# Dynamic Thermal Management in 3D Multicore Architectures

Ayse K. Coskun[†], José L. Ayala[⋆], David Atienza[‡], Tajana Simunic Rosing[†], Yusuf Leblebici[§]

[†]Computer Science and Engineering Dept. (CSE), University of California, San Diego, USA.
[⋆]Computer Architecture and Automation Dept. (DACYA), Complutense University of Madrid, Spain.
[‡]Embedded Systems Laboratory (ESL), Ecole Polytechnique Fédérale de Lausanne (EPFL), Switzerland.
[§] Microelectronics Systems Laboratory (LSM), EPFL, Switzerland.

***Abstract*—**
**Technology scaling has caused the feature sizes to shrink continuously, whereas interconnects, unlike transistors, have not followed the same trend. Designing 3D stack architectures is a recently proposed approach to overcome the power consumption and delay problems associated with the interconnects by reducing the length of the wires going across the chip. However, 3D integration introduces serious thermal challenges due to the high power density resulting from placing computational units on top of each other. In this work, we first investigate how the existing thermal management, power management and job scheduling policies affect the thermal behavior in 3D chips. We then propose a dynamic thermally-aware job scheduling technique for 3D systems to reduce the thermal problems at very low performance cost. Our approach can also be integrated with power management policies to reduce energy consumption while avoiding the thermal hot spots and large temperature variations.**

## I. Introduction

Technology scaling has caused the feature sizes to shrink continuously, whereas interconnects, unlike transistors, have not followed the same trend. In the nanometer era, a larger portion of the total chip capacitance comes from the interconnects. With the introduction of vias and repeaters to compensate for the performance loss of the long wires, the interconnect power consumption rises dramatically [14].

Designing 3D integrated circuits is one of the recently proposed approaches to overcome the problems associated with interconnects. When components are placed on a 3D architecture, the length of the interconnects and the large power consumption associated with them can be reduced. However, 3D integration introduces challenges due to the high power density resulting from the placement of computational units on top of each other. High power densities are already a major concern in 2D circuits, and in 3D systems the problem is even more severe [4], [20]. In this work, we investigate the thermal behavior of 3D architectures under real-life workloads, and propose dynamic management policies to reduce the adverse affects of temperature on reliability at low performance cost.

Thermal hot spots increase cooling costs, negatively impact reliability and degrade performance. The significant increase in cooling costs requires designing for temperature margins that are lower than the worst-case. Hot spots accelerate failure mechanisms such as electromigration, stress migration, and dielectric breakdown, which cause permanent device failures [13]. Leakage is exponentially related to temperature, and an incremental feedback loop exists between temperature and leakage, which can cause dramatic increases in temperature and damage the circuit if it remains out of control. High temperatures also adversely affect performance, as the effective operating speed of devices decreases as temperature increases.

Addressing thermal hot spots alone is not enough to achieve better reliability, as temperature gradients in time and space determine device reliability at moderate temperatures [17]. The failure rate due to thermal cycling increases with the magnitude and frequency of temperature cycles [13]. Also, large spatial temperature variations across the chip can cause performance and logic failures. Negative bias temperature instability (NBTI) and hot-carrier injection (HCI) cause circuits to fail in meeting timing constraints [15].

To date, temperature related problems in 2D chips have typically been addressed by techniques that lower the average temperature or keep the temperature under a given threshold. Power-aware synthesis, dynamic power management (DPM), dynamic voltage-frequency scaling (DVFS) and dynamic thermal management are examples of such techniques [8]. A significant bottleneck of such methods is the performance impact associated with stalling or slowing down the processor [22]. When the workload that is going to run on the system is known (as in some embedded systems), voltage/frequency levels, architecture configuration or job allocation can be adjusted at the design stage to avoid dynamic thermal management as much as possible [24]. Finally, several temperature-aware job allocation and task migration techniques have been proposed (e.g. [7],[8]) to reduce thermal hot spots and temperature variations dynamically at low cost.

Chip cross-sectional power density increases with the number of vertically stacked circuit layers [28]. This increase exacerbates temperature related reliability, performance and design challenges in 3D systems [4], [20]. 3D integration complicates the implementation of dynamic thermal management techniques because of the heat transfer between vertically adjacent units and the heterogeneous cooling efficiencies of different layers (e.g., the components closer to the heat sink cool down easier than those further away). Therefore, traditional 2D thermal management policies are not sufficient to optimize the temperature profile of multicore 3D systems.

In this work, we first investigate how the existing policies for dynamic thermal management handle the thermal hot spots and temperature gradients in 3D systems. We then propose a low overhead policy for temperature-aware job allocation in 3D architectures. Our contributions can be summarized as follows:

- We investigate both dynamic voltage and frequency scaling (DVFS) and workload migration/scheduling techniques for 3D systems. A thorough comparison demonstrating guidelines and trade-offs for thermal management of 3D circuits is provided.
- We consider two layer and four layer 3D multicore systems in our experiments, designed with various common strategies for the floorplanning of the units, such as placing the cores and memory blocks in separate layers. This way we explore the trade-offs between dynamic policies and design choices.
- We propose a new policy, *Adapt3D*, which takes the thermal history of the processing cores and the 3D system characteristics into account to balance the temperature and reduce the frequency of hot spots. *Adapt3D* has negligible performance overhead and can be combined with DVFS or dynamic power management to reduce energy consumption as well.
- We evaluate the management policies on various 3D systems, whose design is based on the extension of UltraSPARC T1 [18]. Using commercial cores and real-life workload traces collected on them, we are able to provide a realistic experimental infrastructure.

* This work was supported in part by the Swiss NanoTera NTF Project - CMOSAIC, the Spanish Government Research Grants TIN2008-00508 and CSD00C-07-20811, Sun Microsystems, UC MICRO, Center for Networked Systems (CNS) at UCSD, GSRC/DARPA and NSF Greenlight.

The rest of the paper starts with a brief discussion of the related work in temperature management and multicore scheduling in Section II. Section III provides the details of the management techniques investigated in our work, including the new policy we propose. The experimental framework and the results are explained in Sections IV and V, respectively. In Section VI we summarize the main conclusions of this work.

## II. Related Work

In this section, we discuss the prior work in multicore scheduling, and also energy and thermal management in both 2D and 3D domains. Optimizing multicore scheduling with energy and performance (or timing) constraints has been studied quite extensively in the literature [11], [14], [5], [21]. As power-aware policies are not sufficient to prevent temperature-induced problems, thermal modeling and management methods have been proposed. HotSpot [22] is a thermal modeling tool, which calculates transient temperature response given the physical characteristics and power consumption of the units in the die. In [3], a highly-accurate FPGA-based thermal emulation framework is proposed to reduce simulation time for large multicore systems.

Static methods for thermal and reliability management are based on thermal characterization at design time. A task allocation algorithm for platform-based system design, that includes temperature as a constraint in the co-synthesis framework, is introduced in [12]. RAMP [24] provides a reliability model at the architecture level for temperature related failures, and optimizes the architectural configuration and power/thermal management policies for reliable design. In [21], it is shown that aggressive power management can adversely affect reliability due to fast thermal cycles, and the authors propose an optimization method for multicore architectures that saves energy while meeting reliability constraints. A hardware-software emulation framework for reliability analysis is proposed in [2], and a reliability-aware register assignment policy is introduced as a case study.

One of the first works on dynamic thermal management is [6], where the authors explore performance trade-offs between different dynamic management mechanisms to tune the thermal profile at runtime. Computation migration and fetch toggling are examples of dynamic management techniques [22]. Heo et al. reduce peak junction temperature by activity migration between multiple replicated units [11]. Heat-and-Run performs temperature-aware thread assignment and migration for multicore multithreaded systems [10]. Kumar et al. propose a hybrid method that coordinates clock gating and software thermal management techniques such as temperature-aware priority management [16]. The multicore thermal management method introduced in [8] combines distributed DVFS with process migration. For multicore systems, the temperature-aware task scheduling method proposed in [7] achieves more desirable thermal profiles than conventional thermal management techniques without introducing a noticeable impact on performance.

For thermal management of 3D circuits, most of the prior work has addressed design stage optimizations, such as thermally-aware floorplanning (e.g. [9]). For dynamic thermal management in 3D systems, a task assignment algorithm that takes leakage power consumption into account is proposed in [26]. The authors optimize the power profile and chip peak temperature, but their work does not consider runtime management or balancing temperature. The most recent work on thermal management of 3D circuits is presented in [28], where the authors evaluate several policies for task migration and DVFS attending to the feedback information provided by thermal sensors and integrated performance counters. The approach also contains an offline workload profiling phase.

The temperature-aware job scheduling algorithms we propose in this paper optimize the thermal profile in a 3D multiprocessor chip without noticeable impact on the performance of the system. The closest work in literature to our work is [28]. Instead of using offline application profiling for computing the optimal frequency and voltage settings, we propose a fully runtime mechanism. By avoiding the offline phase and the IPC estimation per application, we achieve low and stable temperature profiles at lower design cost. We also evaluate the behavior of the policies for various 3D designs, and show how the thermal behavior changes as the number of layers increases. Finally, we analyze how dynamic thermal management methods affect the temperature variations in addition to hot spots, since thermal variations cause reliability and performance problems.

## III. Thermal Management for 3D Circuits

As we have discussed in the previous section, a number of dynamic thermal strategies have been proposed in the literature for 2D multicore architectures. Our goal is first to analyze how effectively previous methods address the thermal issues in the 3D domain. We then propose a new temperature-aware job allocation technique that handles the 3D-specific thermal issues. Next, we present the details of all the policies we investigate in this work.

### *A. Clock Gating and DVFS Based Techniques*

**Clock Gating (CGate)**, is modeled as proposed in [8], where each core runs at the default (highest) frequency and voltage setting until a core reaches the thermal threshold. At this point, the core that reaches the hot spot is stalled and its clock is gated to reduce power consumption. If the temperature of the core goes below the threshold, execution continues in the next sampling interval.

**DVFS with Temperature Trigger (DVFS_TT)** reduces voltage and frequency (V/f) to the next lower V/f setting when the temperature of a core exceeds the threshold. After that, if the core is still above the threshold, DVFS_TT uses the next lower V/f setting in the next scheduling interval. When the temperature of a core is below the threshold, the V/f setting is increased one step at each scheduling interval. For this policy, it is assumed that we are able to scale down the voltage and frequency of every core independently. Also, in our setup, every core has three 3 V/f levels (i.e., default, 95% of the default and 85% of the default), similar to the assumption in [8].

**Utilization-Based DVFS (DVFS_Util)** observes the core workload in the last interval and, if under-utilized, adjusts the V/f setting to the lowest setting matching the current core workload.

**DVFS with Floorplan Considerations (DVFS_FLP)** assigns a lower V/f setting to cores with higher susceptibility to thermal hot spots. This policy attends to the thermal profile principle in 2D chips, where the cores located closer to the central region of the die get hotter than the cores in the sides and corners. Additionally in 3D systems, while the same principle for 2D applies, the cores on the layers further from the heat sink are more prone to hot spots.

### *B. Job Allocation / Migration Techniques*

**Migration (Migr)** moves the currently running job from a core if the core temperature exceeds the threshold to the coolest core (i.e., the coolest core which did not already receive a migrated job during the current scheduling tick). When the coolest core selected is already running a job, we swap the jobs among the hot and cool cores. This technique can be considered as an extension of core-hopping or activity migration techniques [11], [10].

**Adaptive-Random (AdaptRand)**, which is a policy introduced in [7], updates probabilities of sending workload to cores at each interval based on an analysis of the temperature history on the chip. The use of the thermal history provides the ability to allocate workload on units exposed to lower thermal stress, or located on cooler regions of the multicore architecture.

**Adaptive Policy for 3D (Adapt3D)** is a policy we have designed to specifically address the characteristics of the 3D system. The *Adaptive-Random* policy considers the thermal history of each core; however, it does not differentiate between cores on different layers. As cores on layers closer to the heat sink can be cooled faster in comparison to cores further from the heat sink, *Adapt3D* policy assigns a *thermal index* to each core in order to distinguish the location of the cores. If the thermal index is higher, this shows that the core is more prone to hot spots.

Each core has a probability value assigned at time $t$, $P_t$, which represents the likelihood of the core to receive workload in this interval. When new workload arrives, the allocation is performed based on the probability values of the cores. The $P_t$ values are updated at regular scheduling intervals as follows:

$$P_t = P_{t-1} + W \qquad (1)$$

$$W_{diff} = (T_{pref} - T_{avg}) \qquad (2)$$

$$W = \begin{cases} \beta_{inc} \cdot W_{diff} \cdot \frac{1}{\alpha_i} & : T_{pref} \geq T_{avg} \\ \beta_{dec} \cdot W_{diff} \cdot \alpha_i & : T_{pref} < T_{avg} \end{cases} \qquad (3)$$

where $W$ is the weight factor, $T_{pref}$ is the preferred operating temperature, $T_{avg}$ is the average temperature observed in the history window, $\alpha_i$ ($0 < \alpha_i < 1$) is the thermal index of core $i$ and $\beta$ is an empirically determined constant to decide the rate of change in the probability values. The $\beta$ values for incrementing and decrementing $W$ are different, due to the inclusion of $\alpha$ and $1/\alpha$ in the equations. In our experiments, we set $\beta_{inc} = 0.01$ and $\beta_{dec} = 0.1$, and use a history window length of 10 temperature values (e.g., 1 second interval for a sampling rate of 100ms). At each step, the probability values for all cores are summed up and normalized to 1 to maintain consistency in the computations. Note that other $\beta$ and history window length values can be set, depending on the system and applications.

Using the equation above, the policy updates the probabilities during the execution. The policy favors the cores that are less likely to heat up in the near future by increasing the workload allocation probabilities of cooler cores. In addition, by increasing/decreasing weight factors proportionally with the difference to the preferred operating temperature ($T_{pref}$), *Adapt3D* achieves temperature balancing across the chip. When decreasing the weights, the cores with a higher thermal index ($\alpha$) value have faster decrease in their probability ($P$) values in comparison to cores with lower index (see the weight computation above). This is to ensure that the cores more prone to hot spots due to their location receive fewer jobs than cores with lower temperature. Similarly, when increasing the weights, we increase the weights of the cores with higher indices in a slower fashion.

The thermal index values, $\alpha_i$, can be set offline based on the steady state temperature of cores for typical workloads (and therefore implicitly based on the location of cores), or can be set/updated at runtime by looking at the temperature history. To determine the thermal index values at runtime, a larger history window (e.g. several minutes) needs to be observed, since short time intervals can be misleading in determining the typical thermal characteristics of cores. We experimented with both static and dynamic selection, and set the $\alpha_i$ values offline, as the results were very similar for both options.

$T_{pref}$ is a safe operating temperature, which should be set at a value of several degrees below the critical temperature threshold of the system. In our experiments, $T_{pref}$ has been set to $80^oC$, which is a safe temperature for the type of commercial multicore architectures studied in this work [18].

If the temperature of a core exceeds the pre-set threshold value ($85^oC$ in our experiments) in the last observed interval, the core's probability value is set to zero to avoid heating up the core and to prevent reliability failures. Note that as the policy effectively manages the temperature, this case is a rare occurrence. It should also be noticed that, as the probability values are adjusted in proportion with the temperature difference from the target temperature value, we do not overload cores that are already highly utilized and getting warm.

### C. Hybrid Techniques

We also design hybrid policies which integrate both DVFS and job allocation/migration. In particular, we combine the best-performing job allocation policy (*Adapt3D*) with each of the DVFS policies. A thorough evaluation is available in Section V.

## IV. Experimental Framework

The 3D multicore systems studied in our experiments are based on the architecture of the UltraSPARC T1 (i.e., Niagara-1) [18], which is manufactured at 90nm technology. The average power consumption, including leakage, area distribution of the units on the chip and the floorplan of UltraSPARC T1 are available in [18] and in [7]. This architecture is composed of 8 cores with multithreading capability, and a shared L2-cache for every two cores.

### A. 3D Floorplan of the Units

In this work, we have considered several design possibilities for the layout of the 3D system. The proposed floorplan diagrams are provided in Figure 1.

***EXP-1:*** One approach to design the 3D system is to place the logic units (i.e., the processing cores) and the memory blocks (i.e., caches, etc.) on separate layers. Placing cores and their associated memories on separate layers is a preferred scenario for systems with a large number of memory accesses, such as systems targeting multimedia applications. In this way, the length of interconnections between the cores and their caches can be reduced, achieving higher performance. Such an architecture also allows the use of different process technologies for manufacturing the cores and memories, which can result in better optimized designs. Thus, in our first set of experiments, we place cores and L2 caches (i.e., scdata) of the UltraSPARC T1 on separate layers (see Figure 1).

***EXP-2:*** As stated above, separating the core and memory layers is an attractive approach for some systems, but it brings other challenges. For example, testing the layer that contains only memory blocks independently requires the development of special test structures, because the layer does not contain logic units. Therefore, our second configuration is a 2-layered system, where each layer has four cores and their L2 caches (see Figure 1).

***EXP-3, EXP-4:*** Finally, we have developed 4-tier systems to provide a thorough investigation of the effects of thermal management policies in 3D systems. EXP-3 and EXP-4 use the same layer structure in EXP-1 and EXP-2 respectively, but duplicates the layers to build a system with 16 cores (as shown in Figure 1).

### B. Workload and Power Model

The first step to construct our experimental framework is gathering detailed workload characteristics of real applications on an UltraSPARC T1. We sampled the utilization percentage for each hardware thread at every second using `mpstat`. During this profiling, we ran half an hour long traces for each benchmark. Also, the length of user and kernel threads were recorded using `DTrace` [19] to determine the active/idle time slots of cores more accurately.

We have used various real-life benchmarks including web servers, database management, and multimedia processing. A typical server

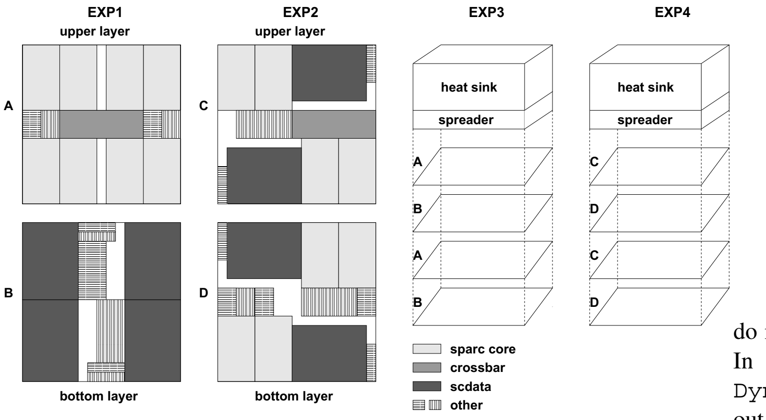

Fig. 1. Floorplans

workload was generated by running SLAMD [23] with 20 and 40 threads per client to achieve medium and high utilization, respectively. For generating representative database applications, we experimented with MySQL using `sysbench` for a table with 1 million rows and 100 threads. We also ran the `gcc` compiler and the `gzip` compression/decompression benchmarks as samples of SPEC-like benchmarks. Finally, we ran several instances of the `mplayer` (integer) benchmark with 640x272 video files as typical examples of multimedia processing. A detailed summary of the benchmarks workloads is shown in Table I. The utilization ratios are averaged over all cores throughout the execution. We also recorded the cache misses and floating point (FP) instructions per 100K instructions using `cpustat`. For EXP-3 and EXP-4, which have 16 cores on the architecture, we duplicated the workload collected on the 8-core UltraSPARC T1.

TABLE I. WORKLOAD CHARACTERISTICS

| | **Benchmark** | Avg<br>Util (%) | L2<br>I-Miss | L2<br>D-Miss | FP<br>instr |
|---|---|---|---|---|---|
| 1 | Web-med | 53.12 | 12.9 | 167.7 | 31.2 |
| 2 | Web-high | 92.87 | 67.6 | 288.7 | 31.2 |
| 3 | Database | 17.75 | 6.5 | 102.3 | 5.9 |
| 4 | Web & DB | 75.12 | 21.5 | 115.3 | 24.1 |
| 5 | gcc | 15.25 | 31.7 | 96.2 | 18.1 |
| 6 | gzip | 9 | 2 | 57 | 0.2 |
| 7 | MPlayer | 6.5 | 9.6 | 136 | 1 |
| 8 | MPlayer&Web | 26.62 | 9.1 | 66.8 | 29.9 |

The peak power consumption of SPARC is close to its average power [18]. Thus, we assumed that the instantaneous power consumption is equal to the average power at each state (active, idle, sleep). The `active` state power is taken as 3 Watts, based on [18]. The cache power consumption is 1.28W per each L2, which is computed with CACTI [27], and verified by the percentage values in [18]. In order to simulate `dynamic voltage/frequency scaling (DVFS)`, we estimated the power at lower voltage levels based on the equation $P \propto f \cdot V^2$. Three built-in voltage/frequency settings are assumed in our simulations. The crossbar power consumption was modeled by scaling the average power value according to the number of active cores and the memory access statistics.

The leakage power of the processing cores is calculated according to different structural areas of the system and their temperature. We assume a base leakage power density of $0.5W/mm^2$ at 383K as in [5]. To account for the temperature and voltage effects on leakage power, we used the second-order polynomial model proposed in [25]. We determined the coefficients in the model empirically to match the normalized leakage values in [25].

Many current systems have power management capabilities to reduce energy consumption. Although power management techniques do not directly address temperature, they affect the thermal behavior. In addition to the investigated DVFS techniques, we implement `Dynamic Power Management (DPM)`. We utilize a fixed time-out policy, which puts a core to sleep state if it has been idle longer than the timeout period. We set a `sleep` state power of 0.02 Watts, based on the `sleep` power of similar cores.

TABLE II. THERMAL MODEL AND FLOORPLAN PARAMETERS

| **Parameter** | **Value** |
|---|---|
| Die Thickness (one stack) | $0.15mm$ |
| Area per Core | $10mm^2$ |
| Area per L2 Cache | $19mm^2$ |
| Total Area of Each Layer | $115mm^2$ |
| Convection Capacitance | 140 J/K |
| Convection Resistance | 0.1 K/W |
| Interlayer Material Thickness (3D) | 0.02 mm |
| Interlayer Material Resistivity | 0.25 mK/W |

### C. Thermal Model

HotSpot Version 4.2 [22] was employed as thermal modeling tool. We used the 3D capability available in the grid model of the tool, and the proposed layouts of the system were incorporated for the analysis. For the package, the default characteristics in HotSpot V.4.2. were used, as these represent a modern CPU package. The thermal sampling interval was 100 ms, which provided sufficient precision. HotSpot was initialized with steady state temperature values. The model parameters are shown in Table II.

The interface material in between the silicon layers is modeled as a homogeneous layer (identified by thermal resistivity and specific heat capacity values) in the default thermal model. We computed the thermal impact of the through-silicon-vias (TSV) connecting the layers by assuming a homogeneous via distribution on the die, and calculated the "combined" resistivity of the interface material based on the TSV density. A similar model has also been utilized in [28]. We examined the joint resistivity for various TSV density values ($d_{TSV}$), where $d_{TSV}$ is the ratio of the total area overhead introduced by the TSVs to the total layer area. We have observed that even when the TSV density reaches 1-2%, the effect on the temperature profile is limited to only a few degrees, which justifies using a homogeneous TSV density in the model. We assumed that the effect of the TSV insertion to the heat capacity of the interface material is negligible, sine the total area of TSVs constitutes a very small percentage of the interface material area.

The resistivity as a function of the via density is shown in Figure 2. Each via has a diameter of $10\mu m$, according to the current TSV technology [28], and the spacing required around the TSVs is assumed as $10\mu m$. For our experiments, we used a joint interlayer resistivity value of $0.23mK/W$, assuming an abundant number of vias (i.e., total number of vias is 1024) while keeping the area overhead below 1%. Note that, while the exact location of TSVs might demonstrate a further reduction in temperature in comparison to the homogeneous TSV distribution model, our assumption places over 8 TSVs per $mm^2$. Assuming a relatively high TSV density in our model reduces the temperature difference in comparison to modeling the exact location of TSVs.

### D. Dynamic Management Infrastructure

We have integrated the job scheduler and power manager with the thermal simulator to be able to control the system at runtime and measure the thermal behavior. We assume that each core has a temperature sensor, which provides temperature readings at regular intervals (e.g., 100ms). Modern OSes have a multi-queue structure, where each CPU core is associated with a dispatching queue, and the job scheduler allocates the jobs to the cores according to the current

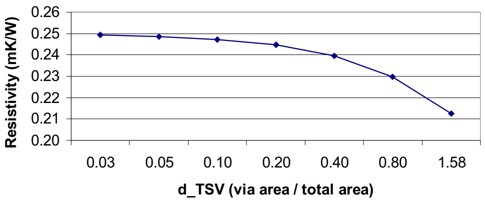

Fig. 2. Effect of Vias on the Resistivity of the Interface Material

policy. In our simulator, we implemented a similar infrastructure, where the queues maintain the threads allocated to cores, and execute them in order.

## V. Results

In this section, we compare the thermal behavior and performance of all the techniques discussed in Section III. As baseline policy to compare in our analysis, we employ the **Dynamic Load Balancing (Default)**, since it is the default policy in most modern OSes (such as the Solaris SUN-OS operating system running on Niagara-1). This policy assigns an incoming thread to the core where it ran previously. If the thread has not run recently, then the dispatcher assigns it to the core that has the lowest priority thread in the queue. In this context, the dispatcher tries to assign the thread based on locality (e.g., if several cores are sharing a cache or on the same chip, etc.). Then, at runtime, if there is a significant imbalance among the queues, the threads are migrated to achieve a more balanced utilization.

### *A. Performance*

We have evaluated the impact on performance of the different policies by computing the average delay in the completion time of jobs with respect to the default policy. Based on our measurements of thread migration in Solaris-OS running on the actual UltraSPARC T1 architecture, we assumed for these experiments that each thread migration takes $1ms$. Regarding DVFS, we assumed that the performance of an application scales linearly with the frequency of the core where the application is running.

Adapt3D and Adaptive-Random only update the probability values at workload arrivals, so both methods are extremely light-weight, and have negligible performance overhead in comparison to the default policy. The random number generator needed for the policies can be implemented through a linear-feedback shift register (LFSR), which often exists on the chip for test purposes.

Figure 3 compares the performance of all the policies, normalized with respect to their default performance. Performance is shown as a line graph, quantified on the secondary (right) y-axis. We observe that, a job scheduling policy, which makes decisions based on floorplan and runtime characteristics, can be a significant aid to a good DVFS policy with its ability to find a beneficial thread-to-core assignment. When Adapt3D is combined with the DVFS policies, we can achieve much better thermal results while reducing the performance overhead of DVFS considerably.

### *B. Thermal Hot Spots*

In our first set of experimental results, we evaluate the effect of the policies on the occurrence of thermal hot spots. Our results demonstrate the percentage of time spent above $85^oC$, which is considered a high temperature for our systems.

Figure 3 shows the thermal hot spots for all the experiments and all the policies for the setup without dynamic power management (DPM). The most successful policies are the hybrid policies. For EXP-1, Adapt3D performs very similarly to Adaptive-Random, and the hybrid policies that utilize Adapt3D and DVFS provide limited benefit in comparison to the policies with only DVFS. However, for EXP-3 and EXP-4, combining Adapt3D with the DVFS policies achieve between 20-40% reduction in hot spots in comparison to DVFS. Therefore, more complex 3D architectures with multiple active layers clearly benefit from the proposed 3D-specific policy. We do not report the hybrid policy of Adaptive-Random and DVFS, as Adapt3D already achieves up to 32% reduction in hot spots in comparison to Adaptive-Random, and the hybrid policy utilizing Adapt3D outperforms the one utilizing Adaptive-Random.

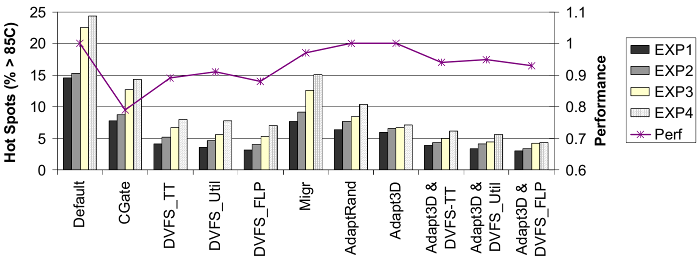

Fig. 3. Thermal Hot Spots (Without DPM) and Performance

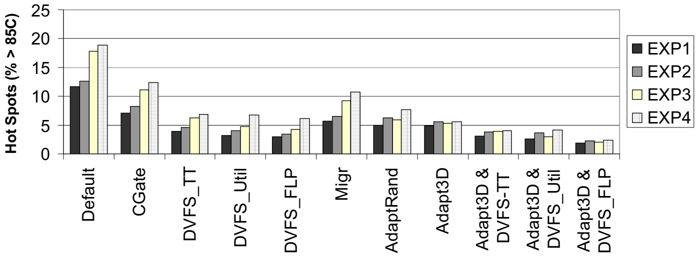

Fig. 4. Thermal Hot Spots - With DPM

In Figure 4, the hot spot frequencies are shown for all the policies integrated with DPM. In comparison to the previous results in Figure 3, we see that a significant reduction in the occurrence of thermal hot spots is achieved. This is due to the fact that when the cores are in `sleep` state, the temperature reduces considerably, hence reducing the amount of thermal emergencies. The policies without DVFS benefit more from DPM, as DVFS fills in part of the idle time slots by reducing frequency and extending execution time.

Some of the policies presented in Figure 4 are similar to the approaches proposed in [28]. In [28] the authors present a similar technique to `DVFS_TT` called *distributed DVFS with clock throttling*. Also, `DVFS_Util` is similar with *global power-thermal budgeting*, considering `DVFS_Util` is also a performance oriented policy. In global power-thermal budgeting, the authors measure the IPC instead of the utilization. The combination of our `DVFS_FLP` policy with `Migration` has the same principle as the *distributed thermal-aware workload migration*; however, this technique requires IPC measurements in addition to temperature, and utilizes offline profiling.

### *C. Spatial Gradients*

To evaluate the spatial gradients on the 3D systems, our results show the percentage of time that gradients above $15^oC$ occur, as gradients between $15-20^oC$ start causing clock skew and impact on circuit delay [1]. The spatial distribution is calculated by evaluating the temperature difference between hottest and coolest units on each layer, and getting the maximum of the per-layer gradients at each sampling interval.

In our experiments, we investigated vertical gradients as well, considering that the temperature difference of blocks on top of each other (i.e., on adjacent layers) may affect the performance and reliability of the TSVs. However, we observed that the vertical gradients between adjacent layers are limited to a few degrees only,

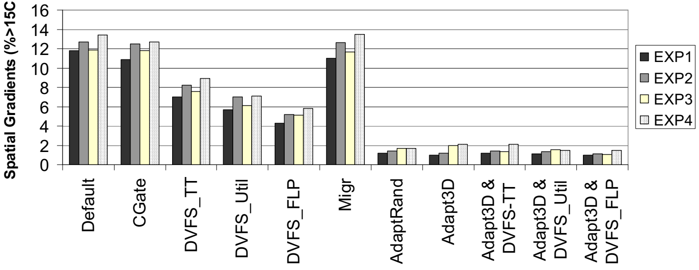

Fig. 5. Spatial Gradients - With DPM

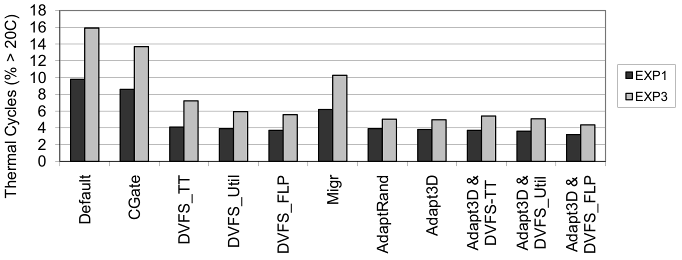

Fig. 6. Thermal Cycles - With DPM

due to the fact that the interlayer material is thin and has sufficient conductivity.

Figure 5 shows the frequency of large spatial gradients for all the experiments. Adaptive scheduling policies, which balance out the temperature on the chip, outperform the other techniques by large in reducing the gradients. Also, note that there is only a slight increase in the percentage of gradients for EXP-3 and EXP-4, which is due to the higher frequency of hot spots in the 4-tiered systems. The reason for having a slight increase is that we only consider per-layer gradients, and do not look at the interlayer variations.

### *D. Temperature Cycles*

Next, we study the temporal cycles of the 3D systems; namely, we analyze the frequency of thermal fluctuations above $20^oC$. For metallic structures, assuming the same frequency of thermal cycles, failures happen 16× more frequently when $\Delta T$ increases from 10 to $20^oC$ [13]. In our experiments, the $\Delta T$ values are computed over a sliding window and averaged over all cores. We only report the cycles for the case with DPM, as switching to `sleep` state causes cycles large enough to degrade reliability.

Figure 6 shows the thermal cycling results for the various policies discussed. In complex 3D architectures with four layers, such as EXP3, large thermal cycles occur more often. The reason for this is that, as the average temperature on chip is higher in comparison to the 2-layer system, the magnitude of the cycles is also typically higher. We see that Adapt3D reduces the frequency of large cycles by over 60%. Note that Adapt3D achieves similar thermal profiles as DVFS at a much lower performance cost.

## VI. Conclusion

The design of 3D stack architectures is a promising approach for improving the performance in multicore systems. However, 3D integration increases power density and accelerates the temperature related problems. In this work, we have presented a thorough analysis of the behavior of well known 2D thermal management techniques in 3D multicore architectures. We have considered two- and four-layered systems designed based on the multicore UltraSPARC T1 system, and our experimental work has shown the trade-offs between achieving more reliable thermal profiles and performance.

We have proposed a novel low-cost technique, *Adapt3D*, for dynamic thermally-aware job scheduling in 3D systems. We have shown that our technique provides a significant reduction on the frequency of hot spots, spatial gradients and thermal cycles. *Adapt3D* achieves similar results to DVFS in the optimization of thermal profiles, while the performance cost is kept to a minimum. The impact of the location of cores has been considered in *Adapt3D* to balance the temperature across the chip more effectively. We have demonstrated that such location impact is significant especially for 3D systems with more than two layers. When combined with DVFS, our approach improves the reduction of hot spots by an additional 20%-40% in comparison to performing only DVFS, and reduces the performance cost.

## References

[1] A. H. Ajami, et al. Modeling and analysis of nonuniform substrate temperature effects on global ULSI interconnects. *IEEE Transactions on CAD*, 24(6):849–861, June 2005.

[2] D. Atienza, et al. Reliability-aware design for nanometer-scale devices. In *ASPDAC*, 2008.

[3] D. Atienza, et al. A fast HW/SW FPGA-based thermal emulation framework for multi-processor system-on-chip. In *DAC*, 2006.

[4] B. Black, et al. Die stacking (3d) microarchitecture. In *MICRO*, 2006.

[5] P. Bose. Power-efficient microarchitectural choices at the early design stage. In *Keynote Address on PACS*, 2003.

[6] D. Brooks, et al. Dynamic thermal management for high-performance microprocessors. In *HPCA*, 2001.

[7] A. K. Coskun, et al. Temperature aware task scheduling in MPSoCs. In *DATE*, 2007.

[8] J. Donald, et al. Techniques for multicore thermal management: Classification and new exploration. In *ISCA*, 2006.

[9] M. Healy, et al. Multiobjective microarchitectural floorplanning for 2-d and 3-d ICs. *IEEE Transactions on CAD*, 26(1), Jan 2007.

[10] M. Gomaa, et al. Heat-and-Run: leveraging SMT and CMP to manage power density through the operating system. In *ASPLOS*, 2004.

[11] S. Heo, et al. Reducing power density through activity migration. In *ISLPED*, 2003.

[12] W.-L. Hung, et al. Thermal-aware task allocation and scheduling for embedded systems. In *DATE*, 2005.

[13] Failure mechanisms and models for semiconductor devices, JEDEC publication JEP122C. http://www.jedec.org.

[14] P. Kapur, et al. Power estimation in global interconnects and its reduction using a novel repeater optimization methodology. In *DAC*, pages 461–466, 2002.

[15] H. Kufluoglu, et al. A computational model of NBTI and hot carrier injection time-exponents for MOSFET reliability. *Journal of Computational Electronics*, 3 (3):165–169, Oct. 2004.

[16] A. Kumar, et al. HybDTM: a coordinated hardware-software approach for dynamic thermal management. In *DAC*, 2006.

[17] C. J. Lasance. Thermally driven reliability issues in microelectronic systems: status-quo and challenges. *Microelectronics Reliability*, 43(12):1969–1974, Dec. 2003.

[18] A. Leon, et al. A power-efficient high-throughput 32-thread SPARC processor. *ISSCC*, 2006.

[19] R. McDougall, et al. Solaris Performance and Tools. *Sun Microsystems Press*, 2006.

[20] K. Puttaswamy, et al. Thermal herding: Microarchitecture techniques for controlling hotspots in high-performance 3d-integrated processors. In *HPCA*, 2007.

[21] T. S. Rosing, et al. Power and reliability management of SoCs. *IEEE Transactions on VLSI*, 15(4), April 2007.

[22] K. Skadron, et al. Temperature-aware microarchitecture. In *ISCA*, 2003.

[23] SLAMD Distributed Load Engine. www.slamd.com.

[24] J. Srinivasan, et al. The case for lifetime reliability-aware microprocessors. In *ISCA*, 2004.

[25] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif. Full-chip leakage estimation considering power supply and temperature variations. In *ISLPED*, 2003.

[26] C. Sun, L. Shang, and R. P. Dick. 3d multiprocessor system-on-chip thermal optimization. In *CODES+ISSS*, 2007.

[27] D. Tarjan, S. Thoziyoor, and N. P. Jouppi. CACTI 4.0. Technical Report HPL-2006-86, HP Labs, Palo Alto, 2006.

[28] C. Zhu, Z. Gu, L. Shang, R. P. Dick, and R. Joseph. Three-dimensional chip-multiprocessor run-time thermal management. *IEEE Transactions on CAD*, 27(8):1479–1492, August 2008.