Optimizing Energy Efficiency of 3-D Multicore Systems* with Stacked DRAM under Power and Thermal Constraints

Jie Meng Katsutoshi Kawakami Ayse K. Coskun
Electrical and Computer Engineering Department, Boston University, Boston, MA, USA
{jiemeng, kkawakam, acoskun}@bu.edu

ABSTRACT

3D multicore systems with stacked DRAM have the potential to boost system performance significantly; however, this performance increase may cause 3D systems to exceed the power budget or create thermal hot spots. This paper introduces a framework to model on-chip DRAM accesses and analyzes performance, power, and temperature tradeoffs of 3D systems. We propose a runtime optimization policy to maximize performance while maintaining power and thermal constraints. Our policy dynamically monitors workload behavior and selects among *low-power* and *turbo* operating modes accordingly. Experiments with multithreaded workloads demonstrate up to 49% energy efficiency improvements compared to existing thermal management policies.

Categories and Subject Descriptors

C.4 [**Performance of System**]: Modeling techniques

General Terms

Design, Experimentation, Management, Performance

Keywords

energy efficiency, thermal management, 3D multicore system

1. INTRODUCTION

3D stacking is a promising technique to increase transistor density per footprint without scaling the technology node, and it also enables stacking different technologies into a single chip. Using 3D stacking, it is possible to place a sizable DRAM layer within the chip, reducing the delays associated with accessing off-chip memory [1, 2]. On the other hand, 3D systems exacerbate the already existing thermal challenges because of higher thermal resistivities and power densities per chip footprint. Thermal hot spots and large temporal or spatial temperature variations adversely affect reliability and performance while increasing the cooling costs [3]. In addition to the temperature rise on the logic layers, temperature of the DRAM layers substantially increases because of the high memory access rate and the heat transfer from the logic layer. High DRAM temperatures severely affect memory reliability and performance [4, 5].

*This work has been funded by DAC A. Richard Newton Scholarship and NSF CAREER grant #1149703.

DAC 2012, June 3-7, 2012, San Francisco, California, USA.

The thermal challenges in 3D systems require a joint assessment of performance, energy, and temperature tradeoffs. Also, as workload dynamics change during the lifetime of a system, it is imperative to have runtime optimization techniques that monitor and actively manage the interplay among performance, power, and temperature of 3D systems. Prior energy and temperature management methods for 3D systems include workload scheduling, dynamic voltage-frequency scaling (DVFS), thermally-aware floorplanning, and job allocation (e.g., [6, 7, 8, 3]). However, these techniques do not jointly evaluate and optimize performance, power, and temperature profiles at run-time for logic and DRAM layers in 3D systems simultaneously.

This paper focuses on optimizing the energy efficiency and temperature of 3D multicore systems with on-chip DRAM. We first model the performance, power, and thermal impacts of the on-chip DRAM and analyze how the reduced memory access latency changes runtime dynamics. We then propose a novel optimization technique that dynamically monitors application behavior through performance counters and adjusts the operating points for adapting to varying application phases. Our policy selects among *low-power* and *high-performance*, or *"turbo"*, execution modes from available voltage-frequency (V-F) settings by utilizing predictions from a regression-based model. In this way, we maximize throughput while maintaining the power and temperature constraints.

The optimization policy is motivated by two observations derived from our analysis of 3D systems with on-chip DRAM. First, we observe that memory-bound benchmarks have significant performance improvements when running on 3D systems with on-chip DRAM compared to the 2D baseline with off-chip memory. However, power and temperatures on both logic and DRAM layers rise significantly. In this case, our policy selects a *low-power* V-F setting to maximize throughput under power and thermal constraints. Second, for CPU-bound benchmarks, we observe limited performance improvement compared to the 2D baseline. However, for CPU-bound applications, stacking the DRAM layer with the logic layer provides a *temperature slack* as the DRAM layer is much cooler than the logic layer and helps maintain low temperature. In this case, we boost system performance using high-frequency *turbo* modes without creating thermal problems. Our specific contributions are as follows:

- We design a simulation framework to model the on-chip DRAM accesses and jointly analyze performance, power, and temperature for both logic and memory layers on 3D systems with stacked DRAM. Using the framework, we analyze on-chip DRAM accesses at various bandwidths. Enabling parallel access to the DRAM improves performance by up to 86.9% compared to single-bus access.

- We propose a novel runtime optimization policy for selecting V-F settings to maximize system performance subject to power and thermal constraints. Our experiments demonstrate that our policy achieves an average performance improvement of 36.1% for a 16-core 3D system with stacked DRAM compared to a statically optimized 3D system with fixed V-F settings. We reduce the energy-delay product (EDP) by up to 49.4% compared to a 3D system managed by a temperature-triggered DVFS policy.

The rest of the paper starts with an overview of the related work. Section 3 introduces the experimental methodology. Sections 4 and 5 propose the runtime optimization policy and present the experimental results, respectively. Section 6 concludes the paper.

2. RELATED WORK

Recent literature has studied the performance and energy benefits of 3D systems with on-chip DRAM. However, most prior work considers the evaluation of performance, power, and temperature separately. Loi et al. analyze 3D system performance with thermal considerations using a standard heat flow model [9], and Loh explores 3D-stacked memory architectures [2] with temperature analysis using HotSpot [10]. However, their thermal simulations are based on coarse-grained power estimates instead of using power traces obtained from detailed performance statistics.

Prior research on 3D system energy and thermal management includes design-time optimization methods and runtime management polices based on task scheduling and DVFS techniques. Cong et al. introduce a thermally-aware 3D placement approach based on transformation techniques [8]. Healy et al. propose a microarchitectural floorplanning algorithm for 3D ICs using linear programming and simulated annealing [11]. These static optimization methods are implemented at design stage, and do not address dynamic changes in workload profiles.

Dynamic power management on multicore 2D systems has been well studied. Isci et al. present a runtime phase prediction methodology to control DVFS based on frequency of memory operations [12]. Cochran et al. propose a scalable method for determining the optimal V-F settings under power constraints [13]. For dynamic management on 3D systems, Zhu et al. propose a runtime thermal management approach using task migration and DVFS [6]. Zhou et al. introduce an OS-level scheduling algorithm for optimizing 3D system temperature using dynamic workload scheduling [14]. These methods targeting 3D systems, however, do not consider detailed performance analysis of the workloads.

Our research differentiates from prior work as we provide a modeling and management methodology to jointly analyze and optimize performance, power, and temperature for 3D systems with on-chip DRAM. We analyze the performance impact of 3D stacked DRAM for single-bus or parallel access scenarios, and design a detailed on-chip DRAM performance and power model. We then propose a runtime optimization method that selects low-power or turbo operating modes based on processor and DRAM utilization in the 3D stack.

3. METHODOLOGY

Our research targets 3D multicore processors with on-chip DRAM. Figure 1 provides an illustration of the logic layer of a 16-core 3D system with stacked DRAM. In the 3D system, all the processing cores and caches are on one layer and a 3D DRAM layer is stacked below it. Through-silicon vias (TSVs) are used for vertically connecting the core and DRAM layers. We assume face-to-back, wafer-to-wafer bonding for building the 3D systems. Wafer-to-wafer bonding allows for reliably manufacturing larger 3D systems. Both the target 3D system with on-chip DRAM and the 2D baseline with off-chip memory have the same core architecture and the same floorplan for the logic layer. The core architecture of the target system is modeled based on the AMD Family 10h microarchitecture used in AMD Magny-Cours processors. Each core has multiple-issue and out-of-order execution. The architectural parameters for cores and caches are listed in Appendix $S2$. We assume the target processor is manufactured with 45nm technology, has a total die area of $376mm^2$, and can be operated under five different V-F settings, as listed in Table 2.

3.1 Modeling Memory Accesses

In order to accurately quantify the performance improvements of our target 3D systems, we model the memory access latency by examining the different components that contribute to the latency. For multicore systems, there are three main components of the memory access latency from the last-level caches to main memory: the propagation delay between last-level caches to memory controller (LLC-to-controller delay), the data request time spent at the memory controller (memory controller processing latency), and the data retrieval time spent at the DRAM.

To model the LLC-to-memory controller delay, we assume that all the private L2 caches are connected to the memory controllers through a shared bus. Figure 1 illustrates the physical layout of the logic layer. We assume that the global bus interconnect is routed around the chip in a serpentine fashion. For modeling the bus interconnect, we use energy-optimized repeater-inserted pipelined channels to reduce the global wire delay. The wire propagation delay is linear with respect to the wire length, owing to the repeaters that are inserted to partition the wire into smaller segments. Each pipeline stage is designed using predictive technology model for 45nm and has a propagation delay of 183ps per mm. We estimate the average distance from an L2 cache to a memory controller block as 9.4mm based on the layout. Thus, the round trip LLC-to-memory controller latency is 4ns (rounded up).

Memory access latency is strongly governed by the memory controller processing time. Modern memory controllers typically consist of a memory request queue that buffers the pending requests waiting to get scheduled, and a scheduler that selects the next request to be serviced. The memory controller processing latency, thus, refers to the time spent

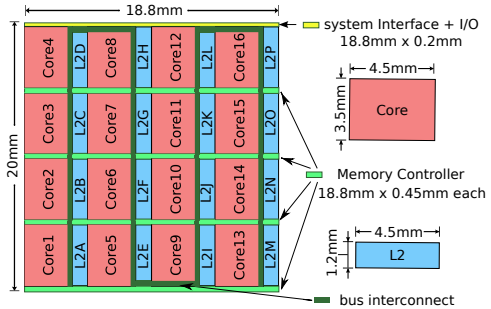

Figure 1: The layout for the logic layer of target 3D system.

Table 1: DRAM access latency

	2D-baseline design	3D system with regular memory access
memory controller	4ns LLC-to-controller delay, 48ns memory controller processing time	4ns LLC-to-controller delay, 24ns memory controller processing time
main memory	off-chip DRAM, $t_{RAS} = 36ns$, $t_{RP} = 15ns$	on-chip DRAM, $t_{RAS} = 36ns$, $t_{RP} = 15ns$
memory bus	off-chip bus, 200MHz, 8-Byte bus width	on-chip bus, 2GHz, 64-Byte bus width
total delay	103ns	79ns

by a memory request waiting to get scheduled. We set the overall memory controller processing latency as 100 cycles from the simulation results reported in prior work [15], where the memory controller latency of a 16-core processor with 4 memory controllers running PARSEC benchmark suite is studied. For the 3D system, we assume that the memory controller latency is reduced by 50% [16].

We use the same DRAM structure for the off-chip DRAM in 2D baseline and for the DRAM layer in 3D system, where we consider a 1GB DRAM consisting of 4 ranks, each of which has 4 banks (16 banks total). We use the row active time $t_{RAS} = 36ns$ and row precharge time $t_{RP} = 15ns$ as reported by MICRON's DDR3 SDRAM. We use the same timing parameters for the DRAM layer of the target 3D system, which is consistent with the assumptions used in earlier studies [2, 9]. Table 1 summarizes the memory access times for 2D and 3D systems.

To simulate the data transfer between logic layer and DRAM layer, we consider `regular memory access` and `parallel memory access`, both with a fast memory bus at 2GHz. As illustrated in Figure 2, the `parallel memory access` allows the four on-chip memory controllers to access the four DRAM ranks at the same time. We deploy 512 TSVs on each memory controller, which provide a 64-Byte bus width for each memory controller with only 0.2% chip area overhead. From our simulation results for the NAS and PARSEC benchmarks, we observe the accesses to the main memory are evenly distributed among the four ranks, as shown in Appendix $S3$. Therefore, we assume the memory access latency with `parallel access` is one-fourth of the latency of `regular access`. Note that this is a conservative assumption as the simultaneous accesses also enable faster processing at the memory controller because of fewer pending requests in the request queues.

3.2 Performance Simulation

We use M5 full-system simulator [17] to conduct the performance simulation for our target systems. We use the Alpha instruction set architecture (ISA) as it is the most stable ISA currently supported in M5. The full-system mode in M5 models a DEC Tsunami system to boot Linux OS. We model the 3D system with on-chip DRAM in M5 by configuring the main memory access latency and the bus width/speed between L2 caches and main memory to mimic the high data transfer bandwidth provided by the TSVs. The architectural configuration parameters and memory access latencies are shown in Appendix $S2$ and in Table 1, respectively.

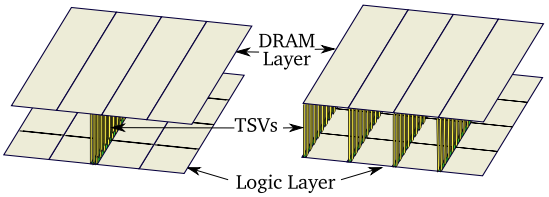

Figure 2: An illustration of the 3D system with DRAM stacking that has (a) `regular memory access` and (b) `parallel memory access`.

We select parallel applications from the PARSEC [18] and NAS Parallel Benchmarks (NPB) suite [19] as our target workloads. We run PARSEC benchmarks in M5 with simlarge input sets and NAS with class B problem sets. For each benchmark, we fast-forward to get past the serial initialization phase. Then, we execute each benchmark with the out-of-order CPUs with detailed memory access simulations, and collect statistics at every 100 million instructions for 100 sampling steps. In order to collect the access statistics for the 3D stacked DRAM, we track the memory accesses to each DRAM bank by observing the least significant bits for the physical memory addresses at every interval. The performance statistics collected from M5 simulations are used as inputs for the processor and DRAM power models.

3.3 Power Model

We use McPAT 0.7 [20] for 45nm process to obtain the run-time dynamic power of the cores. In our McPAT simulations for 2D baseline, we set V_{dd} to 1.1V and operating frequency to 2.1GHz. For our target 3D system, we use five V-F settings as shown in Table 2 (see Appendix $S4$ for average core power results). The L2 cache power is calculated using Cacti 5.3 [21], where the dynamic L2 power is scaled using L2 access rates. The average L2 cache power is 0.62W.

We calibrate the McPAT run-time dynamic core power using measurements that we collect on an AMD Magny-Cours processor. We derive the average dynamic core power values from power simulation across the benchmark suite, and compute the calibration factor, R, to translate the McPAT raw data to the target power scale. Then, we use R to scale dynamic core power consumption. A similar calibration approach has been introduced in prior work [22]. At nominal temperature of 343K, we assume the leakage power for the cores is 35% of the total core power, which matches the measurements on the AMD Mangy-Cours system. We also take the temperature and voltage impact on leakage power into account. The impact of temperature on leakage power is exponentially dependent on the temperature [23]. Prior work shows close-to linear relation between V_{dd} and leakage when variation of V_{dd} is small [24]. As voltage change is limited to 10% of the default setting in our system, we model leakage dependence on V_{dd} as linear.

The DRAM power in the 3D system is calculated using MICRON's DRAM power calculator [25], which takes the memory read and write access rates as inputs. We obtain detailed DRAM power traces for each of the DRAM banks at every sampling interval. The average on-chip DRAM bank power across all the benchmarks in 3D system with parallel access is 1.44W. The on-chip memory controller power for both 2D and 3D systems is estimated based on Intel's 48-core single-chip cloud computer as 5.9W [26]. We assume the system interface and I/O power as well as the on-chip bus

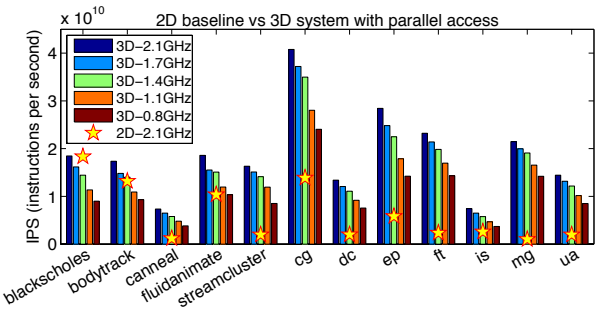

Figure 3: IPS for PARSEC and NAS benchmarks running on 2D baseline and the 3D system with parallel access.

power are negligible with respect to the total chip power. It has been shown that the total on-chip bus power for running PARSEC and NAS workloads is less than 2.0W even for a 64-core system [27].

3.4 Thermal Model

We use HotSpot 5.0 [10] for thermal simulations. We extend HotSpot (see Appendix $S1$) to account for the TSVs in 3D systems by utilizing the methodology for modeling the interlayer material heterogeneity introduced in prior work [3]. Appendix $S5$ provides the thermal parameters for the HotSpot simulations.

Our simulation framework is able to periodically sample runtime events at every fixed time interval or at a certain number of instructions. In this way, we observe the dynamically changing performance patterns that cannot be captured by average or coarse-grained performance estimates. For each benchmark and for each V-F setting, we record the power and performance data at every 100 million instructions into a database. Thermal simulator polls this database to gather power traces based on a fixed or dynamically set V-F setting, as determined by the policy running. Instruction-based intervals are converted to time-based samples as required by thermal simulations. This approach enables decoupling thermal simulation from lengthy performance-power simulations and achieves significant speedups. Even when applying DVFS policies, we are able to maintain accuracy due to two main reasons: (1) we change V-F settings of all the cores together; (2) for all the benchmarks in our evaluation set, the distribution of executed instructions among all the cores is very similar when running at different V-F settings, which allows the runtime V-F setting changes in our modeling methodology.

4. RUN-TIME OPTIMIZATION POLICY

The goal of our runtime optimization policy is to select operating points maximizing performance while maintaining the power and temperature constrains for both logic and DRAM layers. Our optimization policy is motivated by the observations of running PARSEC and NAS benchmarks on our simulation framework under different V-F settings. Figures 3, 4, and 5 present the performance, temperature, and power results of the 2D baseline and the target 3D system with stacked DRAM, respectively.

We notice that, for most of the benchmarks, the average IPS of 3D systems running at 0.8GHz are sufficiently high to match the performance of the 2D baseline. We also observe that applications dramatically differ in their performance behavior. For the memory-intensive benchmarks, such as *streamcluster* and *mg*, the high memory access rates result in significant performance improvements when running on 3D systems with stacked DRAM in comparison to 2D baseline; however, the peak temperature also considerably increases. Thus, we run such benchmarks at the *low-power* mode by exploiting the *performance slack*. Figure 3 shows that, even at *low-power* mode, the memory-intensive benchmarks running on the 3D system still have significant performance improvements in comparison to running on 2D baseline. For CPU-intensive workloads, on the other hand, the low memory access rate result in a cooler DRAM layer that shares the temperature of the hotter core layer. For benchmarks such as *blackscholes*, we switch to the *turbo* mode with higher V-F settings for boosting the performance by taking advantage of the *temperature slack*.

The basic concept of our optimization method is presented in Equation (1), where (F, V) is the set of available V-F settings. Our goal is to maximize throughput (instructions per second, IPS) under power and thermal constraints. P_{cap} is the power budget of the target system, and T_{thld} is the peak temperature threshold to ensure reliable operation. As shown in Figure 4, we set T_{thld} at 85^oC. Figure 5 shows three P_{cap} settings. Our policy satisfies T_{thld} and P_{cap} at the same time. For example, at a loose P_{cap} of 200W, T_{thld} at 85^oC dominates the optimization decisions. A more strict P_{cap} at 175W or 155W requires taking peak power into account. Peak power management is an increasingly important feature owing to power supply limitations and potential energy cost reduction opportunities at large computer clusters.

$$\begin{aligned} &\underset{(f,v)\in(F,V)}{\text{maximize}} \quad IPS(f,v) \\ &\text{subject to} \quad power(f,v) \le P_{cap},\ temperature(f,v) \le T_{thld}. \end{aligned} \tag{1}$$

Our runtime optimization policy is illustrated in Figure 6. We start running the application with the lowest V-F setting to ensure reliable operation, and collect the performance statistics at regular intervals of 100 million instructions. Based on a model we construct offline, we predict the highest V-F setting satisfying the constraints using the performance statistics as inputs. We continue running the application with the predicted V-F setting. This process is repeated at every interval.

We choose instructions per cycle (IPC) and memory access per instruction (MA) to construct a regression-based model for selecting the V-F settings. This is because IPC is a good indicator of the power of the logic layer and MA is a good indicator of the power of the DRAM layer. Power densities on both layers affect chip peak temperature on the 3D system. Our V-F prediction model is in the form of $VF = c_0 + c_1 \cdot MA + c_2 \cdot IPC + c_3 \cdot MA * IPC$. We train

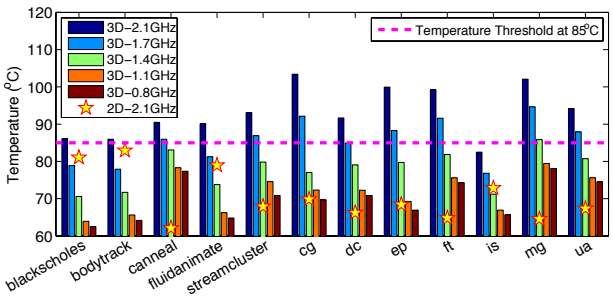

Figure 4: Peak chip temperature on the 3D system with parallel access running at different V-F settings.

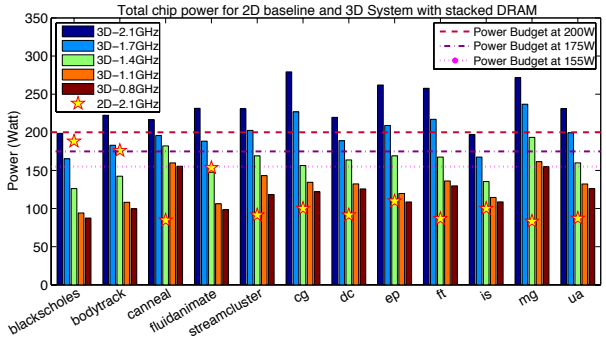

Figure 5: Total chip power on the 3D system with parallel access running at different V-F settings.

the regression model with power and performance statistics from simulations across all benchmarks. Note that we need to use different coefficients in the model depending on the current V-F setting, as MA and IPC vary with the V-F setting. As an example of the V-F prediction for a 3D system with 85^oC/175W constraints, we list the coefficients of the regression-based model for all the V-F settings in Table 2. The regression model provides accurate prediction as shown in Figure 10, and can be refined at runtime if needed. The overhead of the run-time prediction is negligible, since computing a simple equation at every interval has very low computational cost.

Table 2: Regression coefficients for a target 3D system with 85^oC/175W constraints for all the V-F settings.

V-F setting	c_0	c_1	c_2	c_3
2.1GHz/1.1V	3.68	-147.95	-0.059	0.19
1.7GHz/1.06V	3.74	-141.77	-0.071	0.23
1.4GHz/1.02V	3.76	-145.71	-0.075	0.36
1.1GHz/1.0V	3.80	-147.08	-0.087	0.41
0.8GHz/0.98V	3.87	-152.01	-0.072	0.58

5. EXPERIMENTAL RESULTS

This section evaluates our technique on 3D systems with parallel access, and compares our optimization policy against using static V-F settings, a temperature-triggered DVFS policy, and a DVFS policy guided by memory accesses.

Figure 7 demonstrates the performance improvement of the 3D system with parallel on-chip DRAM access running at 2.1GHz and 0.8GHz. We show that enabling parallel access to the 3D DRAM layer improves IPS by up to 86.9% compared to using regular access. `streamcluster` and `mg` show higher IPS improvements than the other benchmarks, since they have higher memory access rates and thus benefit more from reduced average memory access time.

Table 3 presents the performance and energy-efficiency improvements for 3D systems running our runtime optimization policy compared to using static V-F settings. We notice that the peak temperatures go over the thermal constraint

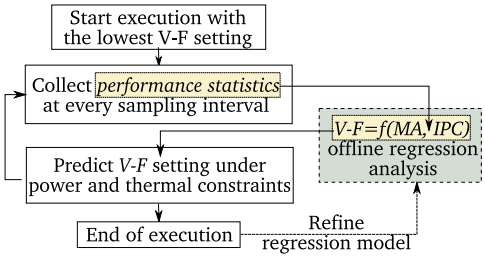

Figure 6: The flowchart of our runtime optimization policy.

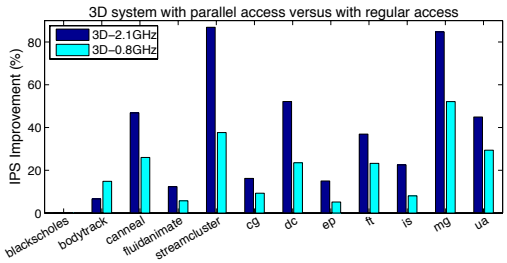

Figure 7: Performance improvement on 3D system with parallel access compared to 3D system with regular access.

of 85^oC for applications running on the 3D systems with frequency settings higher than 1.1GHz. With a loose power constraint of 200W, we compare our policy with the static V-F setting at 1.1GHz/1.0V which maintains temperature below 85^oC for all the benchmarks. Our policy achieves an average IPS improvement of 24.6% and EDP reduction of 22.4% across all the benchmarks.

We present the runtime V-F selection process of our optimization policy in Figure 8. For *ua*, 1.4GHz/1.02V is the reliable static operating point, maintaining the temperature below 85^oC. However, the phase change of *ua* creates a temperature slack periodically. Our policy takes advantage of the temperature slack and switches to 1.7GHz during periods of low temperature.

We demonstrate the advantage of our runtime optimization policy over applying temperature-triggered DVFS in Figure 9. Temperature-triggered DVFS is a well-known policy for thermal management on 2D systems [10, 28]. It tracks chip peak temperature and selects the operating point based on temperature sensor readings. For safe operation while maintaining system performance, we choose two temperature thresholds as 80^oC and 70^oC. Temperature-triggered DVFS reduces/increases the V-F setting when temperature goes above/below 80^oC/70^oC. Our policy improves EDP by up to 61.9% and IPS by 32.2% on average across all the benchmarks in comparison to the temperature-triggered DVFS policy. The performance of *blackscholes* and *is* does not differ between our policy and the temperature-triggered DVFS policy. This is because they have low temperature while running at 2.1GHz/1.1V. The benchmarks that have high temperatures when running on 3D systems with stacked DRAM, such as *streamcluster*, show larger performance improvement using our runtime policy. Our policy selects the highest V-F settings to operate at safe temperatures, while temperature-triggered DVFS may oscillate around the high temperature threshold.

We also compare our optimization policy against memory access driven DVFS, in which V-F selections are mainly guided by the memory access rate (e.g., [29]). For implementing memory access driven DVFS, we construct a regression-based model for selecting V-F setting with only

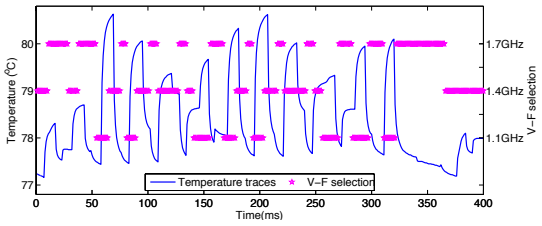

Figure 8: Temperature trace of *ua* on the 3D system running at 1.4GHz/1.02V and the V-F setting selected by our runtime management policy.

Table 3: Comparison of our runtime optimization policy against 3D systems with static settings.

Policy	Static V/F settings (GHz/V)					Runtime optimization		
	0.8/0.98	**1.1/1.0**	**1.4/1.02**	**1.7/1.06**	**2.1/1.1**	**85^oC/155W**	**85^oC/175W**	**85^oC/200W**
Peak P (W)	154.72	161.53	193.37	236.79	279.25	154.85	168.63	189.62
Peak T (^oC)	78.10	79.46	85.85	94.65	103.39	77.97	80.81	83.32
EDP$^{*}\star$ ($J{\cdot}s$)	246.42	167.63	135.18	132.19	119.82	185.67	145.11	130.03
IPnS$^{**}\star$	10.63	12.86	15.73	16.93	18.93	14.47	15.68	16.02

* *EDP per 10billion instructions,* ** *IPnS stands for instructions per nanosecond,* $\star$ *Average across all benchmarks.*

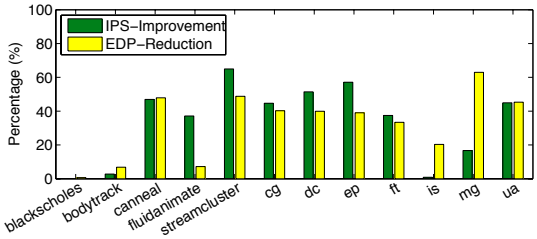

Figure 9: 3D system with runtime management policy in comparison to temperature-triggered DVFS policy.

MA. We show the V-F prediction for 3D system with 85^o/ 175W constraints in Figure 10. By only using MA, three out of twelve benchmarks end up with different V-F settings than the optimal ones; while the predictions are all accurate using both IPC and MA as in our policy. The benchmarks that are predicted incorrectly using only MA are *blackscholes*, *is*, and *mg*. *blackscholes* has low MA but high IPC, *is* has both low MA and low IPC, and *mg* has high MA and relatively higher IPC than the other memory-bound benchmarks. Our policy provides accurate prediction as we take the power and temperature constraints on both logic and DRAM layers into account on 3D systems with stacked DRAM, where both high IPC and memory access rate could result in high chip power and peak temperature.

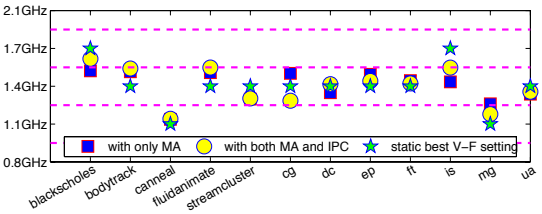

Figure 10: Prediction accuracy of our runtime management policy versus memory access (MA) driven DVFS.

6. CONCLUSION

In this paper, we have provided a methodology to evaluate 3D systems with stacked DRAM and proposed a runtime management policy for dynamically selecting operating points. We have evaluated various access bandwidths to DRAM and demonstrated up to 86.9% performance improvement using parallel access to the DRAM layer compared to regular memory access. Our experiments show that our optimization policy achieves performance improvement of 36.1% for 3D systems with stacked DRAM in comparison to using static V-F settings and EDP reduction of up to 49.4% compared to a temperature-triggered DVFS policy.

REFERENCES

[1] B. Black *et al.*, "Die stacking (3D) microarchitecture," in *MICRO*, pp. 469–479, 2006.

[2] G. H. Loh, "3D-stacked memory architectures for multi-core processors," in *ISCA*, pp. 453–464, 2008.

[3] A. K. Coskun, D. Atienza, T. S. Rosing, T. Brunschwiler, and B. Michel, "Energy-efficient variable-flow liquid cooling in 3D stacked architectures," in *DATE*, pp. 111–116, 2010.

[4] M. Ghosh and H.-H. S. Lee, "Smart refresh: An enhanced memory controller design for reducing energy in conventional and 3D die-stacked DRAMs," in *MICRO*, pp. 134–145, 2007.

[5] S. Liu *et al.*, "Hardware/software techniques for DRAM thermal management," in *HPCA*, pp. 515–525, 2011.

[6] C. Zhu *et al.*, "Three-dimensional chip-multiprocessor run-time thermal management," *IEEE Transactions on CAD (TCAD)*, vol. 27, no. 8, pp. 1479–1492, 2008.

[7] W. Hung *et al.*, "Interconnect and thermal-aware floorplanning for 3D microprocessors," in *ISQED*, pp. 98–104, 2006.

[8] J. Cong *et al.*, "Thermal-aware 3D IC placement via transformation," in *ASP-DAC*, pp. 780–785, 2007.

[9] G. L. Loi *et al.*, "A thermally-aware performance analysis of vertically integrated (3-D) processor-memory hierarchy," in *DAC*, pp. 991–996, 2006.

[10] K. Skadron *et al.*, "Temperature-aware microarchitecture," in *ISCA*, pp. 2–13, 2003.

[11] M. Healy *et al.*, "Multiobjective microarchitectural floor-planning for 2-D and 3-D ICs," *TCAD*, vol. 26, no. 1, pp. 38–52, 2007.

[12] C. Isci, G. Contreras, and M. Martonosi, "Live, runtime phase monitoring and prediction on real systems with application to dynamic power management," in *MICRO*, pp. 359–370, 2006.

[13] R. Cochran, C. Hankendi, A. K. Coskun, , and S. Reda, "Identifying the optimal energy-efïňAcient operating points of parallel workloads," in *ICCAD*, pp. 608–615, 2011.

[14] X. Zhou *et al.*, "Thermal management for 3D processors via task scheduling," in *ICPP*, pp. 115–122, 2008.

[15] M. Awasthi *et al.*, "Handling the problems and opportunities posed by multiple on-chip memory controllers," in *PACT*, pp. 319–330, 2010.

[16] C. C. Liu *et al.*, "Bridging the processor-memory performance gap with 3D IC technology," *IEEE Design Test of Computers*, vol. 22, no. 6, pp. 556–564, 2005.

[17] N. L. Binkert *et al.*, "The M5 simulator: Modeling networked systems," *IEEE Micro*, vol. 26, no. 4, pp. 52–60, 2006.

[18] C. Bienia, *Benchmarking Modern Multiprocessors*. PhD thesis, Princeton University, January 2011.

[19] D. Bailey *et al.*, "The NAS parallel benchmarks," tech. rep., 1994.

[20] S. Li *et al.*, "McPAT: An integrated power, area, and timing modeling framework for multicore and manycore architectures," in *MICRO*, pp. 469–480, 2009.

[21] S. Thoziyoor *et al.*, "CACTI 5.1," tech. rep., April 2008.

[22] R. Kumar *et al.*, "Single-ISA heterogeneous multi-core architectures: the potential for processor power reduction," in *MICRO*, pp. 81–92, 2003.

[23] J. Srinivasan *et al.*, "The case for lifetime reliability-aware microprocessors," in *ISCA*, pp. 276–287, 2004.

[24] H. Su *et al.*, "Full chip leakage estimation considering power supply and temperature variations," in *ISLPED*, pp. 78–83, 2003.

[25] "DRAM power calculations." `http://www.micron.com/`. Micron Technology Inc.

[26] J. Howard *et al.*, "A 48-core IA-32 message-passing processor with DVFS in 45nm CMOS," in *ISSCC*, pp. 108–109, 2010.

[27] J. Meng, C. Chen, A. K. Coskun, and A. Joshi, "Run-time energy management of manycore systems through reconfigurable interconnects," in *GLSVLSI*, pp. 43–48, 2011.

[28] A. K. Coskun *et al.*, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," in *SIGMETRICS*, pp. 169–180, 2009.

[29] C. Isci *et al.*, "An analysis of efficient multi-core global power management policies: Maximizing performance for a given power budget," in *MICRO*, pp. 347 –358, 2006.

SUPPLEMENTAL MATERIAL

S1. Modeling the Thermal Impact of TSVs on 3D Systems with Stacked DRAM

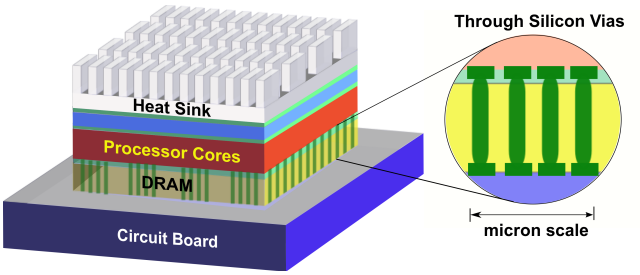

Figure 11: An illustration of the 3D multicore processor with stacked DRAM. TSVs are used to connect the on-chip DRAM layer with the logic layer.

We extend HotSpot to enable the modeling of TSVs in 3D multicore systems with stacked DRAM. The HotSpot extension utilizes the methodology for modeling the interlayer material heterogeneity introduced in prior work [3].

HotSpot 5.0 [10] has the functionality of modeling stacked 3D chips through a layer configuration file which provides the set of layers and the physical properties for each layer. In the default HotSpot tool, the properties across a single layer of the chip are homogenous with same resistivity and capacitance values for all the units. Each layer of the 3D chip is resolved to a grid and the temperature responses are calculated for each grid cell using the parameters from the layer configuration file.

In order to model the thermal effect of the TSVs in 3D stacked systems, our HotSpot extension allows the user to model the heterogeneity in the layer by modifying the resistivity and capacitance for any unit on the chip. For modeling this heterogeneity, we add an additional data structure to each grid to hold grid-specific resistivity and capacitance values. When the temperature responses are being calculated, the tool then uses the grid-specific parameters rather than the layer-specific ones.

Figure 11 provides an illustration of the 3D multicore processor with stacked DRAM. The TSVs connect the on-chip DRAM layer with the logic layer in the 3D multicore systems. Note that TSVs go through both the DRAM and thermal interface layers to connect the active regions of the DRAM and the logic layers. In our target 3D system with parallel access, there are 512 TSVs on each memory controller block. The TSVs have a diameter of $10\mu m$ and a center-to-center pitch of $20\mu m$. To calculate the thermal resistivity of the blocks with TSVs, we assume that the TSVs are evenly spread throughout the memory controller. As we know the dimensions of a single Copper TSV, we can calculate the area the TSVs cover in the memory controller block ($Area_{TSV}$) as well as the area of the memory controller block without TSVs. The joint parallel resistivity (of Copper and thermal interface material, TIM) can be calculated as follows:

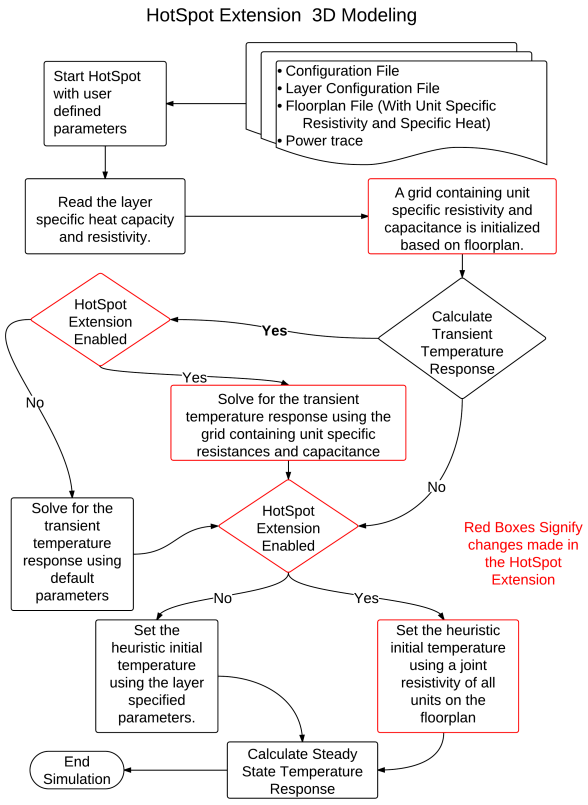

Figure 12: A flow chart for illustrating the HotSpot extension for modeling 3D systems with TSVs by enabling the interlayer material heterogeneity.

$$R_{Joint} = \frac{Area}{\frac{Area - Area_{TSV}}{R_{TIM}} + \frac{Area_{TSV}}{R_{Copper}}} \quad (2)$$

where $Area$ is the area of a memory controller block where TSVs are located at, $Area_{TSV}$ is the area of the memory controller block with TSVs, R_{TIM} is the thermal resistivity of TIM, and R_{Copper} is the thermal resistivity of Copper. Thus, we get the thermal resistivity for the memory controller block with TSVs as $0.156mK/W$, which is lower than the original TIM resistivity of $0.25mK/W$. We also model the TSVs going through the DRAM layer, and compute the joint thermal resistivity of silicon and Copper as $0.0098mK/W$.

We present a flow chart for the implementation of our HotSpot extension in Figure 12. The black boxes are based on the default HotSpot implementation, while the red boxes indicate changes made in the HotSpot tool. In addition to reading the parameters from the layer configuration file, the tool also reads unit-specific parameters from each of the floorplan files. We use the resistivity and capacitance values specified in the floorplan file and assign them to specific grids on each layer. The addition of grid-specific values changes the heuristic initial temperature for steady-state temperature computations. HotSpot will set the initial temperature using only the layer-specific vertical resistance for each layer while ignoring any lateral resistances. Our extension will find the weighted mean of all the vertical resistances

in each grid in the layer. When calculating the initial temperature for each layer, the extension will use this weighted mean instead of using the layer-specific vertical resistance. The Hotspot extension described above has been recently released by our group at: http://lava.cs.virginia.edu/HotSpot-/links.htm.

S2. Target System Modeling Parameters

We model the core architecture of our target system based on the AMD Family 10h microarchitecture used in AMD Magny-Cours processors. Each core has multiple-issue, out-of-order execution, and 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 architectural parameters for cores and caches are listed in Table 4. These parameters are used for the target system configuration in our performance simulations, as described in Section 3.2.

Table 4: Core Architecture Parameters

Architectural Configuration	
CPU Clock	2.1 GHz
Branch Predictor	tournament predictor
Issue	out-of-order
Reorder Buffer	84 entries
Issue Width	3-way
Functional Units	3 IntALU, 1 IntMult, 3 FPALU, 1 FPMultDiv
Physical Regs	128 Int, 128 FP
BTB size	2048 entries
RAS size	24 entries
Load Queue	32 entries
Store Queue	32 entries
L1 I/DCache	64 KB @2 ns, 2-way, 64B block
L2 Cache(s)	16 private L2 Caches, each L2: 16-way set-associative, 64B block 512 KB @6 ns

S3. Additional Details for Parallel Memory Access Modeling

We consider both `regular memory access` and `parallel memory access` to simulate the data transfer between logic layer and DRAM layer in our target 3D system. The `parallel memory access` allows the four on-chip memory controllers accessing the four DRAM ranks at the same time. We deploy 512 TSVs on each memory controller, and these TSVs provide a 64-Byte bus width for each memory controller. In our experiments, we consider TSVs with a diameter of $10\mu m$ and a center-to-center pitch of $20\mu m$. Thus the total TSV area only takes up less than 0.2% of the chip area overhead. From our simulation results for the NAS and PARSEC benchmarks as shown in Figure 13, we observe the main memory accesses are evenly distributed between the four ranks. This provides the justification for assuming the memory access latency with `parallel access` is one fourth of the latency with `regular access`.

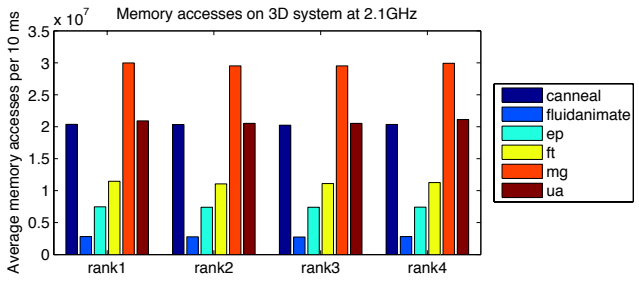

Figure 13: Average memory accesses per 10ms on different DRAM ranks on the 3D system with stacked DRAM.

S4. Power Modeling Parameters

We use five V-F settings in our power model for the target 3D system, matching the five V-F settings in AMD 10h processors. The V-F settings and the corresponding average core power across all the benchmarks for the 3D system with parallel access are shown in Table 5.

Table 5: V-F settings and average per core power values for the 3D system with parallel access.

Frequency(GHz)	2.1	1.7	1.4	1.1	0.8
Voltage(V)	1.10	1.06	1.02	1.0	0.98
Core Power(W)	10.57	8.98	6.98	5.30	4.86

S5. Temperature Modeling Parameters

We provide the thermal parameters used in the HotSpot simulations for 2D and 3D architectures in Table 6. In HotSpot, we set chip thickness at 0.1mm, DRAM thickness at 0.05mm, thermal conductivity of DRAM at 100W/mK (thermal conductivity of silicon), and sampling interval at 1ms. All the other parameters are the same as the default HotSpot configuration to represent efficient packages in high-end systems. The power traces are the inputs for the thermal model. All simulations use the HotSpot grid model for higher accuracy and are initialized with the steady-state temperatures.

Table 6: Thermal simulation configurations in HotSpot.

Parameters	Values
Chip thickness	0.1mm
Silicon thermal conductivity	100 W/mK
Silicon specific heat	1750 kJ/m^{3}K
Sampling interval	0.001s
Spreader thickness	1mm
Spreader thermal conductivity	400 W/mK
DRAM thickness	0.05mm
DRAM thermal conductivity	100 W/mK
Interface material thickness	0.02mm
Interface material conductivity	4 W/mK
Heatsink thickness	6.9mm
Heatsink resistance	0.1K/W