory computing or near-data processing, builds on the basic idea of integrating computation directly in memory devices [31, 14, 17, 53, 37, 21, 11, 36, 50, 52]. After decades of dormancy, it re-emerges in a new form due to recent application and technology trends. On the application front, in-memory databases [57, 60], web-scale applications [47, 15], high-performance computing [54, 10, 30, 62], and in-situ data processing such as scientific visualization for real-time analysis [38] manipulate increasingly large volumes of data in memory. Data movement between CPU and memory is becoming one of major contributors to system energy consumption and performance degradation [64, 3, 46]. This motivates the demand for moving computation close to memory, where working data is located. On the technology front, recent advances of 3D-stacked memory [48, 12, 34, 67, 68, 9] enable the stacking of a logic (silicon) die implemented by a high-performance technology process with one or more memory (e.g., DRAM) layers. The logic die offers sufficient silicon area and performance capability to implement various logic and computation functions, such as adders, memory copiers, CPU cores, and GPUs [13, 4]. Recent studies [19, 1, 2, 32, 51, 33, 65, 20, 4, 13, 65, 33] demonstrate that such integration technologies is likely to enable PIM in a practical manner.

One major concern in adopting PIMs with die-stacking memory is thermal feasibility. Prior studies demonstrated the thermal feasibility of integrating programmable PIMs with 3D-stacked memory [13]. However, most previous related work only focuses on studying the thermal issues of PIM-based memory stack itself; the thermal feasibility of the integrated system – the host CPU and the memory stack – remains largely unknown. Die-stacking memories are typically integrated with a host CPU on a silicon interposer [61, 35] in a single chip. This effectively reduces the footprint of processor-memory system, but also increases its density. As such, the integration of memory, PIMs, and the host CPU can significantly increase system power density and impede heat dissipation, reducing the thermal feasibility of PIM-based designs. Studying the memory stack alone is insufficient to understand the thermal feasibility of the integrated system.

The thermal interaction between the integrated host CPU and the memory stack can intricate the thermal analysis. Host CPU heat dissipation can heavily impact the temperature of

MEMSYS 2016 October 3–6, 2016, Washington, DC, USA
© 2016 ACM. ISBN 978-1-4503-4305-3…$15
DOI: http://dx.doi.org/10.1145/2989081.2989093

the memory stack. With much higher logic density than the memory stack, the host CPU can dissipate much more heat than the memory stack. Therefore, instead of heated by itself, the memory stack may be heated up by the host CPU.

On the flip side, the memory stack can also impact thermal constraint of the host CPU. Most high-end CPUs used by data-intensive applications can tolerate over 100 °C[25], which is much higher than DRAM's typical operating temperature range (typically under 85 °C). As a result, the integration of the memory stack can tighten the thermal constraint of the processor chip[1].

The goal in this paper is to investigate the thermal feasibility of the system that consists of the host CPU and the PIM-based memory stack. Toward this end, we perform a comprehensive thermal analysis with a variety of system configurations and applications. First, we demonstrate that large-scale programmable PIMs require commodity-server or high-end active cooling solutions (Table 4), even though we only consider the stand-alone memory stack without the thermal impact of the host CPU. Second, we investigate the thermal interaction between the host CPU and the PIM-based memory stack as a function of the distance between the two. Third, we explore the PIM design space by stretching the scale of a variety types of PIMs under given thermal and area budgets. Finally, we demonstrate the thermal feasibility of PIM-based systems by evaluating the run-time thermal distribution with two high-performance applications. This paper makes the following contributions.

- We comprehensively investigate thermal feasibility of the entire integrated system consisting of the host CPU and PIM-based memory stack with various cooling solutions.
- We demonstrate the thermal interaction between the host CPU and the memory stack, and its impact on system thermal constraint and feasible cooling solutions.
- We explore the PIM design space under certain thermal and area constraints, and identify the key constraints to the scale of various types of PIMs.
- Based on our experimental study, we reveal a set of thermal implications for PIM-based system design and configuration.

2. BACKGROUND AND MOTIVATION

Thermal analysis for the integrated host CPU and memory as a system is essential to demonstrate the feasibility of PIM. However, this thermal analysis is challenging due to the complex interaction among system components. This section describes background on modern PIM techniques, processor-memory integration, and thermal management. We also motivate our thermal study by investigating the thermal constraints and challenges in PIM-based systems.

[1]In fact, the first die-stacking-memory-based processor products, e.g., AMD's Fury X GPUs, need to adopt liquid cooling to meet thermal constraints [5]. We envision that the integration of CPU and die-stacking memory also require higher-end cooling solutions than before.

2.1 Processing In Die-Stacking Memory

Since its debut in 90s, PIM has been explored as a promising solution to accelerate data processing and improve host CPU utilization. But traditional PIM techniques were not widely adopted due to their cost, design complexity, and use case limitations. The interest in PIM is reignited due to recent application and technology advancement.

Application Requirement. Many modern applications process large and heterogeneous data sets stored in memory with increasingly large capacity. Data movement between memory and CPU is becoming a critical system performance and energy bottleneck. In addition, a large portion of data processing merely involve simple arithmetic, data movement, and data duplication operations. These do not require complex logic and computation power offered by CPU.

Technology Feasibility of PIM. Modern PIM techniques may be built with various emerging technologies and architectures, such as 3D die-stacked memory [48, 22], nonvolatile memory [8], and automata processor [23]. We study on a die-stacking-memory-based PIM design, which is one of the most promising PIM approaches being explored in both academia and industry [13, 4]. Today, die-stacking memories can stack several memory dies on top of a logic die. The logic die can accommodate sophisticated logic and computation functionality, as long as it employs true logic process. One such example is hybrid memory cube (HMC)-style die-stacking memories [48, 22]. Whereas initial versions of HMC merely offer simple logic at the base (e.g., NoC, I/O drivers, memory controllers, and simple arithmetic and atomic functions) with large process node, its logic die has plenty of area to accommodate processor cores and accelerators by adopting recent process nodes [48, 22][2].

Processor-Memory Integration. 3D die-stacked memories are electrically connected with the host CPU (also referred to as the host processor) side-by-side using a silicon interposer [61, 35, 56] (Figure 1). Silicon interposer is a passive substrate with through-silicon vias (TSVs) [61] and wires that interconnect multiple dies sitting side by side, which is often referred to as 2.5D integration. As such, systems integrated with host CPU and 3D-stacked memory are referred to as 2.5D+3D integrated systems. Such technology provides orders of magnitude denser interconnections between the integrated host CPU and memory stack. It also dramatically reduces the distance among host CPU, memory, and PIMs, significantly increasing the density of logic and memory components.

2.2 Thermal Issues with Processing In Die-Stacking In-Package Memory

[2]JEDEC standard high-bandwidth memory (HBM) [29] logic die employs DRAM process technology so its logic capability may be less powerful than HMC; programmable PIMs may not be able to be integrated closely with the memory stack but need to sit side-by-side with the memory stack and the host CPU on a silicon interposer [61, 35]. The loose integration can reduce the flexibility of PIMs to exploit memory internal interconnection, yet incurs less thermal constraints. Therefore, our study focuses on the HMC-style design.

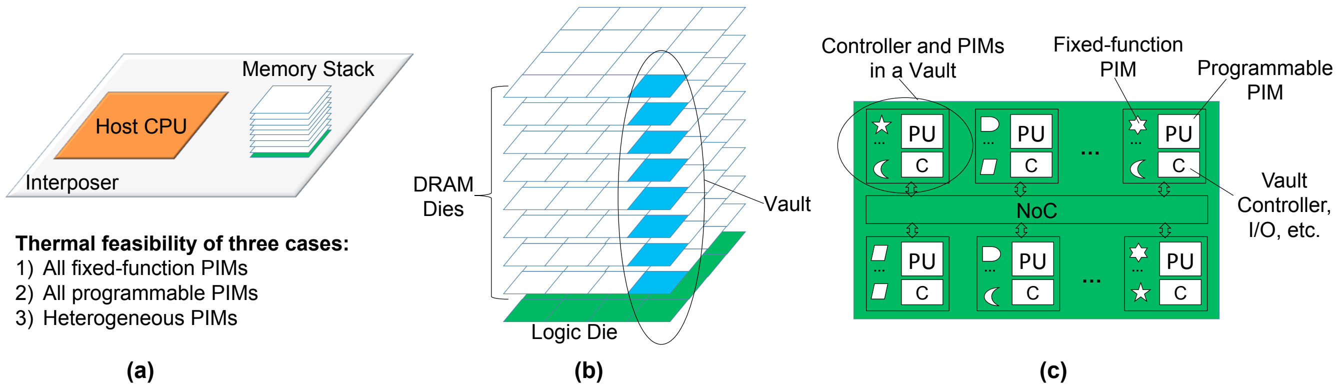

Figure 1: System configuration. (a) Overview of the integrated host CPU and memory stack on a silicon interposer. (b) Memory stack configuration. (c) A logic view of the logic die with heterogeneous PIMs, which consists of both fixed-function and programmable PIMs.

Most previous work focuses on investigating the thermal feasibility of implementing PIMs in memory stack [13, 55, 42, 49]. However, when the memory stack is integrated with the host CPU in a single package; the thermal feasibility of the processor-memory PIM system remains largely unexplored. In particular, the thermal dissipation of host CPU, memory, and PIMs can interact with each other, making the thermal profile of the whole system not straightforward.

The host CPU performs compute-intensive operations with large-scale, power-hungry logic and arithmetic components. The host CPU can operate at a much higher temperature than memory. Therefore, the heat dissipation of the host CPU can substantially impact the temperature of the memory stack, while the heat generated by the memory stack itself may be less significant. In addition, increasing the distance between the host CPU and the memory stack can mitigate such thermal impact. Therefore, our study evaluates the thermal impact of the host CPU to the memory stack as a function of the distance between the two.

The memory stack can also impact the thermal constraint of the host CPU due to the lower operation temperature range of memory. JEDEC stipulates that systems running beyond 85 °C need to double memory self-refresh rate [28]; the rate continues to double for every ~10 °C degree beyond 85 °C˜ [41]. Doubling the refresh rate incurs much higher energy and performance overhead, hence is especially undesirable. As a result, the processor-memory system needs to maintain a temperature lower than 85 °C even though the host CPU can tolerate much higher temperature [25]. Recent work [44, 45] shows that it is critical to study the thermal feasibility of the processor-memory system rather than the memory stack alone; yet these studies did not investigate the scenarios with PIMs in the logic layer. As such, we investigate the thermal impact of such thermal constraint to the feasibility of PIM-based system design.

3. SYSTEM CONFIGURATIONS AND MODELING

We study a system that consists of a host CPU and a memory stack integrated on a silicon interposer, as depicted in Figure 1. The memory stack consists of DRAM dies and a logic die that incorporates PIMs. We investigate the thermal feasibility of the system with various types and configurations of PIMs. While a processor-memory integrated system can have multiple memory stacks, this work focuses on studying systems with one memory stack. Multiple memory stacks can interact with the host CPU in similar manners and therefore will have the similar thermal feasibility as the case of one memory stack. We leave the investigation of multiple memory stacks as future work.

3.1 Host CPU Configurations

We model the host CPU with an architecture similar to Intel Xeon processors with four cores. Table 1 lists detailed architecture configurations and modeling parameters of the host CPU.

3.2 Memory Stack Configurations

We model an HMC-style memory stack, which has eight DRAM layers stacked on top of a logic die. The memory stack is divided into 16 vertical slices, also called *vaults* as shown in Figure 1(b). Each vault has its own independent TSV bus and vault controller [48]. This allows each vault to operate in parallel similar to independent channels operating in conventional DRAM-based memory systems [4].

DRAM Die Configurations. The memory stack has a total capacity of 4GB, which is similar to the latest implementations of die-stacking memory [27, 13, 48]. The HMC-style memory can adopt either DDR3 or DDR4 DRAMs. We employ DDR4, which is the latest DRAM technique.

Logic Die Configurations. We assume that the area of the logic die matches that of the DRAM dies. This is in line with the HMC designs [48, 22]. While this is not a hard constraint for the memory stack design, doing so can reduce manufacturing and vertical routing complexity. In addition, this area is sufficient to accommodate various PIMs, because PIM is intended to supplement the host CPU rather than implement full processing capability [13]. As shown in Figure 1(c), the

Table 1: Host CPU Parameters.

Technology node	22 nm
Die size	354 mm^2
Number of cores	4
Clock rate	3.7 GHz
Thermal design power (TDP)	140 W
L1 cache (private)	SRAM, 64 KB per core
L2 cache (private)	SRAM, 1 MB per core
L3 cache (shared)	SRAM, 10 MB

logic die incorporates vault controllers (memory controlling logic, such as I/O drivers and memory controllers), NoC, and PIMs.

PIMs in the Logic Die. In general, PIMs can be classified as fixed-function PIM and programmable PIM [43]. Fixed-function PIMs offer simple computing/logic functions and are accessed through assembly-level intrinsic or simple library calls [1, 32]. Programmable PIMs are able to execute standard (although possibly PIM-enhanced) programming paradigms that are appropriate for a specific type of computing device [16, 21, 59].

In this work, we investigate the thermal feasibility of integrating 1) only fixed-function PIMs, 2) only programmable PIMs, and 3) heterogeneous PIMs that consist of both types. With fixed-function PIMs, we model various simple logic and arithmetic functions, e.g., adder, multiplier, AND, OR, XOR, shifting, dot product, memory copier, memory mover, compare-and-swap, fetch-and-add, test-and-set, sorting, and scatter-gather. Our power and area modeling (Section 3.3) shows that they can be classified into two categories – simple PIM and complex fixed-function PIM. A typical simple fixed-function PIM can contain a multiplier, a divider, or a combination of an adder, a shifter, and a logical unit; a complex fixed-function PIM can contain floating point units or multi-functional logic and arithmetic functions. Our thermal analysis abstracts these fixed-function PIMs as either simple or complex ones without distinguishing among individual functions. With programmable PIMs, we model in-order processing unites (PUs) distributed across the 16 vaults. Each PU is modeled as an ARM Cortex-A9 core with a 2.0 GHz clock rate, but modified to have an in-order pipeline. Each PU is placed next to its home vault router to reduce routing complexity and improve performance.

Previous work shows that neither type of PIMs is an obvious performance winner, given the variety of applications that are likely to benefit from PIMs [19, 1, 2, 32, 51, 33, 65, 20, 4, 13, 65, 33]. Therefore, we also examine the feasibility of a heterogeneous PIM design, which integrates both fixed-function and programmable PIM in the logic layer.

We refer the logic die components other than the PUs as fixed-function units, because most of them offer fixed functionality. These fixed-function units can include vault controllers, NoC, and fixed-function PIMs (if any). We also refer the fixed-function units in each vault as a fixed-function unit group (FFUG).

3.3 Modeling and Evaluation Methodology

Table 2: Peak power breakdown of each DRAM die.

ACT	18.8 mW
Total activate power	18.8 mW
RD	39.2 mW
WR	30.5 mW
READ I/O	45.3 mW
Write ODT	9.7 mW
Total RD/WR/Term power	124.8 mW
ACT_ STBY	37.1 mW
PRE_ STBY	12.0 mW
ACT_ PDN	4.7 mW
PRE_ PDN	2.0 mW
REF	6.7 mW
Total background power	62.5 mW
Total DRAM layer power	206.0 mW

Area and Power Modeling. We adopt process technology nodes used in modern processors and memories: 22nm technology node with the host CPU and the logic die of the memory stack; 25nm technology node with the DRAM dies. We calculate peak power and area of the host CPU based on published parameters of Intel Xeon processors [24]. Table 1 lists detailed parameters of the host CPU. Similar to the latest processing in die-stacking memory implementations, each DRAM die is 68 mm^2 [48, 22, 13], i.e., each vault has 4.25 mm^2. We calculate the power of each DDR4 DRAM die with 1.2V voltage supply, using Micron's DRAM power calculator [26]. Table 2 illustrates the total maximum power and power breakdown of each DRAM layer. We calculate the area and power (leakage and peak dynamic) of the programmable PIM PUs using McPAT [39]. Each PU has an area of 1 mm^2 and 0.96 W peak power. We estimate the power and area of various types of fixed-function logic and computation units using Synopsys Design Compiler. To simplify thermal analysis, we categorize them into two classes: simple fixed-function PIMs have peak dynamic power and area below 0.01 W and 0.06mm^2, respectively; complex ones have peak dynamic power and area larger than 0.015 W and 0.25 mm^2.

In the case of heterogeneous PIMs, the area budget left for FFUG (fixed-function PIMs, vault controller, and interconnects) in each vault is 3.25 mm^2 (4.25 mm^2 per vault excluding a 1 mm^2 PU). In particular, Figure 2 illustrates the floorplan of an example logic layer with heterogeneous PIMs, where CORE1-16 represent PUs and FIX1-16 represent FFUGs.

We also model run-time dynamic power in order to investigate the system dynamic thermal distribution, when we execute various data-intensive applications. To model dynamic power of the host CPU, vault controllers, and PUs, we feed performance statistics into McPAT [39] power model. The performance statistics are obtained by gem5 [7] simulation of our evaluated workloads. Dynamic power of fixed-function PIMs are estimated with the activity ratio based on the simulation.

Thermal Modeling. We use HotSpot 6.0 [66] to analyze thermal distribution of the processor-memory system with

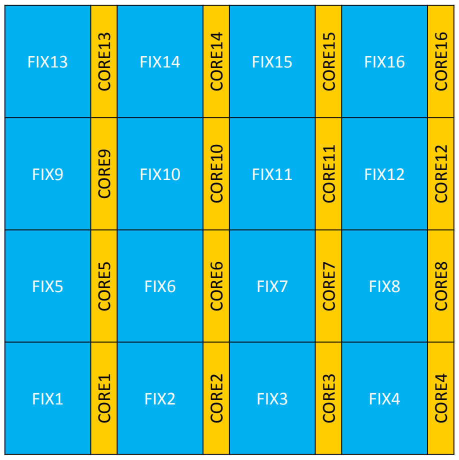

Figure 2: An example logic die floorplan with heterogeneous PIMs.

Table 3: Memory stack thermal modeling parameters.

Parameter	**Value**
Silicon thermal resistivity	0.0083 $\frac{m-K}{W}$ [42, 13]
Metal-layer thermal resistivity	0.083 $\frac{m-K}{W}$ [42, 13]
Die-to-die-layer thermal resistivity	0.0166 $\frac{m-K}{W}$ [42, 13]
Die size	68 mm^2
Die count	8 (memory), 1 (logic)
Ambient temperature (computer box)	45 °C [58, 45, 42]

various PIM designs. To evaluate system thermal distribution, we draw floorplans according to the die photos of corresponding Xeon CPU and ARM cores. We then generate thermal maps by feeding the floorplans, power traces, and thermal configurations of system components into HotSpot [66]. Table 3 lists the key parameters in our memory stack thermal modeling.

We evaluate both peak steady-state and dynamic thermal distributions: the former can implicate thermal constraints introduced by various PIM designs; the latter shows the thermal feasibility of running various applications with PIMs. To evaluate peak steady-state thermal distribution, we employ the leakage and peak dynamic power power obtained from McPAT [39] to generate power traces with various PIM designs. To evaluate dynamic thermal distributions, we simulate two HPC workloads with gem5 simulator [7] and obtain performance statistics of various program phases. We annotate the application source code to identify the program phases. We then generate run-time power traces by feeding the performance statistics into McPAT [39].

We evaluate various cooling solutions to investigate the cost of required cooling to meet the system thermal constraint. Table 4 lists four cooling solutions employed in our study. They all employ heat sinks. The passive cooling only adopts a heat sink; the three active cooling solutions adopt heat sinks plus cooling fans with various cost-performance trade-offs as illustrated in Table 4. Among them, the low-end active cooling adopts inexpensive consumer-level heat sinks

Table 4: Evaluated cooling solutions.

Cooling Solutions	**Convection Thermal Resistance (°C/ W)**	**Cost**
Passive cooling	4.0 [18]	~$12
Low-end active cooling	2.0 [13]	~$30
Commodity-server active cooling	0.5 [49]	~$90
High-end-server active cooling	0.2 [13]	~$250

and fans [13]. The convection thermal resistance values are calculated from specific cooling solutions. The costs of these cooling solutions are collected from various vendors.

Similar to previous studies [13], to simplify the thermal modeling and analysis, we only studies the logic die components that can have significant thermal impacts, such as PUs, fixed-function PIM execution units, and buffers. Other hardware resources (e.g., memory controllers) have relatively low thermal impacts. Therefore, we do not model them in detail, yet leave constant power budget for them in our thermal analysis.

4. THERMAL FEASIBILITY ANALYSIS

4.1 Memory Stack Only Analysis

We first investigate thermal distribution of the memory stack (no host CPU in the package) as a baseline. We examine the cases with both fixed and various ambient temperatures.

Our first set of experiments use constant ambient temperature (45 °C a typical ambient temperature in computer boxes [58, 45, 42]), and evaluate system temperature with various cooling solutions. Figure 3 shows the thermal map with an active cooling solution used for high-end servers. Note that the temperature 86.89 °C shown on the right-hand side of Figure 3 is the peak temperature of the whole memory stack. Yet only the peak temperature of DRAM dies matters to thermal feasibility, because DRAM needs to operate at lower than 85 °C [28]. Therefore we also analyze peak DRAM temperature based on HotSpot text output (not shown in the thermal map). Our evaluation results demonstrate that the peak power of the logic die cannot exceed 5.16 W per vault, in order to maintain DRAM temperature below 85 °C. This power budget is sufficient for accommodating high-end programmable PIM computing capabilities. Similarly, we also evaluate the impact of other cooling solutions. Commodity-server active cooling solutions can sustain up to 3 W per vault in the logic layer without violating the thermal constraint of DRAMs. This power budget is sufficient for accommodating the programmable PIMs and/or a large number of fixed-function PIMs. We will investigate the feasible number of fixed-function PIMs in Section 4.3. However, ***neither passive nor low-end active cooling solutions can secure thermal feasibility for memory stacks with 16 programmable PIM PUs, which is required to accelerate data-intensive applications in some existing work [1, 4].*** The peak temperature of DRAM dies can exceed 85 °C, even

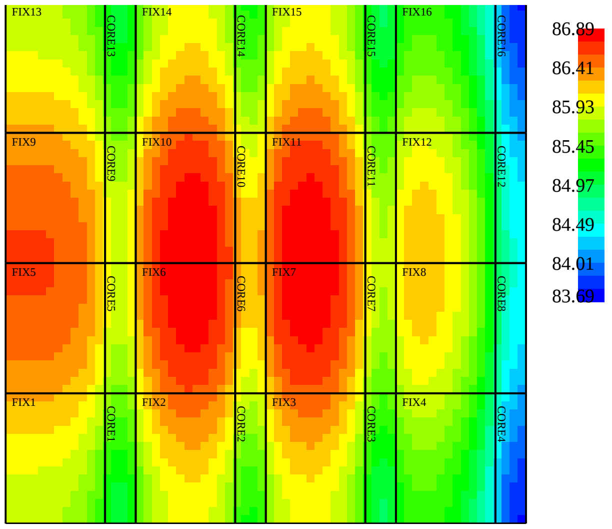

Figure 3: **Thermal map of the hottest DRAM die in the memory stack with high-end server active cooling solutions.**

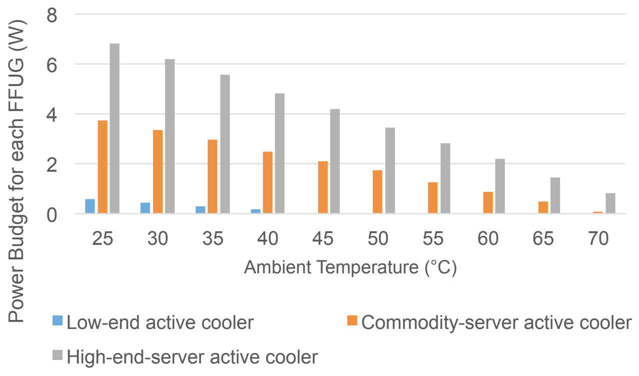

Figure 4: **FFUG power budgets under various ambient temperatures.**

if only 16 PUs are running and the rest of the logic layer is idle.

Power Budget of FFUGs. Fixed ambient temperature is not always the case. To examine the impact of various ambient temperatures to the logic die power budget, we evaluate the thermal distribution by varying the ambient temperatures from 25 °C to 70 °C across various cooling solutions. Figure 4 illustrates our results. Not surprisingly, the higher the ambient temperature, the lower the power budget will be. We do not show the power budget of each FFUG with passive cooling solutions, because the power budget is either very low (<0.1 W) or zero. The figure also shows that low-end active cooling solutions are infeasible, when ambient temperature exceeds 40 °C. An ambient temperature of 70 °C results in a very low power budget for FFUGs, even with active cooling solutions at the grade of commodity or high-end server.

4.2 Thermal Interaction Between the Host CPU and the Memory Stack Integrated with PIMs

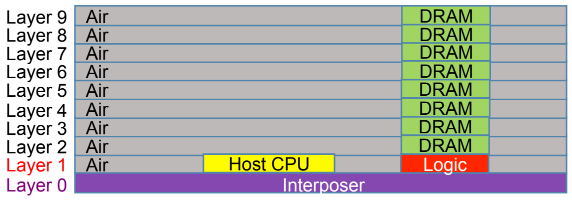

Figure 5: **The side view of the HotSpot model of our 2.5D+3D PIM.**

We investigate the thermal interaction between the host CPU and the memory stack, when they are integrated on a silicon interposer in the same package. The physical proximity of the host CPUs and the memory stack can exacerbate thermal issues and increase system power. As such, we expect to see an increase in temperatures of both the host CPU and the memory stack when placing them close to each other.

So far, HotSpot [66] does not support the 2.5D+3D integration. Therefore, we modify HotSpot [66] to model our processor-memory system in a way shown in Figure 5. Layer 0 is the interposer. Because it is a passive layer, we assume its power is zero or negligible. We place both the host CPU and the logic die of the memory stack in Layer 1; the rest regions of the layer are dummy components, which are modeled as air and consume zero power. Layers 2 to 9 consist of DRAM dies and dummy components. We add these dummy components in each layer, because HotSpot [66] requires that the dimensions of all layers in a 3D stack need to be identical.

Thermal Impact of the Host CPU to the Memory Stack as a Function of Distance. To investigate the thermal impact of the host CPU to the memory stack, we explore the scenario when only the host processor is active and the memory stack is idle. Again, we set the ambient temperature to be 45 °C and employ high-end server active cooling solutions. Originally, we set the distance between the host processor and the memory stack to be 10 mm. The 10 mm distance is impractical, but sufficiently long to illustrate the low temperature coupling between the host CPU and the memory stack [69]. As shown in Figure 6, although the host processor only occupies a portion of the package, it increases temperature of the memory stack to be above the ambient temperature. As a result, the "effective ambient temperature" in the package rises to ∼72 °C. Peak temperatures of the host processor and DRAMs are 85.71 °C and 72.63 °C , respectively. Therefore, the evaluated system configuration is feasible in terms of thermal, when the memory stack is idle.

To further explore the thermal impact of the host CPU, we vary the distance between the host CPU and the memory stack (still in idle) and examine the peak DRAM temperature. As shown in Figure 7, we reduce peak DRAM temperature from 75.23 °C to 72.63 °C by increasing the distance from 1 mm to 10 mm. In addition, we observe that the declining rate is close to a linear rate (slightly slower than the linear rate). Therefore, the thermal impact of the host CPU

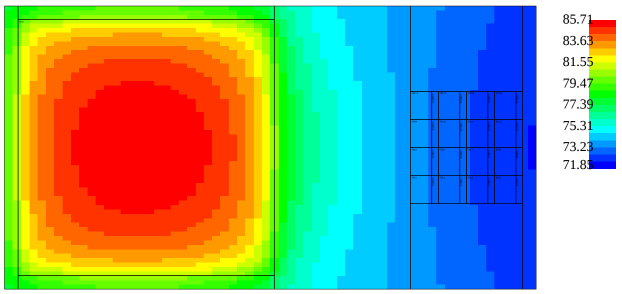

Figure 6: **Thermal map when only the host processor is active. The component on the left is the host processor.**

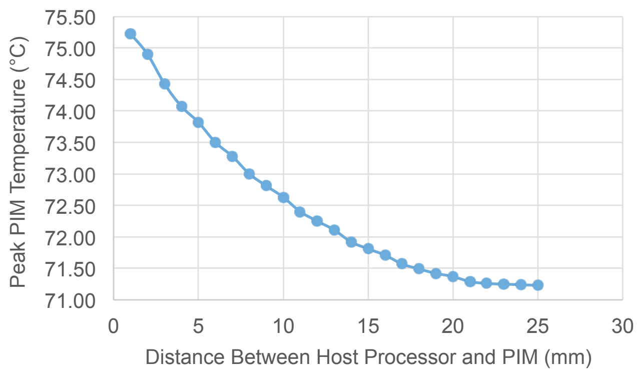

Figure 7: **Peak PIM temperature with different distances to the host Processor, when PIM is passive.**

is roughly a linear function of the distance between the host CPU and the memory stack. However, after this distance is longer than 15 mm, the thermal impact tends to be much stabler when the distance further increases.

Thermal Interaction Between the Host CPU and the Memory Stack. To study the thermal interaction, we make the memory stack active (in normal operating state) too. We set the distance between the host processor and the memory stack to be 10 mm. We fix the power of FFUG (with or without fixed-function PIMs) to be 1 W, which is similar to the peak power of a PU. Figure 8 shows a thermal map in this scenario. Comparing with Figure 6, we observe the thermal interaction between the host CPU and the memory stack, while ***the thermal impact of the host CPU to the memory stack dominates.*** The peak temperature of the host processor increases by 6.48 °C , from 85.71 °C to 92.19 °C . The peak temperature of DRAMs is 88.48 °C , which is increased by 13.25 °C . In this case, the DRAM temperature exceeds the 85 °C thermal constraint, even though the power of FFUG is only 1 W – we need to further reduce FFUG power budget to make the system design feasible. On the flip side, the memory stack constraints the choice of cooling solutions. Most server CPUs can tolerate the 92.19 °C with passive or low-end active cooling solutions. However, ***the integration of the host CPU with memory determines that high-end active cooling solutions must be used.***

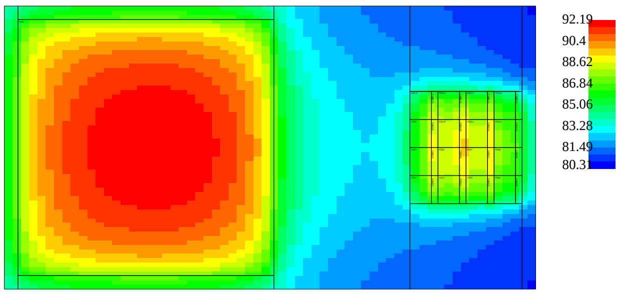

Figure 8: **Thermal map when both the host processor and the memory stack are active.**

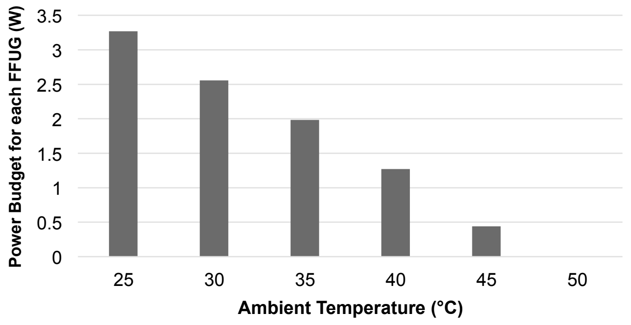

Figure 9: **Power budget of each FFUG under various ambient temperatures. (Only the power budgets with the active cooling solutions for high-end servers are shown.)**

Power Budget of FFUGs. In order to examine the feasible power budget of each FFUG, we keep the distance between the host CPU and memory stack to be 10 mm yet vary the ambient temperature and cooling solutions. With the above configuration, all of the cooling solutions except the active cooler for high-end servers cannot control the peak temperature of DRAM under 85 °C. Figure 9 shows the results for the power budgets with the active cooler for high-end servers.

4.3 Feasible Scale of PIMs Under Given Thermal and Area Budgets

To explore the design space of PIMs, we study the impact of thermal and area constraints on the scale of programmable, fixed-function, and heterogeneous PIMs. The maximum scale of PIMs need to meet either thermal or area constraint, whichever is tighter.

Scale of Programmable PIMs. Given 68 mm^2 as the total area of the logic die, each vault has a 4.25 mm^2 area budget if we evenly divide the logic die among 16 vaults. This area budget can only fit up to four PUs in each vault, assuming no FFUGs – this is impractical, but provides an extreme case for the scale of programmable PIMs. We investigate the thermal distribution with our previous experimental deployment: 45 °C ambient temperature, the active cooling solutions for high-end servers, and 10 mm distance between host processor and the memory stack. Our results indicate that we need to maintain the peak power of each PU below 0.4 W in order to avoid violating the thermal constraint of 85 °C.

Scales of Fixed-Function and Heterogeneous PIMs. Table 5 illustrates the maximum feasible scales of fixed-function and heterogeneous PIMs in the whole logic die. With the heterogeneous PIM, we assume that each vault has only one PU. The table also shows a tighter constraint between area and thermal budgets. Determined by area and peak power, a typical simple fixed-function PIM can contain a multiplier, a divider, or a combination of an adder, a shifter, and a logical unit. Complex fixed-function PIMs can contain floating point units or multi-functional logic/arithmetic functions. In addition to simple and complex fixed-function PIMs, we also calculate the feasible scale for heterogeneous PIMs with several representative functions, such as in-memory data movement and atomic operations.

The result shows that ***the scale of fixed-function PIMs is subject to the area constraint.*** up to 1168 simple fixed-function PIMs can fit in the logic layer without violating any constraints. But further increasing the number can violate the area constraint. With larger area per PIM, only up to 224 complex fixed-function PIMs can be placed in the logic layer before violating the area constraint. To study the design space of heterogeneous PIMs, we first place one PU in each vault in the logic die. We then place as many fixed-function PIMs as possible until either the thermal/power constraint or the area constraint is broken. Our results show that ***the scale of heterogeneous PIMs is most likely subject to the thermal/power constraint,*** because the PUs consume a large portion of the power budget. Only heterogeneous PIMs with complex fixed-function units are constrained by area, because complex fixed-function units, such as floating-point units, have relatively larger area overhead.

4.4 Run-time Thermal Analysis

In this section, we evaluate the thermal feasibility of PIM designs by executing data-intensive applications. The ambient temperature (45 °C), cooling solution (active cooling used for high-end servers), and the distance between the host CPU and the memory stack (10 mm) are set as the same as in previous experiments. In addition, we employ a detailed floorplan of the host CPU, with details of each CPU core and L1/L2/L3 caches. We modify gem5 simulator [7] to collect performance statistics for each program phase on our processor-memory system with PIMs. The performance statistics include instruction count, memory access, cache misses, on our processor-memory system with PIMs. We then feed the performance statistics into McPAT [40] to estimate system dynamic power as described in Section 3.3.

Workload Characteristics. We investigate CG and MG workloads from the NAS parallel benchmark suite [6].

CG uses the inverse power method to find an estimate of the largest eigenvalue of a symmetric positive definite sparse matrix with a random pattern of non-zeros. The computation of CG is dominated by a multiplication-addition operation

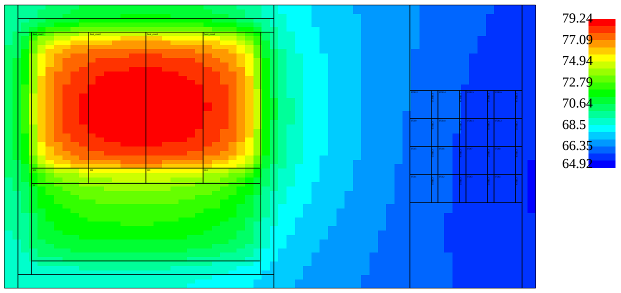

Figure 10: Thermal map of executing CG with the host CPU, while all the PIMs are disabled. The peak dynamic temperatures achieved by the host CPU and DRAMs are 79.24 °C and 65.51 °C, respectively.

represented as $a = b + c * d$. In many cases, a, b, c and/or d in the operation are the elements of specific vectors or matrices. These memory accesses come from indirect data references. These memory accesses can be random and have poor data locality. The memory access pattern of CG with indirect data references is because of the compressed row storage (CRS) format for storing sparse vectors/matrices. The memory access pattern with indirect data references is common in sparse linear algebra. Because of the poor data locality in this memory access pattern, the traditional CPU-based computation can cause lots of cache misses and frequent data movement between CPU and main memory. For CG to use fixed-function PIMs, we offload the primitive multiplication-addition operations to the PIMs. To use the programmable PIM, we offload the most computation-intensive loop (particularly the one in the $conj\_grad$ routine).

MG approximates the solution of a three-dimensional discrete Poisson equation using the V-cycle multi-grid method on a rectangular domain with periodic boundary conditions. In the V cycle, the computation starts from the finest refinement level, going down level by level toward the bottom, then back up to the top. The V-cycle multi-grid method involves applying a set of stencil operations sequentially on the grids at each level of refinement [63]. The stencil operations happen in various execution phases, including restriction, prolongation, evaluation of residual, and point relaxation. The stencil operations in MG often involve a 4-point stencil. To use fixed-function PIMs, we offload these stencil operations to PIMs. To use the programmable PIM, we offload the major computation routines (particularly $mg3P$ and $resid$).

Thermal Analysis Results. We perform four groups of experiments with these workloads with heterogeneous PIMs.

The first experiment group is a baseline, where we make all the PIMs idle and let the host CPU execute the workloads. Figure 10 and Figure 11 demonstrate the results of running CG and MG, respectively. The peak DRAM runtime temperatures achieved during CG and MG execution are 65.51 °C and 68.28 °C, respectively. They are both be-

Table 5: PIM design space exploration under given area and thermal budgets.

PIM Type	Max Scale of PIMs	Key Constraint
Simple fixed-function PIMs Only	1168	Area
Complex fixed-function PIMs Only	224	Area
Heterogeneous PIMs scale 1	16 PUs + 384 simple fixed-function PIMs	Thermal/Power
Heterogeneous PIM scale 2	16 PUs + 160 complex fixed-function PIMs	Area
Heterogeneous PIM scale 3	16 PUs + 160 (memory copiers& memory movers)	Thermal/Power
Heterogeneous PIM scale 4	16 PUs + 464 (compare-and-swap& compare-and-set)	Thermal/Power

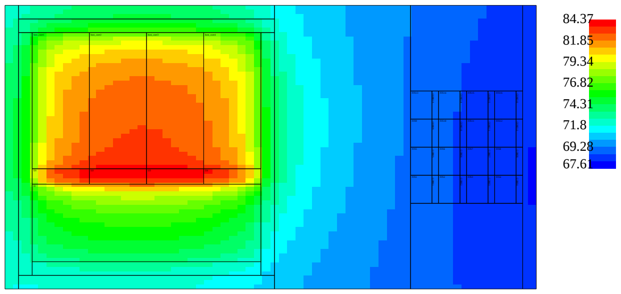

Figure 11: Thermal map of executing MG with the host CPU, while all the PIMs are disabled. The peak dynamic temperatures achieved by the host CPU and DRAMs are 84.37 °C and 68.28 °C, respectively.

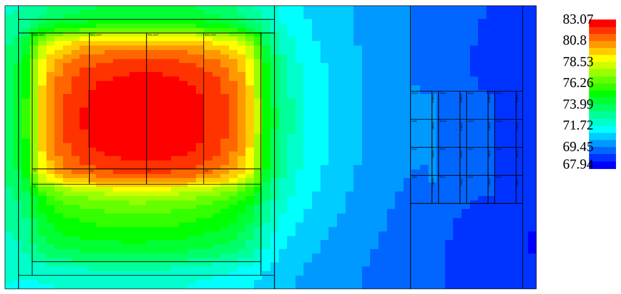

Figure 12: Thermal map of executing CG with the host CPU and programmable PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 83.07 °C and 69.16 °C, respectively.

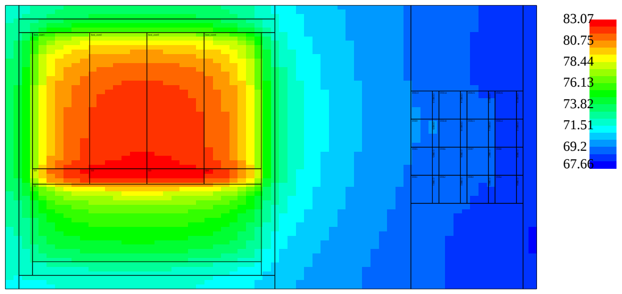

Figure 13: Thermal map of executing MG with the host CPU and programmable PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 83.07 °C and 68.78 °C, respectively.

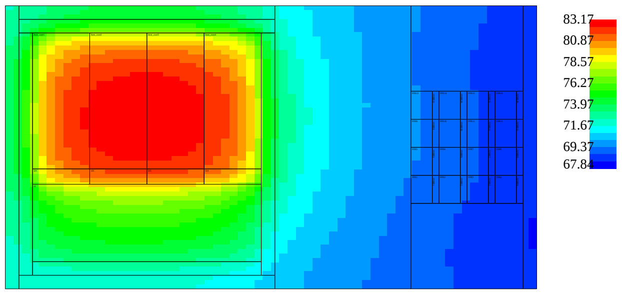

Figure 14: Thermal map of executing CG with the host CPU and fixed-function PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 83.17 °C and 68.78 °C, respectively.

low the 85 °C thermal constraint. We observe that the runtime dynamic power of the host CPU is also lower than its nominal TDP. This is reasonable, because TDP refers to the power that generates the maximum amount of heat that the cooling system is required to dissipate in typical operations. Thus, the thermal effect of the host CPU to the memory stack is weaker than that when the power of the host CPU is close to its TDP.

In the second experiment group, we disable all fixed-function PIMs and only allow the host CPU to offload operations to the programmable PIMs. Figure 12 and Figure 13 show the result of executing CG and MG, respectively. The peak DRAM temperatures during CG and MG execution are 69.16 °C and 68.78 °C, respectively. They are also lower than the 85 °C thermal constraint. Therefore, using programmable PIMs to accelerate CG and MG is feasible in terms of thermal. Furthermore, we make three additional observations by comparing the results with the baseline. First, the peak temperature of the host CPU when running CG with the programmable cores is higher than that when running CG with the host CPU alone. This is due to the different thermal interactions between the host CPU and the memory stack discussed previously. Second, the peak temperature of the host CPU when running MG with the programmable PIMs

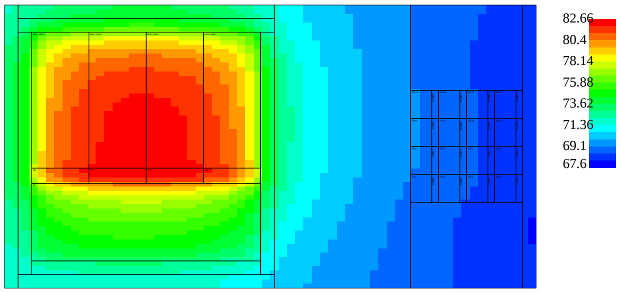

Figure 15: Thermal map of executing MG with the host CPU and fixed-function PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 82.66 °C and 68.53 °C, respectively.

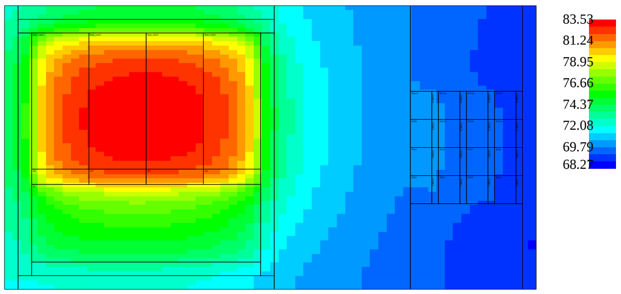

Figure 16: Thermal map of executing CG with the host CPU and heterogeneous PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 83.53 °C and 69.59 °C, respectively.

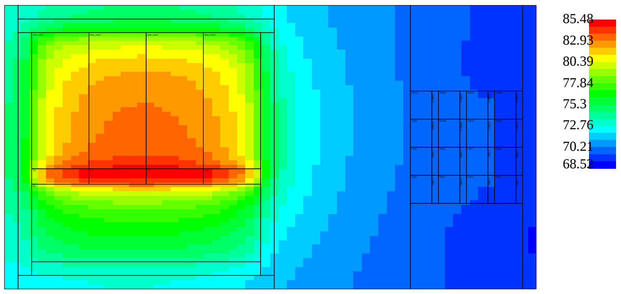

Figure 17: Thermal map of executing MG with the host CPU and heterogeneous PIMs. The peak dynamic temperatures achieved by the host CPU and DRAMs are 85.48 °C and 69.74 °C, respectively.

is slightly lower than that when running MG with the host CPU alone. This is the result of two contradicting effects. On one hand, the thermal interaction between the host CPU and the memory stack increases the peak temperature of the host CPU. On the other hand, because power-hungry operations are offloaded to the PIM, the peak temperature of the host CPU is reduced. For MG, the thermal reduction due to operation offloading offsets the bad thermal effects of the memory stack on CPU. Third, because the run-time power of the programmable PIM cores is much lower than their peak power, the PIM does not seem to have high temperature in the figures.

In the third experiment group, we disable all programmable PIM PUs and only allow the host CPU to offload operations to fixed-function PIMs. The results are shown in Figure 14 and Figure 15. The peak DRAM temperatures during CG and MG execution are 68.78 °C and 68.53 °C, respectively. They are still below the 85 °C thermal constraint and therefore render the feasibility of using fixed-function PIMs to accelerate both workloads.

In the last experiment group, we enable all 16 programmable PIM PUs and fixed-function PIMs, which form a heterogeneous PIM configuration. The results are shown in Figure 16 and Figure 17. The peak DRAM temperatures during CG and MG executions are 69.59 °C and 69.74 °C, respectively. They are still lower than the 85 °C thermal constraint. That said, heterogeneous PIMs are also feasible in terms of thermal.

5. CONCLUSIONS

PIM techniques are re-emerging in a new form, due to recent technology and application advancement. Yet the thermal constraint can be one caveat of adopting PIM. This paper investigates the thermal feasibility of integrating PIMs in the logic die of 3D-stacked memory. Different from most previous work that studies the thermal of the standalone memory stack, we examine the thermal of the whole system consisting of the host CPU and the memory stack. With comprehensive thermal analysis, we provide the following thermal implications to PIM-based system design and configuration:

1. Even considering a standalone memory stack, the stack with large-scale programmable PIMs requires high-end or commodity-server cooling solutions to accommodate to their peak thermal dissipation.
2. The host CPU dominates the thermal impact in the processor-memory system as a function of the distance to the memory stack.
3. The PIM-based memory stack dominates the thermal constraint in the processor-memory system. As a result, commodity-server or high-end active cooling solutions are required by the system, even though the host CPU can tolerate high peak temperatures with passive or low-end active cooling solutions.
4. The feasible scale of fixed-function PIMs is subject to the area constraint of the logic die, while that of heterogeneous PIMs is likely subject to the system thermal/power constraint.
5. Passive and low-end cooling solutions can be feasible, if target applications maintain appropriate activities of

the host CPU, memory, and PIMs and those activities are are much lower than the peak.

6. ACKNOWLEDGMENTS

We thank our shepherd Bruce Jacob and the anonymous reviewers for their insightful comments and valuable suggestions. Yuxiong Zhu and Jishen Zhao is supported by UCSC startup funding. Borui Wang is supported by the UCSC Center for Research in Open Source Software (CROSS). Dong Li is partially supported by the NSF grant CCF-1553645.

7. REFERENCES

[1] J. Ahn, S. Hong, S. Yoo, O. Mutlu, and K. Choi. A scalable processing-in-memory accelerator for parallel graph processing. In *Proceedings of the 42nd Annual International Symposium on Computer Architecture*, pages 105–117, 2015.

[2] J. Ahn, S. Yoo, O. Mutlu, and K. Choi. PIM-enabled instructions: A low-overhead, locality-aware processing-in-memory architecture. In *Proceedings of the 42Nd Annual International Symposium on Computer Architecture*, pages 336–348, 2015.

[3] A. Ailamaki, D. J. DeWitt, M. D. Hill, and D. A. Wood. DBMSs on a modern processor: Where does time go? In *Proceedings of the 25th International Conference on Very Large Data Bases*, pages 266–277, 1999.

[4] B. Akin, F. Franchetti, and J. C. Hoe. Data reorganization in memory using 3D-stacked DRAM. In *Proceedings of the Annual International Symposium on Computer Architecture*, pages 131–143, 2015.

[5] AMD. AMD Radeon R9 series graphics cards. http://www.amd.com/en-us/products/graphics/desktop/r9.

[6] D. Bailey, J. Barton, T. Lasinski, and H. Simon. The NAS parallel benchmarks. Technical Report 103863, NASA, July 1993.

[7] 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*, 39(2):1–7, Aug. 2011.

[8] M. N. Bojnordi and E. Ipek. Memristive Boltzmann Machine: A hardware accelerator for combinatorial optimization and deep learning. In *Proceedings of the IEEE International Symposium on High-Performance Computer Architecture*, pages 1–13, 2016.

[9] K. Chen, S. Li, J. H. Ahn, N. Muralimanohar, J. Zhao, C. Xu, S. O, Y. Xie, J. B. Brockman, and N. P. Jouppi. History-assisted adaptive-granularity caches (HAAG$) for high performance 3D DRAM architectures. In *Proceedings of the 29th International Conference on Supercomputing (ICS)*, 2015.

[10] Y. Cui, E. Poyraz, K. B. Olsen, J. Zhou, K. Withers, S. Callaghan, J. Larkin, C. C. Guest, D. J. Choi, A. Chourasia, Z. Shi, S. M. Day, P. Maechling, and T. H. Jordan. Physics-based seismic hazard analysis on petascale heterogeneous supercomputers. In *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2013.

[11] J. Draper, J. Chame, M. Hall, C. Steele, T. Barrett, J. LaCoss, J. Granacki, J. Shin, C. Chen, C. W. Kang, I. Kim, and G. Daglikoca. The architecture of the DIVA processing-in-memory chip. In *Proceedings of the 16th International Conference on Supercomputing*, pages 14–25, 2002.

[12] R. G. Dreslinski, D. Fick, B. Giridhar, G. Kim, S. Seo, M. Fojtik, S. Satpathy, Y. Lee, D. Kim, N. Liu, M. Wieckowski, G. Chen, D. Sylvester, D. Blaauw, and T. Mudge. Centip3De: A many-core prototype exploring 3d integration and near-threshold computing. *Commun. ACM*, 56(11):97–104, Nov. 2013.

[13] Y. Eckert, N. Jayasena, and G. H. Loh. Thermal feasibility of die-stacked processing in memory. In *WoNDP*, 2014.

[14] D. Elliott, M. Stumm, W. M. Snelgrove, C. Cojocaru, and R. McKenzie. Computational RAM: Implementing processors in memory. *IEEE Des. Test*, 16(1):32–41, Jan. 1999.

[15] T. A. S. Foundation. Spark, 2014.

[16] B. B. Fraguela, J. Renau, P. Feautrier, D. Padua, and J. Torrellas. Programming the FlexRAM parallel intelligent memory system. In *Proceedings of Principles and Practice of Parallel Programming (PPoPP)*, 1999.

[17] M. Gokhale, B. Holmes, and K. Iobst. Processing in memory: The terasys massively parallel PIM array. *Computer*, 28(4):23–31, Apr. 1995.

[18] C. Gonzales and H. M. Wang. Thermal design considerations for embedded applications. In *Intel White Paper*, 2008.

[19] Q. Guo, N. Alachiotis, B. Akin, F. Sadi, G. Xu, T. M. Low, L. Pileggi, J. C. Hoe, and F. Franchetti. 3D-stacked memory-side acceleration: Accelerator and system design. In *WoNDP*, pages 1–6, 20se14.

[20] Z. Guz, M. Awasthi, V. Balakrishnan, M. Ghosh, A. Shayesteh, and T. Suri. Real-time analytics as the killer application for processing-in-memory. In *WoNDP*, pages 1–3, 2014.

[21] M. Hall, P. Kogge, J. Koller, P. Diniz, J. Chame, J. Draper, J. LaCoss, J. Granacki, J. Brockman, A. Srivastava, W. Athas, V. Freeh, J. Shin, and J. Park. Mapping irregular applications to DIVA, a PIM-based data-intensive architecture. In *Proceedings of the 1999 ACM/IEEE Conference on Supercomputing*, 1999.

[22] HMCC. Hybrid memory cube specification 2.0.

[23] HP. Micron's automata processor (AP). http://www.micronautomata.com.

[24] Intel. Intel Xeon processor E5-1630 v3. http://ark.intel.com/products/82764/Intel-Xeon-Processor-E5-1630-v3-10M-Cache-3_70-GHz.

[25] Intel. *Intel Xeon Processor E3-1200 v5 Product Family Datasheet*, 10 2015.

[26] J. Janzen. The Micron system-power calculator. http://www.micron.com/products/dram/syscalc.html.

[27] J. Jeddeloh and B. Keeth. Hybrid memory cube new dram architecture increases density and performance. In *VLSI Technology (VLSIT), 2012 Symposium on*, pages 87–88, June 2012.

[28] JEDEC. *DDR3 SDRAM Specification, JESD79-3E*, July 2010.

[29] JEDEC. High bandwidth memory (HBM) DRAM, 2013. http://www.jedec.org/standards-documents/docs/jesd235.

[30] H. Kashyap, H. A. Ahmed, N. Hoque, S. Roy, and D. K. Bhattacharyya. Big data analytics in bioinformatics: A machine learning perspective. In *arXiv preprint arXiv:1506.05101*, 2015.

[31] W. H. Kautz. Cellular logic-in-memory arrays. *Computers, IEEE Transactions on*, 100(8):719–727, 1969.

[32] T. Kelly, H. Kuno, M. Pickett, H. Boehm, A. Davis, W. Golab, G. Graefe, S. Harizopoulos, P. Joisha, A. Karp, N. Muralimanohar, F. Perner, G. Medeiros-Ribeiro, G. Seroussi, A. Simitsis, R. Tarjan, and S. Williams.

Sidestep: Co-designed shiftable memory and software. Technical report, HP Labs, 2012.

[33] C. D. Kersey, S. Yalamanchili, and H. Kim. SIMT-based logic layers for stacked DRAM architectures: A prototype. In *Proceedings of the 2015 International Symposium on Memory Systems*, pages 29–30, 2015.

[34] D. H. Kim, K. Athikulwongse, M. Healy, M. Hossain, M. Jung, I. Khorosh, G. Kumar, Y.-J. Lee, D. Lewis, T.-W. Lin, C. Liu, S. Panth, M. Pathak, M. Ren, G. Shen, T. Song, D. H. Woo, X. Zhao, J. Kim, H. Choi, G. Loh, H.-H. Lee, and S. K. Lim. 3D-MAPS: 3D massively parallel processor with stacked memory. In *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2012 IEEE International*, pages 188–190, 2012.

[35] N. Kim, D. Wu, and D. e. a. Kim. Interposer design optimization for high frequency signal transmission in passive and active interposer using through silicon via (TSV). In *Proc. of the Electronic Components and Technology Conf.*, pages 1160–1167, 2011.

[36] Y. Kim, T.-D. Han, S.-D. Kim, and S.-B. Yang. An effective memory–processor integrated architecture for computer vision. In *International Conference on Parallel Processing*, 1997.

[37] P. M. Kogge. EXECUBE-a new architecture for scaleable MPPs. In *Proceedings of the 1994 International Conference on Parallel Processing - Volume 01*, pages 77–84, 1994.

[38] S. Lee, J. K. Kim, X. Zheng, Q. Ho, G. A. Gibson, and E. P. Xing. On model parallelization and scheduling strategies for distributed machine learning. In *The Conference on Neural Information Processing Systems*, 2015.

[39] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2009.

[40] S. Li, J. H. Ahn, R. D. Strong, J. B. Brockman, D. M. Tullsen, and N. P. Jouppi. Mcpat: An integrated power, area, and timing modeling framework for multicore and manycore architectures. In *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO 42, pages 469–480, New York, NY, USA, 2009. ACM.

[41] J. Liu, B. Jaiyen, R. Veras, and O. Mutlu. RAIDR: Retention-aware intelligent DRAM refresh. In *Proceedings of the 39th Annual International Symposium on Computer Architecture*, pages 1–12, 2012.

[42] G. H. Loh. 3D-stacked memory architectures for multi-core processors. In *Proc. of the International Symposium on Computer Architecture*, pages 453–464, 2008.

[43] G. H. Loh, N. Jayasena, M. Oskin, M. Nutter, D. Roberts, M. Meswani, D. P. Zhang, and M. Ignatowski. A processing in memory taxonomy and a case for studying fixed-function pim. In *Workshop on Near-Data Processing (WoNDP)*, 2013.

[44] G. H. Loh, N. E. Jerger, A. Kannan, and Y. Eckert. Interconnect-memory challenges for multi-chip, silicon interposer systems. In *Proceedings of the 2015 International Symposium on Memory Systems*, pages 3–10, 2015.

[45] G. L. Loi, B. Agrawal, N. Srivastava, S.-C. Lin, T. Sherwood, and K. Banerjee. A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy. In *Proc. of the Design Automation Conference*, pages 991–996, 2006.

[46] Y. Mao, E. Kohler, and R. T. Morris. Cache craftiness for fast multicore key-value storage. In *Proceedings of the 7th ACM European Conference on Computer Systems*, 2012.

[47] Memcached. Memcached, a distributed memory object caching system, 2014.

[48] Micron. Hybrid memory cube: a re-architected DRAM subsystem. In *Hot Chips*, pages 1–4, 2011.

[49] D. Milojevic, S. Idgunji, D. Jevdjic, E. Ozer, P. Lotfi-Kamran, A. Panteli, A. Prodromou, C. Nicopoulos, D. Hardy, B. Falsari, and Y. Sazeides. Thermal characterization of cloud workloads on a power-efficient server-on-chip. In *IEEE International Conference on Computer Design*, pages 175–182, 2012.

[50] R. C. Murphy, P. M. Kogge, and A. Rodrigues. The characterization of data intensive memory workloads on distributed PIM systems. In *Revised Papers from the Second International Workshop on Intelligent Memory Systems*, pages 85–103, 2001.

[51] L. Nai and H. Kim. Instruction offloading with HMC 2.0 standard: A case study for graph traversals. In *Proceedings of the 2015 International Symposium on Memory Systems*, pages 258–261, 2015.

[52] M. Oskin, F. T. Chong, and T. Sherwood. Active pages: A computation model for intelligent memory. In *Proceedings of the 25th Annual International Symposium on Computer Architecture*, pages 192–203, 1998.

[53] D. Patterson, T. Anderson, N. Cardwell, R. Fromm, K. Keeton, C. Kozyrakis, R. Thomas, and K. Yelick. A case for intelligent RAM. *IEEE Micro*, 17(2):34–44, Mar. 1997.

[54] E. Poyraz, H. Xu, and Y. Cui. Application-specific I/O optimizations on petascale supercomputers. In *Proceedings of the International Conference on Computational Science (ICCS)*, 2014.

[55] S. H. Pugsley, J. Jestes, H. Zhang, R. Balasubramonian, V. Srinivasan, A. Buyuktosunoglu, A. Davis, and F. Li. NDC: Analyzing the impact of 3D-stacked memory+logic devices on MapReduce workloads. In *Proceedings of the International Symposium on Performance Analysis of Systems and Software*, pages 1–11, 2014.

[56] K. Saban. Xilinx stacked silicon interconnect technology delivers breakthrough fpga capacity, bandwidth, and power efficiency. Technical report, Xilinx, Inc., December 2012.

[57] SAP. SAP HANA: an in-memory, column-oriented, relational database management system, 2014.

[58] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Trans. Archit. Code Optim.*, 1(1):94–125, Mar. 2004.

[59] Y. Solihin, J. Lee, and J. Torrellas. Automatic code mapping on an intelligent memory architecture. *IEEE Transactions on Computers*, 2001.

[60] M. Stonebraker and A. Weisberg. The VoltDB main memory dbms. *IEEE Data Eng. Bull.*, 36(2):21–27, 2013.

[61] M. Sunohara, T. Tokunaga, T. Kurihara, and M. Higashi. Silicon interposer with TSVs (through silicon vias) and fine multilayer wiring. In *Proc. of the Electronic Components and Technology Conference*, pages 847–852, 2008.

[62] B. Wang, B. Wang, R. Li, and W. Perrizo. *Big Data Analytics in Bioinformatics and Healthcare*. IGI Global, 1st edition, 2014.

[63] T. Wen. Introduction to the x10 implementation of npb mg. In *IBM Technical Report*, 2006.

[64] W. A. Wulf and S. A. McKee. Hitting the memory wall: Implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24, Mar. 1995.

[65] D. Zhang, N. Jayasena, A. Lyashevsky, J. L. Greathouse, L. Xu, and M. Ignatowski. TOP-PIM: Throughput-oriented programmable processing in memory. In *Proceedings of the*

23rd International Symposium on High-performance Parallel and Distributed Computing, pages 85–98, 2014.

[66] R. Zhang, M. R. Stan, and K. Skadron. Hotspot 6.0: Validation, acceleration and extension. 2015.

[67] J. Zhao, G. Sun, G. Loh, and Y. Xie. Optimizing GPU energy efficiency with 3D die-stacking graphics memory and reconfigurable memory interface. *ACM Transactions on Architecture and Code Optimization (TACO)*, 10(4):24:1–24:25, 2013.

[68] J. Zhao, G. Sun, G. H. Loh, and Y. Xie. Energy-efficient GPU design with reconfigurable in-package graphics memory. In *Proceedings of the 18th International Symposium on Low Power Electronics and Design (ISLPED)*, pages 403–408, 2012.

[69] L. Zheng, Y. Zhang, and M. S. Bakir. A silicon interposer platform utilizing microfluidic cooling for high-performance computing systems. *IEEE Transactions on Components, Packaging and Manufacturing Technology*, 5(10), 2015.