# Exploring Performance, Power, and Temperature Characteristics of 3D Systems with On-Chip DRAM

Jie Meng, Daniel Rossell, and Ayse K. Coskun
*Electrical and Computer Engineering Department, Boston University, Boston, MA, USA*
*{jiemeng, drossell, acoskun}@bu.edu*

***Abstract*—3D integration enables stacking DRAM layers on processor cores within the same chip. On-chip memory has the potential to dramatically improve performance due to lower memory access latency and higher bandwidth. Higher core performance increases power density, requiring a thorough evaluation of the tradeoff between performance and temperature. This paper presents a comprehensive framework for exploring the power, performance, and temperature characteristics of 3D systems with on-chip DRAM. Utilizing this framework, we quantify the performance improvement as well as the power and thermal profiles of parallel workloads running on a 16-core 3D system with on-chip DRAM. The 3D system improves application performance by 72.6% on average in comparison to an equivalent 2D chip with off-chip memory. Power consumption per core increases by up to 32.7%. The increase in peak chip temperature, however, is limited to $1.5^oC$ as the lower power DRAM layers share the heat of the hotter cores. Experimental results show that while DRAM stacking is a promising technique for high-end systems, efficient thermal management strategies are needed in embedded systems with cost or space restrictions to compensate for the lack of efficient cooling.**

## I. Introduction

As the feature sizes of technology nodes shrink and functionality on a microprocessor increases, the communication between last-level caches and main memory becomes the performance bottleneck of today's computer systems. 3D stacking enables integrating DRAM layers and processor cores on the same chip, and therefore has become a promising solution to improve memory bandwidth and to reduce memory access time [1]. In fact, major chip vendors such as Intel and IBM have announced that 3D systems with on-chip DRAM will be available within the next few years.

While 3D systems offer a number of advantages including the opportunities for DRAM stacking, there are unresolved challenges in design and manufacturing, testing, and runtime operation. As 3D stacking increases the chip thermal resistivity, thermal challenges typically accelerate in 3D systems, and high temperatures are among the major concerns. 3D systems with on-chip DRAM have the potential to increase the system performance significantly, which in turn increases power densities. Therefore, performance and temperature tradeoffs should be examined thoroughly to enable high-performance and reliable system operation.

Recently, several research groups have introduced approaches for modeling performance in 3D systems, focusing on a small number of cores (single-core, quad-core) and single-threaded workloads [2], [3], [4]. Thermal hotspots have been a pressing issue due to the cooling costs and reliability challenges in conventional 2D design as well as 3D systems. A number of thermal management techniques exist in the 2D domain for reducing peak temperature and balancing power distribution [5], [6]. Prior work on temperature management of 3D systems includes static optimization methods, such as thermal-aware floorplanning [7], [8], as well as runtime management approaches, such as task migration and dynamic voltage and frequency scaling (DVFS) [9], [10]. However, detailed performance analysis and thermal optimization for 3D systems have been mostly disjoint so far. For example, thermal management policies focusing on 3D systems provide performance estimates based on worst-case scenarios, without providing an architecture-level evaluation [11].

In this paper, we present a comprehensive approach to evaluate performance, power, and temperature for 3D systems with on-chip DRAM. We use the evaluation framework to quantify the benefits and challenges for 3D systems running parallel applications that represent future multicore workloads. Our contributions are:

- We present a model to compute the memory access latency of 3D systems with on-chip DRAM. This model is then used in the architecture-level performance simulation. We integrate the performance simulation with the power and thermal models to enable a thorough evaluation of 3D systems with on-chip DRAM.
- Using the evaluation infrastructure, we evaluate the performance, power, and temperature of a 16-core 3D system with on-chip DRAM. We run the parallel benchmarks in the PARSEC suite [12], and show that instructions per cycle (IPC) for the applications is on average 72.6% higher in comparison to 2D systems. The performance improvement causes an increase in per core power by up to 32.7%.
- We use a thermally-aware thread allocation policy [13] for reducing the temperature in the 3D system. We observe that the peak temperature of the 3D system increases by at most $1.5^oC$ with respect to the 2D peak temperature for a high-end system. In fact, for most of the cases, we observe a slight decrease in peak power, as the DRAM layers have low power density. However, for smaller or lower cost thermal packages as in embedded systems, we demonstrate that efficient thermal management strategies are needed to enable reliable operation.

978-1-4577-1221-0/11/$26.00 ©2011 IEEE

The rest of the paper starts with an overview of the related work. Section III provides the details of the simulation framework integrating performance, power, and thermal models for 3D systems with on-chip DRAM. Section IV explores the target 3D system using the proposed framework, and Section V concludes the paper.

## II. Related Work

In this section, we review the related work on 3D systems research. Performance analysis for 3D systems with on-chip DRAM typically focuses on comparing the system performance between 2D and 3D systems. Loi et al. show that on-chip memory delivers significant amount of speedup for three SPEC benchmarks [3]. Liu et al. report up to 126% speedup for single-core processors with 3D memory stacking [2]. These approaches use architecture-level simulation, but only consider single-core processors and do not evaluate temperature. Loh explores 3D-stacked memory architectures for 4-core processors [4]. They conduct a thermal analysis of 3D-stacked DRAMs using the HotSpot toolset [14]. However, their experiments are also limited to single-threaded benchmarks.

For modeling power consumption in 3D systems, most of the previous work focuses on the power delivery and distribution issues in 3D chips from a circuit-level point of view [15], [16]. Wu et al. use the power density analysis and power delivery consideration in their 3D cost model [17]. However, they do not provide a comprehensive power evaluation for the components on the 3D chips.

A number of prior thermal optimization methods are implemented at design time. For example, Hung et al. present a thermally-aware floorplanner for 3D architectures [7]. Cong et al. propose transformation techniques for 3D IC placement [8]. For dynamic thermal management in 3D systems, Sun et al. propose an optimization algorithm for task assignment and scheduling. Zhu et al. propose thermal management techniques that use task migration and DVFS polices implemented within the OS [9]. These dynamic management methods are effective, however they do not explicitly consider DRAM stacking.

Our research focuses on integrating performance, power, and thermal models for providing a comprehensive evaluation of 3D systems with on-chip DRAM, particularly focusing on future multicore architectures running parallel applications. We use the simulation framework to demonstrate the performance and temperature tradeoffs for a 16-core 3D system. In addition to high-end systems, we investigate the thermal behavior of 3D systems with lower cost or smaller packages, as high-performance multicore systems are making their way into a number of embedded applications.

## III. Methodology

In this section, we present our methodology for exploring 3D systems with on-chip DRAM. We describe our target system and introduce our simulation infrastructure for evaluating performance, power, and temperature.

### A. Target System

In this paper, we use a 16-core processor as our target system. We model both the 2D baseline and the 3D system with a 2-layer stacked DRAM. The architectural configurations for the cores and caches in both 2D and 3D systems are the same. Each core on the 16-core processor has 2-way issue and out-of-order execution. We assume the system is manufactured at 45nm and has a total die area of $128.7mm^2$. The cores operate at 1 GHz and have a supply voltage of 1.14V. Each core has 2 integer and 1 floating point arithmetic/logic units, 1 integer and 1 floating point multiplication/division units, and private 16 KB L1 instruction and data caches. The core architecture is based on the cores used in the Intel 48-core single-chip cloud computer (SCC) [18].

Each core has a 512 KB private L2 cache. All the L2 caches are located on the same layer as the cores and connected by a shared bus. MESI cache coherence protocol is used for communication. The 2D baseline system and the 3D system with on-chip DRAM both have on-chip memory controllers.

TABLE I: Core Architecture Parameters

| Architectural Configuration | |
|---|---|
| **CPU Clock** | 1.0 GHz |
| **Branch Predictor** | tournament predictor |
| **Issue** | out-of-order |
| **Reorder Buffer** | 128 entries |
| **Issue Width** | 2-way |
| **Functional Units** | 2 IntAlu, 1 IntMult,<br>1 FPALU, 1 FPMultDiv |
| **Physical Regs** | 128 Int, 128 FP |
| **Instruction Queue** | 64 entries |
| **L1 ICache** | 16 KB @2 ns (2 cyc) |
| **L1 DCache** | 16 KB @2 ns (2 cyc) |
| **L2 Cache(s)** | 16 private L2 Caches, each L2:<br>4-way set-associative, 64B blocks<br>512 KB @5 ns (5 cyc) |

The layout of the 3D 16-core processor with DRAM-stacking is shown in Fig. 1. We place the processing cores and caches on one layer and stack a 2-layer 3D DRAM below it. We assume face-to-back bonding and through-silicon vias (TSVs) that are etched through the bulk silicon for vertically connecting the layers.

### B. Performance Model for 3D Systems with DRAM Stacking

3D systems with DRAM stacking provide high speed and wide bandwidth for accessing main memory, enabled by utilizing the vertical TSVs. In traditional 2D design, accesses to the off-chip main memory are limited by slow off-chip buses. In addition, off-chip bus width (typically 64 bits) is limited by the I/O pad constraints. In this section, we introduce our approach for modeling 3D DRAM access latency.

To analyze the performance improvement of 3D architectures, we need to have an accurate model for memory latency. The two main components for main memory latency are the data request time spent at the memory controller and the data

TABLE II: DRAM access latency

| | **2D-baseline design** | **3D system with on-chip DRAM** |
|---|---|---|
| **memory controller** | 4 cycles controller-to-core delay, 116 cycles queuing delay, 5 cycles memory controller processing time | 4 cycles controller-to-core delay, 50 cycles queuing delay, 5 cycles memory controller processing time |
| **main memory** | off-chip 1GB SDRAM, 200MHz operating frequency, $t_{RAS} = 40ns$, $t_{RP} = 15ns$, 10ns chipset request/return | on-chip 1GB SDRAM, 800MHz operating frequency, $t_{RAS} = 30ns$, $t_{RP} = 15ns$ [19], no chipset delay |
| **memory bus** | off-chip memory bus, 200MHz, 8Byte bus width | on-chip memory bus, 2GHz, 128Byte bus width |

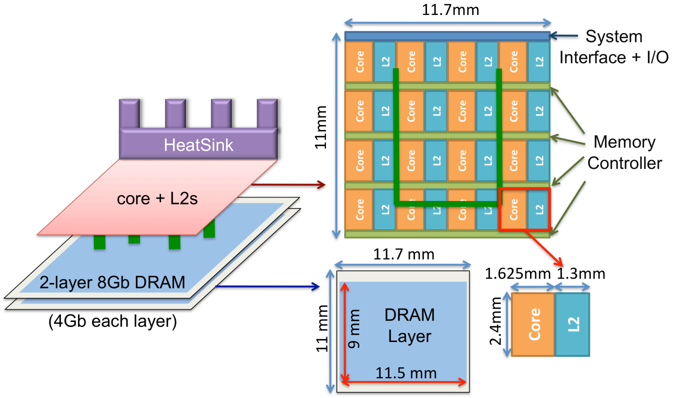

Fig. 1: The layout for 3D systems with on-chip DRAM.

retrieving time spent at the DRAM layers [2]. Memory controller latency includes time needed to translate physical addresses to memory addresses and time for scheduling memory requests. The request scheduling time consists of time spent for converting memory transactions to command sequences and queuing time. We assume the memory controller address translation time is equal to the sum of memory controller processing time and controller-to-core delay [20].

DRAM access latency consists of address decoding time, column and row active time, and data transfer time. Stacking DRAM layers on top of the logic layer makes the data transfer much faster between DRAM and cores. We consider a 1GB DRAM, which consists of two 4Gb layers, and set the row active time $t_{RAS} = 30ns$ and row precharge time $t_{RP} = 15ns$ [19]. To simulate data transfer time, we assume 1024 TSVs, which provide a 128Byte bus width with only 0.3% chip area overhead. Table II summarizes the memory access times for 2D and 3D systems.

### C. *Performance Simulation*

We use the M5 simulator [21] to build the performance simulation infrastructure for our target systems. We use the Alpha instruction set architecture (ISA) as it is the most stable ISA supported in M5. The full-system mode in M5 models a DEC Tsunami system to boot an unmodified Linux 2.6 operating system. We run parallel applications from the PARSEC benchmark suite [12], representing future multicore workloads.

M5 models a split-transaction bus that is configurable in both latency and bandwidth. We model the 3D system with DRAM stacking in M5 by configuring the main memory access latency and the bus width between L2 caches and main memory to mimic the high data transfer bandwidth provided by the TSVs. The configurations for cores and caches are shown in Table I. The configurations for the DRAM and the cache-to-DRAM accesses are based on the analysis in Table II.

We implement thread-binding in M5 for the PARSEC benchmarks to control thread allocation. A thread is bound on a specific core during a time interval and does not move among cores. The default thread-binding policy for the 2D system is in-order assignment, which means thread $i$ is bounded to core $i$ $(1 \leq i \leq 16)$. We use a thermally-aware thread allocation policy for the 3D system, as discussed in Section IV.

We run PARSEC benchmarks in M5 with sim-large input sets and collect the performance statistics at regular time intervals. For each PARSEC benchmark, the start of the region of interest (ROI, i.e., the parallel phase) is pre-defined in the PARSEC hook-libraries. We fast-forward the M5 simulation to the ROI and execute each PARSEC benchmark with the detailed out-of-order CPUs for 1 second (100 time steps, collecting statistics at 10ms intervals). We use the performance statistics collected from M5 simulations as inputs for the processor power model.

We use application IPC [22] as the metric to evaluate the performance. Application IPC is a performance metric for multicore systems running parallel workloads that considers the variations of the execution times of different threads. This metric accumulates all the instructions executed in all threads and divides the total instruction count by the number of cycles for the longest thread, as the longest thread determines the application finish time.

### D. *Power Model*

For modeling the power consumption, we use McPAT 0.7 [23] for 45nm process to obtain the run-time dynamic and leakage power of the cores. The L2 cache power is calculated using Cacti 5.3 [24].

McPAT computes the power consumption by taking the system configuration parameters and M5 performance statistics as inputs. To improve accuracy for run-time power computations, we calibrate the McPAT run-time dynamic power to match the published power of Intel SCC. In our McPAT simulations, we set $V_{dd}$ to 1.14V and operating frequency to 1GHz.

The average total power for a core and its private L2 cache in Intel SCC [18] is 1.83 W. We get the breakdown of the total power using the power results from McPAT and Cacti. We estimate Intel SCC L2 cache power using Cacti, which provides the run-time dynamic power as the total read dynamic power at the maximum frequency. We subtract L2 cache power from the total core and L2 cache power to obtain per core power in the Intel SCC as 1.64W. We assume 35% of the power is due to leakage based on McPAT results and reported leakage values in existing commercial systems at 45nm. Thus, we estimate Intel SCC core's average run-time dynamic power as 1.07W and leakage power as 0.57W. Since our applications are not accessing L2 caches all the time, we scale the dynamic power computed by Cacti for a 512KB cache using the L2 cache access rate collected from M5.

To calibrate the McPAT run-time dynamic core power, we compute the average dynamic core power value from McPAT across all the benchmarks, and obtain the calibration factor, $R$, between the average McPAT dynamic core power and the Intel SCC average dynamic core power. Then, we use the calibration factor $R$ to calibrate each benchmark's dynamic core power consumption. A similar calibration approach has been introduced in prior work [25].

The DRAM power in the 3D system is calculated using MICRON's DRAM power calculator [26], which takes the memory read and write access rates as inputs to compute the power for DRAM. The on-chip memory controller power for both 2D and 3D systems is estimated from Intel SCC as 5.9 W. The system interface and I/O power as well as the on-chip bus power are negligible with respect to the total chip power.

### E. Thermal Model

We use HotSpot 5.0 [14] for thermal simulations. We run simulations for the 2D and 3D systems using the default chip package in HotSpot to represent efficient packages in high-end systems. We simulate two additional packages representing cheaper and smaller packages in embedded systems. Power traces from McPAT are inputs of the thermal model. All simulations use the HotSpot grid model for a higher degree of accuracy and are initialized with the steady-state temperatures. The parameters in HotSpot simulations for 2D and 3D architectures are listed in Table III. We assume the impact of the TSVs to the thermal parameters is negligible, considering TSVs occupy less than 1% of the chip area. The heatsink parameters are changed from the default values to simulate the additional packages as shown in Table IV. We model a medium-cost package, and a system without a heat sink that is estimated by assigning a very small heat sink thickness value.

## IV. Performance, Power, and Temperature Exploration of 3D Systems with DRAM stacking

In this section, we quantify the performance-temperature tradeoffs for the parallel workloads running on the 16-core 3D system. We analyze the temporal performance behavior of selected benchmarks, and also compare the overall performance of the 3D system with on-chip DRAM against the 2D baseline. We also take a detailed look into the power and thermal profiles for the 3D systems, considering the three different thermal packages.

In the evaluations of our 3D target system, the thread allocation is based on the balance_location policy introduced in recent work [13] for 2D systems. As illustrated in Fig. 2, balance_location assigns threads with the highest IPCs to the cores at the coolest locations on the die. The motivation is that, cores located on the corners or sides of the chip are cooler in comparison to the central area of the chip. At each sampling interval, we sort the IPC values for all the threads. Then, we group the four threads with the highest IPC values as $T_{IPC\_I}$ and allocate them to the cores on the corners. In the same way, we group the four threads with lowest IPC values as $T_{IPC\_III}$ and allocate them to the center of the chip. The rest of the threads are marked as $T_{IPC\_II}$, and are allocated on the four sides of the 4x4 grid of cores.

TABLE III: Thermal simulation configuration in HotSpot

| Thermal Parameters | |
|---|---|
| **Parameters** | **2D and 3D** |
| Chip thickness | 0.1mm |
| Silicon thermal conductivity | 100 W/mK |
| Silicon specific heat | 1750 kJ/m$^3$K |
| Sampling interval | 0.01s |
| Spreader thickness | 1mm |
| Spreader thermal conductivity | 400 W/mK |
| **Parameters** | **3D** |
| DRAM thickness | 0.02mm |
| DRAM thermal conductivity | 100 W/mK |
| Interface material thickness | 0.02mm |
| Interface material conductivity | 4 W/mK |

TABLE IV: Heat sink parameters for the three packages

| Heatsink Parameters | | |
|---|---|---|
| **Package** | **Thickness** | **Resistance** |
| High Performance | 6.9mm | 0.1K/W |
| No Heatsink (Embedded A) | 10$\mu$m | 0.1K/W |
| Medium Cost (Embedded B) | 6.9mm | 1 K/W |

We start running an application on our target 3D system using the default in-order thread assignment. Then, we collect performance statistics after each 10ms interval and sort the IPC values across all the threads. We check if any thread is not running on its thermally-preferable location, and migrate threads if needed. The process is repeated periodically until the end of the execution. Assuming a fixed thread migration overhead of 0.01ms [22] every 10ms, performance cost of this approach is very low.

We have observed 1-2$^oC$ peak temperature reduction using the thermally-preferable thread allocation in our 3D system in comparison to in-order thread assignment. A limited temperature reduction is expected for running balance_location on fully utilized systems where power variation among the threads is limited [13].

### A. Performance Evaluation

The temporal performance behavior of two PARSEC benchmarks (fluidanimate and streamcluster) are shown in Fig. 3. We observe that for both 2D and 3D architectures, the IPC of fluidanimate changes periodically while the IPC of streamcluster is stable during simulation time. In addition, running on our target 3D system with DRAM stacking,

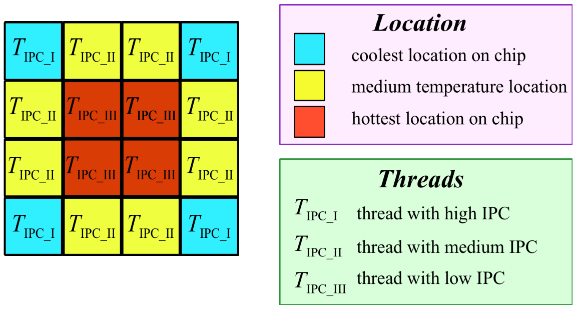
$T_{IPC\_I}$ $T_{IPC\_II}$ $T_{IPC\_II}$ $T_{IPC\_I}$
$T_{IPC\_II}$ $T_{IPC\_III}$ $T_{IPC\_III}$ $T_{IPC\_II}$
$T_{IPC\_II}$ $T_{IPC\_III}$ $T_{IPC\_III}$ $T_{IPC\_II}$
$T_{IPC\_I}$ $T_{IPC\_II}$ $T_{IPC\_II}$ $T_{IPC\_I}$
Location
coolest location on chip
medium temperature location
hottest location on chip
Threads
$T_{IPC\_I}$ thread with high IPC
$T_{IPC\_II}$ thread with medium IPC
$T_{IPC\_III}$ thread with low IPC

Fig. 2: Illustration of the balance-location policy's thermally-preferable thread allocation.

`streamcluster` and `fluidanimate` improve their IPC by 211.8% and 49.8%, respectively, in comparison to the 2D baseline situation. The reason for such a difference on IPC improvements is that, in comparison to fluidanimate, streamcluster has a significantly higher number of main memory accesses (or L2 misses).

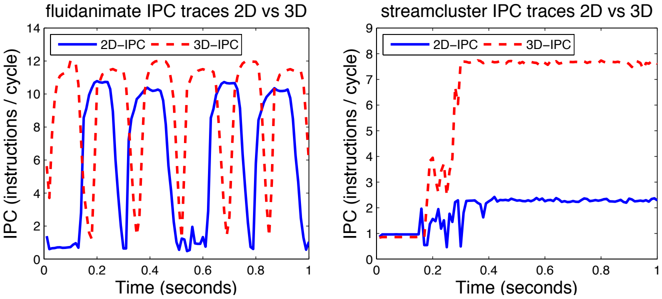

Fig. 3: IPC temporal behavior analysis for 2D-baseline versus 3D-DRAM systems for fluidanimate and streamcluster.

Fig. 4 presents the IPC improvements for the 3D system with DRAM stacking in comparison to the 2D-baseline. By using 3D DRAM stacking, we achieve an average IPC improvement of 72.55% across all the 9 parallel benchmarks in the PARSEC suite in comparison to the 2D system with off-chip DRAM. `streamcluster`, `vips`, and `canneal` achieve higher IPC improvements (over 100%), as these applications are highly memory-bound and therefore benefit more significantly from the reduction in memory access latency.

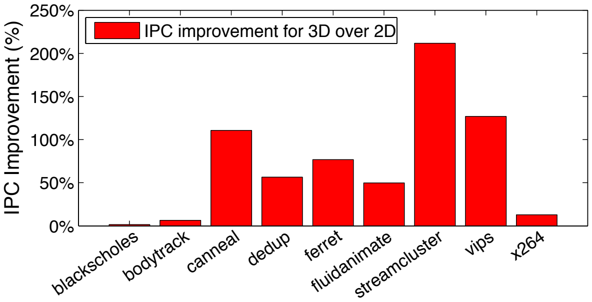

Fig. 4: Percentage of IPC improvement for 3D DRAM stacking architecture over 2D-baseline structure.

### B. Power Evaluation

The core power increase for the 3D system with DRAM stacking with respect to the 2D-baseline is presented in Fig. 5. Power consumption increases by 16.6% on average for the 3D system across the benchmark set. `ferret` has the largest increase in core power, as it is already at a higher power range and has an additional 76.8% increase in IPC.

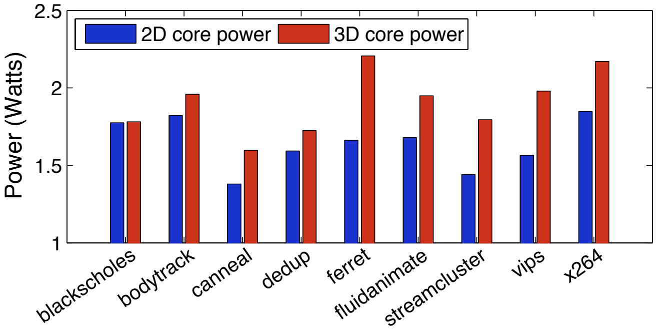

Fig. 5: Average core power for the 3D-DRAM system and the 2D-baseline.

For analyzing the power and temperature behavior of a stacked DRAM layer, we present `dedup`'s per-layer DRAM power and temperature traces in Fig. 6. We see that DRAM power changes within the 1s time interval following the changes in memory access rate. Temperature changes owing to the DRAM power variations as well as to the core power variations on the adjacent layer. Note that we assume a uniform power value for the DRAM. Temperature variations on the DRAM are expected to be larger when memory access patterns are explicitly modeled.

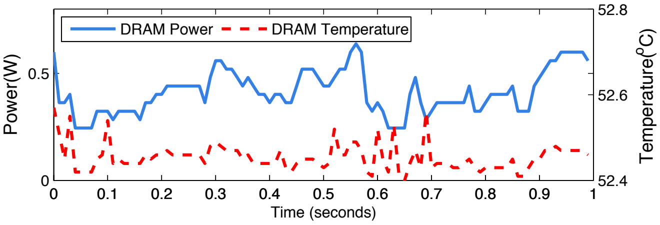

Fig. 6: DRAM power and temperature traces (for one DRAM layer) of dedup running on the 3D system.

### C. Temperature Analysis

We illustrate the thermal behavior for 3D systems using three packages: HotSpot default package, embedded system package A with no heatsink, and embedded system package B which is with medium cost. The peak core temperature in the 2D and 3D systems with the high-performance package is shown in Fig. 7. In such systems, DRAM stacking causes limited temperature rise. The maximum peak temperature increase is $1.52^oC$ for `streamcluster`. In fact, most of the benchmarks running on our target 3D system obtain a

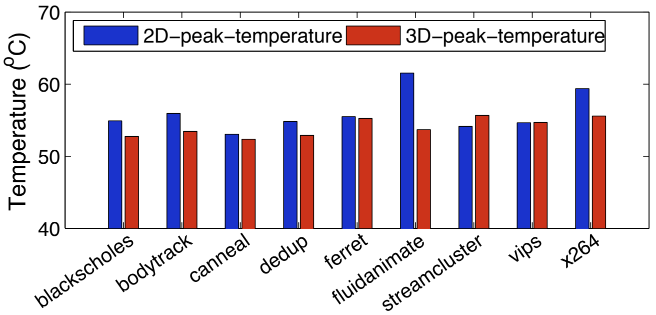

Fig. 7: Peak core temperature for 2D-baseline and 3D stacked DRAM architectures with the default HotSpot package.

peak temperature decrease because the lower power DRAM layer shares the heat of the hotter cores.

The thermal behavior for 2D and 3D systems with the embedded packages are shown in Fig. 8. We notice that the peak temperatures increase more noticeably in the 3D systems with embedded packages. For example, `ferret`'s peak temperature increases by $3.31^oC$ and $8.04^oC$ for the two embedded system packages in comparison to its peak temperature on the 2D system. Efficient thermal management and low-power design techniques are needed to ensure reliable operation for 3D systems with lower cost or smaller packages.

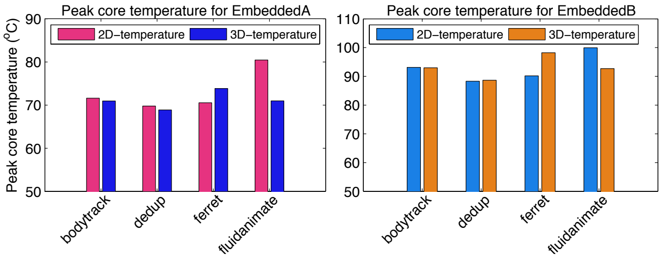

Fig. 8: Peak core temperatures for the 2D-baseline and the 3D stacked DRAM architectures with the embedded packages.

## V. Conclusion

In this paper, we have presented a comprehensive simulation framework for 3D systems with on-chip DRAM. We have explored the performance, power, and temperature characteristics of a 16-core 3D system running the PARSEC parallel benchmark suite. Our results show an average of 72.6% application-IPC improvement and 16.6% average per-core power increase in the 3D system in comparison to the equivalent 2D system. Our thermal analysis demonstrates limited temperature changes in the 3D systems with DRAM stacking with respect to the 2D baseline, owing to the low power density of the DRAM layer. We have also shown the significance of the package choice in sustaining safe temperatures in 3D systems. Our future work includes detailing the DRAM power and temperature models as well as developing efficient thermal management approaches for 3D systems with on-chip DRAM.

## Acknowledgements

The authors thank Dr. Eren Kursun and Thomas Brunschwiler at IBM for their valuable feedback during this project.

## References

[1] B. Black *et al.*, "Die stacking (3D) microarchitecture," in *The 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, December 2006, pp. 469 – 479.

[2] C. Liu, I. Ganusov, M. Burtscher, and S. Tiwari, "Bridging the processor-memory performance gap with 3D IC technology," *Design Test of Computers, IEEE*, vol. 22, no. 6, pp. 556 – 564, November 2005.

[3] G. 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 *The 43rd ACM/IEEE Design Automation Conference (DAC)*, July 2006, pp. 991 – 996.

[4] G. Loh, "3D-stacked memory architectures for multi-core processors," in *The 35th International Symposium on Computer Architecture (ISCA)*, June 2008, pp. 453 – 464.

[5] J. Sartori and R. Kumar, "Distributed peak power management for many-core architectures," in *Design, Automation, and Test in Europe (DATE) Conference.*, April 2009, pp. 1556 – 1559.

[6] Y. Ge, P. Malani, and Q. Qiu, "Distributed task migration for thermal management in many-core systems," in *The 47th ACM/IEEE Design Automation Conference (DAC)*, June 2010, pp. 579 – 584.

[7] W.-L. Hung *et al.*, "Interconnect and thermal-aware floorplanning for 3D microprocessors," in *The 7th International Symposium on Quality Electronic Design (ISQED)*, March 2006.

[8] J. Cong, G. Luo, J. Wei, and Y. Zhang, "Thermal-aware 3D IC placement via transformation," in *The Asia and South Pacific Design Automation Conference (ASP-DAC)*, January 2007, pp. 780 – 785.

[9] C. Zhu *et al.*, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 8, pp. 1479 – 1492, August 2008.

[10] A. K. Coskun, T. S. Rosing, J. Ayala, and D. Atienza, "Modeling and dynamic management of 3D multicore systems with liquid cooling," in *Proceedings of the IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, October 2009.

[11] A. Coskun, D. Atienza, T. Rosing, T. Brunschwiler, and B. Michel, "Energy-efficient variable-flow liquid cooling in 3D stacked architectures," in *Design, Automation, and Test in Europe (DATE) Conference*, March 2010, pp. 111 – 116.

[12] C. Bienia, "Benchmarking modern multiprocessors," Ph.D. dissertation, Princeton University, January 2011.

[13] A. K. Coskun, R. Strong, D. M. Tullsen, and T. Simunic Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," in *SIGMETRICS*. ACM, 2009, pp. 169–180.

[14] K. Skadron, M. R. Stan, W. Huang, V. Sivakumar, S. Karthik, and D. Tarjan, "Temperature-aware microarchitecture," in *Proceedings of Annual IEEE International Symposium on Computer Architecture (ISCA)*, 2003.

[15] N. Khan, S. Alam, and S. Hassoun, "Power delivery design for 3D ICs using different through-silicon via (tsv) technologies," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 4, pp. 647 – 658, April 2011.

[16] A. Shayan *et al.*, "3D power distribution network co-design for nanoscale stacked silicon ICs," in *The IEEE Electrical Performance of Electronic Packaging Conference (IEEE-EPEP)*, October 2008, pp. 11 – 14.

[17] X. Wu *et al.*, "Cost-driven 3D integration with interconnect layers," in *47th ACM/IEEE Design Automation Conference (DAC)*, June 2010, pp. 150 – 155.

[18] J. Howard *et al.*, "A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS," in *International Solid-State Circuits Conference (ISSCC)*, February 2010, pp. 108 –109.

[19] Micron Technology, Inc. DRAM component datasheet. [Online]. Available: http://www.micron.com

[20] X. Dong, Y. Xie, N. Muralimanohar, and N. Jouppi, "Simple but effective heterogeneous main memory with on-chip memory controller support," in *International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2010, pp. 1 – 11.

[21] N. Binkert, R. Dreslinski, L. Hsu, K. Lim, A. Saidi, and S. Reinhardt, "The M5 simulator: Modeling networked systems," *IEEE Micro*, vol. 26, no. 4, pp. 52 –60, July 2006.

[22] J. Meng, C. Chen, A. K. Coskun, and A. Joshi, "Run-time energy management of manycore systems through reconfigurable interconnects," in *Proceedings of ACM Great Lakes Symposium on VLSI (GLSVLSI)*, May 2011.

[23] S. Li, J. H. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *The 39th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2009, pp. 469 –480.

[24] S. Thoziyoor, N. Muralimanohar, J. H. Ahn, and N. P. Jouppi, "CACTI 5.1," HP Laboratories, Palo Alto, Tech. Rep., April 2008.

[25] R. Kumar *et al.*, "Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction," in *The 36th Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, 2003, pp. 81 – 92.

[26] Micron Technology, Inc. DRAM power calculations. [Online]. Available: http://www.micron.com