Amdahl's Law for Lifetime Reliability Scaling in Heterogeneous Multicore Processors

William J. Song, Saibal Mukhopadhyay, and Sudhakar Yalamanchili
School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332
wjhsong@gatech.edu, saibal@ece.gatech.edu, sudha@gatech.edu

ABSTRACT

Heterogeneous multicore processors have been suggested as alternative microarchitectural designs to enhance performance and energy efficiency. Using Amdahl's Law, heterogeneous models were primarily analyzed in performance and energy efficiency aspects to demonstrate its advantage over conventional homogeneous systems. In this paper, we further extend the study to understand the lifetime reliability consequences of heterogeneous multicore processors, as reliability becomes an increasingly important constraint. We present the lifetime reliability models of multicore processors based on Amdahl's Law, including compact thermal estimation that has strong correlation with device aging. Lifetime reliability is analyzed by varying i) core utilization (Amdahl's scaling factor), ii) processor composition (number of big and small cores), and iii) thread scheduling method. The study shows that the heterogeneous processor may have a serious reliability challenge. If the processor is comprised of only one big core and many small cores, stresses can be biased to the big core especially when workloads spend more time on sequential operations. Our study reveals that incorporating multiple big cores can mitigate reliability bottleneck in big cores and enhance processor lifetime, but adding too many big cores will have an adverse impact on lifetime reliability as well as performance.

1. INTRODUCTION

The paradigm of designing processors is shifting from simply improving performance to enhancing energy (or power) efficiency, as it has become a critical barrier to microarchitectural operations. Heterogeneous multicore processors have been studied as alternative implementations to improve energy efficiency and performance. For instance, a processor comprised of a complex core (i.e., out-of-order execution) and many small cores (i.e., in-order execution) can enhance these metrics by using the big core for faster sequential executions and many simple cores for energy-efficient parallel operations. We refer to such microarchitectural asymmetry as *heterogeneous* in this paper. Energy efficiency and performance improvements of the heterogeneous processor over a conventional homogeneous processor are governed by Amdahl's Law [1], as widely studied in prior work [3, 4, 6, 11, 15, 18, 19, 23, 30, 34].

Extending prior work for the performance and energy modeling of heterogeneous processors, this paper presents the lifetime reliability models of heterogeneous multicores and discusses the reliability implication of such heterogeneous designs. Lifetime reliability behaviors of a heterogeneous processor can also be characterized by using Amdahl's Law. Depending on Amdahl's scaling (or parallelization) factor f and processor composition (e.g., number of big and small cores), stresses can be biased to a particular type of core. For instance, assume that the heterogeneous processor includes only one complex core such as Figure 1.(a) and also utilizes the complex core during parallel executions to maximize performance increase via techniques such as bias scheduling [18] or accelerating critical threads [15, 30]. The big core is the busiest computing unit in that it is always turned on and has to execute both serial and a part of parallel phases of a workload. It is subject to extended stresses compared to other computing units, and such *biased stresses* become worse when the application spends more time on sequential operations. This is generally not a critical issue in homogeneous multicore processors, since any one of the cores can be selected to execute sequential operations. Load-balancing or wear-leveling methods can be applied to the homogeneous cores to even out degradation [7, 14, 26]. On the other hand, if sequential executions are substantially short, the chance of failure is greater among many simple cores. When a small core fails, the heterogeneous processor may tolerate the failure if graceful degradation is allowed for a number of duplicated small cores on the die [7, 14, 29]. However, the failure of the only big core immediately leads to the failure of the entire processor since no other cores can replace the role of the failed big core without serious performance degradation.

If the heterogeneous processor accommodates a few number of big cores such as Figure 1.(b), any one of them can be selected to perform serial executions. Unused cores can be power-gated to minimize degradation and save power. Consequently, the failure rate of complex cores can be greatly reduced by sharing loads. However, increasing the number of big cores on the die reduces the small core count under the same area constraint. It may decrease the peak performance of parallel executions, especially for small-size processors where relatively large portion of the area would be taken by big cores. Alternatively, different scheduling methods can be considered to mitigate the reliability problem of the heterogeneous processor. For instance, a complex core can be used only to execute sequential operations and turned off during parallel phases. This type of scheduling policy is particularly advocated in power-constrained processors where not all cores might be able to turn on because of power limitation [6, 34]. Such a scheduling method also limits the peak throughput of parallel executions, but the power-gated big core ben-

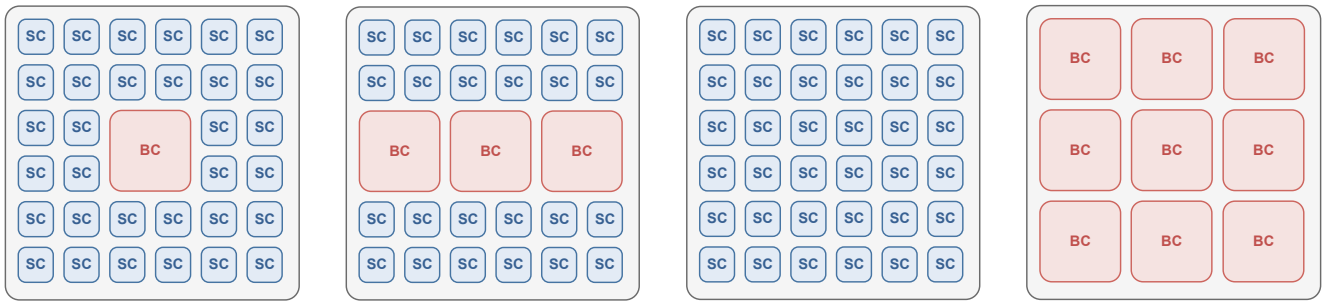


Figure 1: Multicore configurations: (a) heterogeneous processor with one big core (BC) and many small cores (SC), (b) heterogeneous processor with multiple big cores and fewer small cores, (c) homogeneous processor of small cores, and (d) homogeneous processor of big cores.

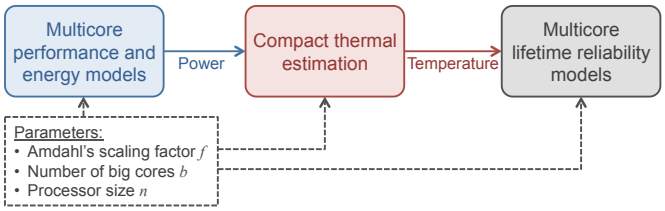


Figure 2: Modeling flow of performance, energy efficiency, thermal, and lifetime reliability characterization of heterogeneous multicore processors.

efits from alleviated stresses and improves overall reliability. Therefore, the lifetime reliability of heterogeneous multicore processors strongly depends on processor configuration (e.g., number of big and small cores) and utilization (e.g., scheduling method), characterized by Amdahl's Law. This paper proposes and uses the approach shown in Figure 2 to evaluate the lifetime reliability of multicore processors. As shown, this paper makes the following contributions.

- *Performance and energy efficiency models*:
  We extend heterogeneous multicore models in the work of Hill and Marty [11] and Woo and Lee [34] to include multiple complex cores and utilize power-gating (of unused cores) in performance and energy calculations.

- *Thermal estimation for reliability modeling*:
  We present a compact thermal model to estimate the temperature of hypothetical heterogeneous processors with different processor compositions (i.e., number of big and small cores) and execution phases (i.e., serial or parallel) for accurate reliability modeling.

- *Lifetime reliability models of heterogeneous processors*:
  Using the preceding models, this paper presents the lifetime reliability models of heterogeneous processors and shows that multicore reliability is subject to Amdahl's Law.

- *Assessing the reliability of heterogeneous processors*:
  We show that the performance, energy efficiency, and lifetime reliability of heterogeneous processors are traded as a function of processor size n, big core count b, and Amdahl's scaling factor f.

This paper is organized as follows. First, we summarize prior work regarding the performance and energy impacts of heterogeneous processors. We revisit the heterogeneous multicore models from prior work and extend them to have various numbers of big and small cores on the die. The performance and energy models are evaluated by varying Amdahl's scaling factor and total core count. We present our methodology to model the lifetime reliability of multicore processors. Finally, we evaluate the performance, energy efficiency, and lifetime reliability of heterogeneous processors with more realistic performance and power models extrapolated from detailed microarchitectural simulations.

2. PRIOR WORK FOR HETEROGENEOUS MULTICORE PROCESSOR ANALYSIS

Applications exhibit different performance and energy (and power) behaviors depending on executing core types. Kumar et al. [19] showed that matching application characteristics to computing resources could enhance energy efficiency. Hill and Marty [11] extended Amdahl's Law to study the performance impact of heterogeneous multicore processors compared to conventional homogeneous processors. Following the work of Hill and Marty, Woo and Lee [34] presented the energy scaling models of multicore processors including a heterogeneous design. Their models enabled the evaluation of energy and power-related metrics such as performance per Joule or Watt for various multicore compositions. Chung et al. [4] explored hypothetical heterogeneous computing models comprised of conventional high-performance cores, minimally-sized baseline cores, and diverse unconventional computing units such as FPGAs and GPGPUs. Their analytical models showed that these unconventional cores could provide better energy efficiency than conventional CPUs for highly parallel applications. As total power becomes a critical limitation, Morad et al. [23] presented theoretical performance models of heterogeneous processors under a power budget constraint. Esmaeilzadeh et al. [6] presented the projection of power-limited multicore systems with the view of emerging dark silicon era.

Using heterogeneous computing units requires sophisticated scheduling methods to maximize utilization and performance. Since different types of computing units have distinct computing capabilities, Koufaty et al. [18] suggested a method

for bias scheduling to handle performance imbalance between heterogeneous cores. Suleman and Joao et al. [15, 30] studied identifying critical threads in parallel executions and executing them on high-performance cores to speed up overall execution. Cao et al. [3] presented measurement-based analysis to support virtual machine services and improve their performance and energy efficiency in heterogeneous processors. In our paper, we do not dive into the details of these scheduling problems. Instead, we assume that these efforts would potentially enable us to maximally utilize heterogeneous multicore microarchitectures.

Heterogeneous processors have been largely studied in performance and energy (or power) aspects, but their reliability implications have been overlooked. Prior work focused on characterizing and improving the lifetime reliability of homogeneous multicore processors. Proposed techniques encompass wear-leveling [7, 14, 26], component redundancy [14, 27], and thermal and power management methods [5, 21, 22]. A greatly simplified theoretical model is found in the work of Huang et al. [13]. The authors studied the lifetime reliability of a heterogeneous processor comprised of a few cores, but they did not include enough details of heterogeneous multicore designs and operations. Yu et al. [35] presented multifaceted analysis of homogeneous processors. Their study explored various multicore compositions in terms of core size and count under an area constraint and evaluated their performance, yield, and reliability. In this paper, we characterize and study the lifetime reliability consequences of heterogeneous multicore processors based on Amdahl's Law, extending prior work for performance and energy analysis.

3. REVISITING AMDAHL'S LAW FOR PERFORMANCE AND ENERGY SCALING OF MULTICORE PROCESSORS

We adopt the performance and energy models from previous work [11, 34] and extend them to analyze the lifetime reliability of heterogeneous processors. In this section, we first review the performance and energy models of various multicore configurations with updated assumptions.

3.1 Homogeneous Processor of Simple Cores

We assume that a baseline homogeneous processor is comprised of n number of simple (small) cores, following the modeling methodology presented in the work of Hill and Marty [11]. Figure 1.(c) illustrates the homogeneous multicore processor composed of small cores. According to Amdahl's Law, maximum performance speed-up is given as Eq. (1). Performance is improved by parallelizing the f fraction of computations with n cores [11]. This is optimistic performance estimation without thread parallelization or migration overhead. We refer to the f fraction as *Amdahl's scaling factor* or *parallelization fraction* in this paper.

$$Perf_{hom:s} = \frac{1}{(1-f)+\frac{f}{n}} \tag{1}$$

We bring energy models from the work of Woo and Lee [34] but modify the assumption such that idle cores (e.g., unused $n-1$ number of cores during serial executions) are ideally turned off and do not contribute to processor power dissipation. We assume that the power consumption of a simple core is normalized to 1. Processor power dissipation during serial executions is equivalent to single-core power, and the processor consumes $n \times 1$ amount of power when all cores are active to execute parallel threads. With the normalization, energy scaling based on Amdahl's Law becomes $E_{hom:s} = 1$, and power scales the same as the performance model in Eq. (1), $W_{hom:s} = Perf_{hom:s}$.

3.2 Homogeneous Processor of Complex Cores

In a homogeneous processor composed of complex (big) cores such as Figure 1.(d), it is assumed that each big core has s times better performance and r times larger area than those of a small core [4, 11, 34]. Pollack's Rule [25] states that performance and area are correlated as $s \propto \sqrt{r}$. Within the same total area as the homogeneous processor of small cores, there can be up to n/r number of big cores. The performance speed-up of the homogeneous processor of complex cores is calculated as Eq. (2). The improvement is achieved by accelerating serial executions (i.e., $1-f$ fragment of a workload) by an s times faster big core and parallelizing the f fraction of the load by n/r number of big cores.

$$Perf_{hom:b} = \frac{1}{\frac{1-f}{s}+\frac{f}{s}\times\frac{r}{n}} \tag{2}$$

Another parameter p is considered to represent the relative power of a big core, which means that the big core consumes p times more power than a small core. We adopt this power expression from the work of Chung et al. [4], where power and area (or performance) are correlated as $p \propto (\sqrt{r})^{\alpha}$ and α is set to 1.75. The energy dissipation of the processor is expressed as Eq. (3). In this equation, it is also assumed that unused cores are power-gated. During serial executions, the processor consumes p amount of power that is equivalent to single big-core power. It dissipates $p \times (n/r)$ amount of power when all big cores are active for parallel executions.

$$E_{hom:b} = \frac{1-f}{s}p + \frac{f}{s}\times\frac{r}{n}\times\frac{n}{r}p \tag{3}$$

3.3 Heterogeneous Processor with Maximum Scheduling

Departing from a simple heterogeneous model that has only one big core and many small cores as studied in prior work [4, 6, 11, 34], we generalize the heterogeneous configuration to incorporate multiple big cores. When executing only one application at a time, one complex core is sufficient to handle the serial part of the application. However, in a general situation such as multiplexed applications and system operations (e.g., virtual environment), there can be a need for including multiple complex cores to handle concurrent serial executions of multiple workloads. Therefore, it is a valid design for the heterogeneous processor to include multiple big cores, and we consider such a design in our analysis. We assume *maximum scheduling* for the heterogeneous processor such that it can fully utilize computing units to maximize performance.

$$Perf_{het:ms} = \frac{1}{\frac{1-f}{s}+\frac{f}{b\times s+(n-b\times r)}} \tag{4}$$

Table 1: Comparison of Simple/Complex-Core Homogeneous Processor Pairs

	IBM Blue Gene/Q	IBM POWER7	Intel Atom Z520	Intel i7 960
Core execution type	In-order	Out-of-order	In-order	Out-of-order
Technology node	45nm	45nm	45nm	45nm
Estimated die area	360mm^2	567mm^2	26mm^2	263mm^2
Number of cores	18	8	1	4
Cores-to-die area ratio	34%	32%	37%	37%

Table 2: Area Scaling Compared to Previous Generation Technologies

	IBM POWER7+	Intel i7 2700K	Intel i7 3770K
Core execution type	Out-of-order	Out-of-order	Out-of-order
Technology node	32nm	32nm	22nm
Core area scaling from prev. gen.	0.68×	0.66×	0.66×
Number of cores	8	4	4
Cores-to-die area ratio	37%	37%	37%

We assume that the heterogeneous processor is comprised of b number of big cores, and the rest of the area is populated with $n-b\times r$ small cores. The total area is equivalent to those of homogeneous processors. A selected complex core is used to execute sequential operations, and parallel executions make use of all cores in the processor. The performance speed-up of the heterogeneous processor with multiple big cores and maximum scheduling is expressed as Eq. (4). Single-thread executions ($1-f$ part of a workload) are accelerated by a complex core that has s times greater performance than a simple core. The f fraction of the workload is parallelized by both big cores ($b\times s$ speed-up) and small cores ($n-b\times r$ speed-up).

$$E_{het:ms} = \frac{1-f}{s}p + \frac{b\times p + (n-b\times r)}{b\times s + (n-b\times r)}f \tag{5}$$

During sequential executions, the processor consumes p amount of power that is equivalent to single big-core power. Other unused cores are assumed to be ideally turned off and do not consume power. In parallel phases, total power is the sum of $b\times p$ by big cores and $(n-b\times r)\times 1$ by small cores, where the power dissipation of a small core is normalized to 1 and a big core has p times larger power. Collectively, the total energy of the heterogeneous processor with multiple big cores and maximum scheduling is calculated as Eq. (5).

3.4 Heterogeneous Processor with Dynamic Scheduling

Another possible case of utilizing the heterogeneous processor is separating the use of distinct core types. For instance, complex cores are only used to execute serial threads, and parallel operations are run only on simple cores. This type of *dynamic scheduling* is advocated especially in power-constrained processors, where big cores may not be able to run simultaneously with a group of small cores because of power limitation [6, 34]. Running parallel threads only on simple cores also solves scheduling issues caused by performance imbalance between different core types. The performance speed-up of the heterogeneous processor with multiple big cores and dynamic scheduling is calculated as Eq. (6). The sequential part ($1-f$) is accelerated by a big core that is s times faster than a small core, and the f fraction is parallelized by $n-b\times r$ number of small cores.

$$Perf_{het:ds} = \frac{1}{\frac{1-f}{s} + \frac{f}{n-b\times r}} \tag{6}$$

The total energy of the heterogeneous processor with dynamic scheduling is expressed as Eq. (7). The processor selects a big core to perform single-thread executions and consumes power p. Other cores are assumed to be power-gated or used by other applications, where the power dissipation of those cores attribute to other applications. In parallel phases, threads run only on small cores and consume $n-b\times r$ amount of power.

$$E_{het:ds} = \frac{1-f}{s}p + \frac{n-b\times r}{n-b\times r}f \tag{7}$$

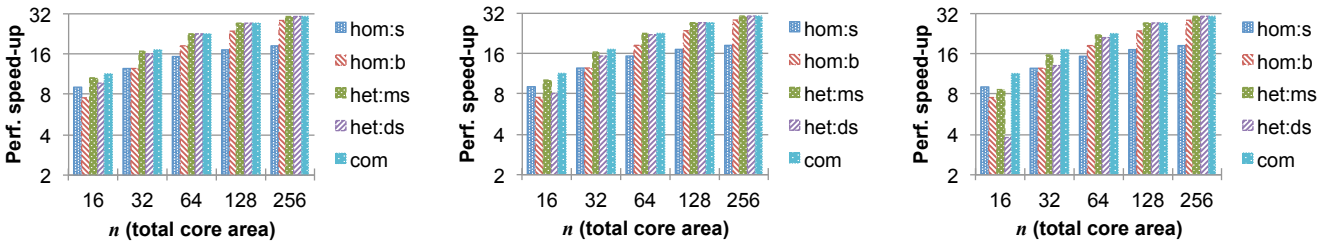
3.5 Composed Processor of Simple Cores

Hill and Marty presented a homogeneous processor model comprised of n small cores, where a set of cores are dynamically combined and help each other to speed up serial executions such as thread-level speculation or helper threads [11, 24]. It is assumed that these helper threads run on separate cores, and a set of r small cores have the same performance as one complex core denoted by s. Instead, the group of small cores consumes r amount of power that can be greater than the power of a big core p. The performance speed-up of the *composed processor* is expressed as Eq. (8), and the total energy is calculated as Eq. (9).

$$Perf_{com} = \frac{1}{\frac{1-f}{s} + \frac{f}{n}} \tag{8}$$

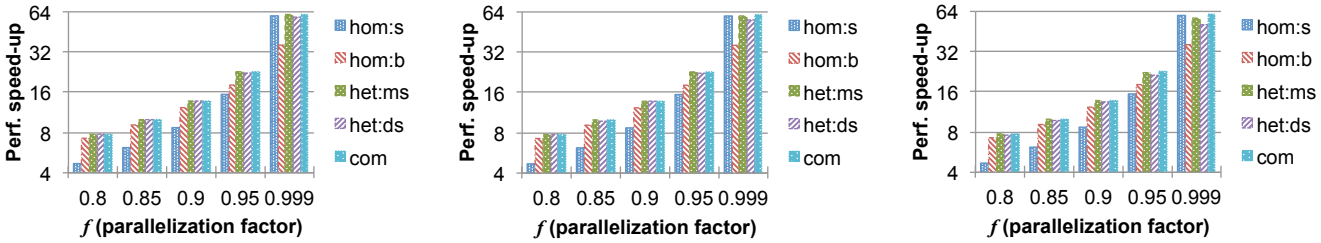
The composed processor accelerates serial executions ($1-f$) by s times, and the f part is parallelized by n small cores. This processor represents an ideal case in that it can accelerate both serial and parallelizable parts of workloads. Since r number of cores are grouped to yield s times greater performance, the processor dissipates $r\times 1$ amount of power during serial executions and $n\times 1$ in parallel phases.

$$E_{com} = \frac{1-f}{s}r + \frac{f}{n}n \tag{9}$$



(a) $f = 0.95$, $b = 1$ (b) $f = 0.95$, $b = 2$ (c) $f = 0.95$, $b = 4$

Figure 3: Maximum performance speed-up of multicore processors with parallelization factor $f = 0.95$, and varying total area (n in unit of small cores) and number of big cores (b) in the heterogeneous processors.



(a) $n = 64$, $b = 1$ (b) $n = 64$, $b = 2$ (c) $n = 64$, $b = 4$

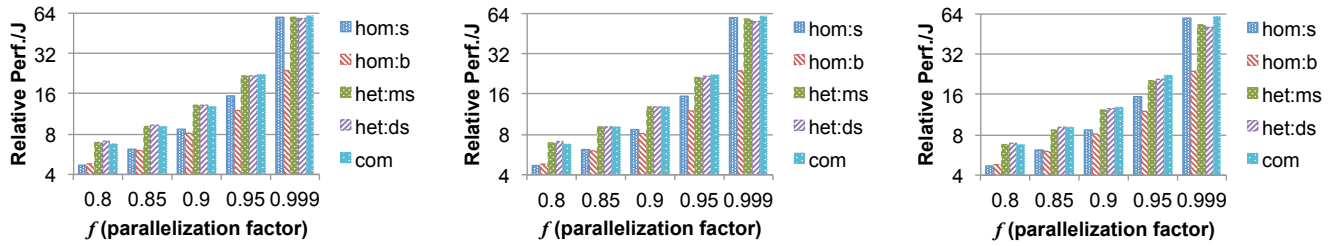
Figure 4: Maximum performance speed-up of multicore processors with $n = 64$ and parallelization fraction scaled between $f = 0.8$ and 0.999. The number of big cores (b) in the heterogeneous processor is varied in the sub-plots.

4. EVALUATING PERFORMANCE AND ENERGY EFFICIENCY SCALING OF MULTICORE MODELS

In this section, we evaluate the multicore performance and energy models presented in the previous section to correlate their impact with lifetime reliability (discussed later in the paper). An out-of-order core in general has 2-4× larger area than a comparable in-order design. Table 1 summarizes the area of simple and complex-core pairs from IBM and Intel processors, estimated from available references and die shots [4, 10, 16, 31, 37]. The area ratio between big and small cores is estimated around 2.5-4.4× for these processors. Based on these examples, we choose an integer number $r = 3$ as the area ratio to compose hypothetical heterogeneous processors. Pollack's Rule [25] states that performance and area are correlated as $s \propto \sqrt{r}$. We adopt the power expression from Chung's model [4], where power is expressed as $W \propto \sqrt{r^{\alpha}}$ and $\alpha = 1.75$. These parameters are applied to Eq. (1)-(9) for the performance and energy efficiency evaluation of multicore processors. Table 1 shows that the proportion of core area on the die is relatively consistent, ranging between 30-40% of the die. Processors at successive technology nodes in Table 2 also have similar cores-to-die area ratio. No discernible correlation is found among these cases between the area ratio and core types, number of cores, or other uncore configurations (e.g., cache sizes, on-chip network). Hence, in our analysis we focus on the core scaling factors and simplify other conditions.

Table 2 shows that core size scales by 0.66-0.68× every technology node. With continued scaling, it is predicted that there can be around a hundred simple cores within a die area similar to Intel i7 at 8nm technology node, or about twice more on a much larger IBM POWER7 die. In our analysis, we scale the number of small cores between $n = 16$ and 256 as shown in Figure 3. This figure plots the maximum performance speed-up of multicore processors with varying n and fixed $f = 0.95$. The big core count of the heterogeneous models differs in each sub-plot. When $n \gg 64$, we observe that overall speed-up is limited by sequential throughput as parallel executions become substantially short (when maximum performance increase is assumed with increasing n). Hence, the homogeneous processor composed of simple cores suffers from low performance. Little performance difference is made in the heterogeneous processors by varying the number of complex cores when n is large. On the other hand, the cases with $n \ll 64$ are more dominated by parallel performance because of narrow parallelization width; relatively longer time is spent on parallel executions. The heterogeneous processor of small n with dynamic scheduling and multiple big cores such as in Figure 3.(c) shows limited performance increase since this processor utilizes only simple cores for parallelization. Based on these observations, we choose an intermediate size of $n = 64$ as an exemplary case to study in this paper.

Figure 4 shows the performance speed-up of various multicore configurations with $n = 64$, and Amdahl's scaling factor is changed between $f = 0.8$ and 0.999. The number of complex cores in the heterogeneous designs is varied by $b = 1$, 2, and 4 in the sub-plots. The composed processor as an ideal design provides the most performance speed-up, and the heterogeneous processors also produce similar performance increases. For moderately parallelizable workloads (e.g., $f = 0.8$), the homogeneous processor of big cores shows similar performance speed-up as the heterogeneous or composed processor since good amount of time is spent on performing single-thread executions. As parallelization fraction f in-



(a) $n = 64$, $b = 1$ (b) $n = 64$, $b = 2$ (c) $n = 64$, $b = 4$

Figure 5: Relative energy efficiency (performance per Joule) of multicore processors for $n = 64$ with scaling factor between $f = 0.8$ and 0.999 and different number of big cores (b) in the heterogeneous processors.

creases, the overall performance speed-up is dominated by parallel executions. If the number of big cores is increased in the heterogeneous processors (e.g., $b = 1$ to 4), peak parallel throughput decreases particularly when $f \to 1.0$. However, increasing big core count has a minor impact if f is small. From the comparison of Figure 3.(a) and (c), big core count shows greater impact for small n ($\ll 64$) since complex cores occupy relatively large area in small-size processors. The opposite happens for large n. Thus, the effect of increasing the number of big cores is more addressed for large $f \to 1.0$ or small n.

When comparing the subplot (a) and (b) of Figure 3 and 4, there are subtle differences in performance for the heterogeneous processors by increasing the number of big cores from $b = 1$ to 2. However, increasing the big core count in heterogeneous processors makes a large difference in lifetime reliability, and this will be discussed later in the paper. Further increasing the number of complex cores such as $b = 4$ penalizes the heterogeneous processors especially when handling highly parallel workloads (e.g., $f = 0.999$).

Energy efficiency can be calculated by using Eq. (1)-(9). Following the methodology presented in the work of Woo and Lee [34], we use performance per Joule (*Perf./J*) to represent energy efficiency. Figure 5 plots the relative energy efficiency of multicore processors under the same conditions as Figure 4. In overall, the results show similar trends as those in Figure 4. The composed processor provides the most increase in energy efficiency as $f \to 1.0$. The heterogeneous processors produce similar or even better energy efficiency when parallelization fraction f is small since we assume that a group of dynamically combined small cores would dissipate more power than a single complex core despite the same performance (refer to Section 3.5). Increasing the number of big cores in the heterogeneous processors reduces peak parallel throughput especially when $f \to 1.0$. As a result, the heterogeneous processors achieve less improvement in energy efficiency when $b = 4$ and $f = 0.999$ as shown in Figure 5.(c), compared to the composed processor.

5. COMPACT THERMAL ESTIMATION

Reliability characteristics strongly depend on temperature. Therefore, assuming a constant core temperature and failure rate is not a useful approach, and we elaborate on estimating the thermal states of multicore processors for accurate reliability modeling.

Using the first-order ordinary differential equation (ODE) [9, 36], temperature is modeled as Eq. (10). $\mathbf{x}$ is the temperature vector of n equal-sized blocks (e.g., small cores as floorplans) at time i ($i = 0$ is an initial state), and $\mathbf{A}$ matrix shows heat spreading process. Matrix $\mathbf{B}$ includes conversion factors from power input $\mathbf{u}$ to temperature $\mathbf{x}$, where an entry of the matrix B_{jk} denotes the conversion rate of power dissipation at location j to thermal increase at location k. The vector $\mathbf{w}$ is an effect due to ambient temperature.

$$\mathbf{x}(i) = \mathbf{A}^i\mathbf{x}(0) + (\mathbf{A}-\mathbf{I})^{-1}(\mathbf{A}^i-\mathbf{I})\mathbf{B}\mathbf{u} + \mathbf{w} \tag{10}$$

In this analysis, we estimate thermal effects based on a steady-state model since lifetime reliability is governed by long-term behaviors. Steady-state temperature is denoted by the vector $\mathbf{x}$ when $i \to \infty$ as shown in Eq. (11). $\mathbf{x}'$ means $\mathbf{x}(\infty)$, and $\mathbf{A}^i \to 0$ in Eq. (10) for $i \to \infty$.

$$\mathbf{x}' = (\mathbf{A}-\mathbf{I})^{-1}\mathbf{B}\mathbf{u} + \mathbf{w} \tag{11}$$

Assume that vector $\mathbf{x}'$ is the temperature vector of the homogeneous processor of small cores with 100% parallel executions, where all cores are active and each small core has the normalized power of 1 (refer to Section 3.1). We use this thermal state vector $\mathbf{x}'$ as a *baseline state* and analyze how different processor compositions (e.g., heterogeneous) or execution phases (e.g., single-thread executions) create thermal differences to the baseline. The purpose of compact thermal modeling is to estimate temperature difference to the baseline state rather than calculating absolute magnitude of temperature. By substituting a portion of the small-core die with complex cores, it creates changes to the power distribution that is expressed as $\Delta\mathbf{u}$. Power and thermal changes are also created within a processor depending on execution modes (e.g., serial or parallel phases). In any scenarios, changes in power distribution ($\Delta\mathbf{u}$) result in the temperature difference of $\Delta\mathbf{x}'$ as shown in Eq. (12). $(\mathbf{A}-\mathbf{I})^{-1}\mathbf{B}$ is substituted with a matrix $\mathbf{C}$ in the equation.

$$\Delta\mathbf{x}' = (\mathbf{A}-\mathbf{I})^{-1}\mathbf{B}\Delta\mathbf{u} = \mathbf{C}\Delta\mathbf{u} \tag{12}$$

The entries of matrix $\mathbf{C}$ are steady-state power-to-thermal conversion factors between any two locations on the die. For instance, thermal change at block j is expressed as $\Delta x'_j$ that is the sum of two terms in Eq. (13). The first term in this equation, $C_{jj}\Delta u_j$, means temperature change due to the power difference (with respect to the baseline) at the same location. The second term, $\sum C_{kj}\Delta u_k$, is the thermal contribution to lo-

Table 3: Validation of Compact Thermal Estimation

Processor type	Max difference (°C) to HotSpot model			
	Sequential		Parallel	
	BC	SC	BC	SC
Homogeneous: small cores	N/A	-0.41	N/A	Baseline
Homogeneous: big cores	+0.63	N/A	+0.45	N/A
Heterogeneous: max. sch.	+0.63	Unused	-0.19	-0.01
Heterogeneous: dyn. sch.	+0.63	Unused	Unused	-0.58
Composed: small cores	N/A	-0.32	N/A	Same as baseline

cation j as a result of power changes at other locations $k \neq j$.

$$\Delta x'_j = C_{jj}\Delta u_j + \sum_{k\neq j}^{n} C_{kj}\Delta u_k \quad (13)$$

The matrix **C** depends not only on the thermal properties of package but also floorplanning of multicore die. For a hypothetical heterogeneous processor without known floorplanning (or many possible combinations of floorplanning), we attempt to estimate thermal behaviors by using a scalar model as in Eq. (14). In this equation, the thermal change of block j is primarily induced by power difference at the same location (Δu_j) multiplied by conversion factor C_{jj}. Thermal impact by other blocks to location j is estimated by calculating the average of power change $\Delta\bar{u}$ multiplied by scaling factor $\bar{C}_{kj}$. These scaling factors can be obtained from thermal models (e.g., HotSpot [12]) by varying power input at location j and measuring thermal changes at location j for C_{jj} and $k \neq j$ for $\bar{C}_{kj}$ (average for all k).

$$\Delta x'_j = C_{jj}\Delta u_j + \bar{C}_{kj}\Delta\bar{u} \quad (14)$$

When block j belongs to a big core, we use $\Delta\bar{u}_b$ in Eq. (15) and $\Delta\bar{u}_s$ for a small core in Eq. (16) to represent $\Delta\bar{u}$ of Eq. (14). Using these equations, it becomes possible to estimate thermal differences between heterogeneous and baseline homogeneous processors, as well as thermal changes between execution phases within a processor.

$$\Delta\bar{u}_b = \frac{(b-1)r}{n-r}\delta(p_b-1) + \frac{n-b\times r}{n-r}\delta(p_s-1) \quad (15)$$

$$\Delta\bar{u}_s = \frac{b\times r}{n-1}\delta(p_b-1) + \frac{n-1-b\times r}{n-1}\delta(p_s-1) \quad (16)$$

In Eq. (15), $\Delta\bar{u}_b$ means the power contribution of all other cores (at location $k \neq j$) to the temperature of a big core that spans over location j. These other cores include $b-1$ number of big cores and $n-b\times r$ small cores. In the first term of Eq. (15), $(b-1)r/(n-r)$ is the area fraction of big cores over total core area except one big core at location j. When $b=1$, it means that there are no other big cores that affect the temperature of the only big core in the heterogeneous processor. In the area of other big cores expressed as $(b-1)r$, it has the power density difference of p_b-1 compared to that of a small core. p_b is the relative power of a big core, which is $p_b = p/r$ when the core is active or $p_b = 0$ when power-gated. δ is the power density of a small core (in W/cm^2) of the baseline state. In the second term of $\Delta\bar{u}_b$, $(n-b\times r)/(n-r)$ is the area fraction of small cores over the total core area. p_s is the normalized power of a small core that is $p_s = 1$ at active state or $p_s = 0$ when turned off. Similarly, $\Delta\bar{u}_s$ in Eq. (16) is the power contribution of all other cores to the temperature of a small core at location j.

Table 4: Failure Models [17, 33]

Failure types	Description and models
Hot carrier injection (HCI)	Particles that gain sufficient kinetic energy overcome the barrier to gate oxide and cause degradation. $\text{MTTF}_{HCI} = A_{HCI}\, I_{sub}^{-n}\, e^{\frac{E_{a\,HCI}}{kT}}$ (17) A_{HCI} = technology-dependent constant; I_{sub} = substrate current; n = acceleration factor; T = absolute temperature; $E_{a\,HCI}$ = activation energy; k = Boltzmann's constant
Negative bias temperature instability (NBTI)	PMOS devices under negative gate voltage at elevated temperature cause threshold voltage shift and timing errors. $\text{MTTF}_{NBTI} = A_{NBTI}\, V_{gs}^{-r}\, e^{\frac{E_{a\,NBTI}}{kT}}$ (18) A_{NBTI} = process-related constant; V_{gs} = gate voltage; r = voltage acceleration factor; $E_{a\,NBTI}$ = activation energy

Table 3 shows the accuracy of our compact thermal estimation compared to a HotSpot steady-state model [12] after calibration. We created homogeneous and heterogeneous floorplans under the area constraint of $n = 64$. In the heterogeneous processor, a complex core is placed at the center, similar to Figure 1.(a). The multicore processors in HotSpot simulations show as large as 20°C temperature variations between execution phases or core types. Hence, disregarding thermal effects will lead to significant inaccuracy in lifetime reliability modeling. In overall, our thermal estimation yields less than 1°C difference to a detailed model, and it enables us to simplify theoretical analysis for modeling the lifetime reliability of heterogeneous processors. The estimated temperatures are applied to Eq. (17) and (18) to calculate resulting mean-time-to-failure (MTTF).

6. EXTENDING AMDAHL'S LAW FOR LIFETIME RELIABILITY SCALING OF MULTICORE PROCESSORS

The lifetime reliability of multicore processors is subject to Amdahl's scaling factor f, processor composition with b and n (e.g., number of big and small cores), and scheduling method (e.g., maximum or dynamic scheduling). This section presents the lifetime reliability models of multicore processors as functions of aforementioned parameters.

6.1 Failure Phenomena and Models

Gradual device degradation leads to the failure of processor components. Hot Carrier Injection (HCI) and Negative

Bias Temperature Instability (NBTI) are known to be critical failure mechanisms as device technology continues to scale. These failures are primarily caused by charges trapped in gate oxide, which result in threshold voltage shift and timing errors [17, 28]. We adopt HCI and NBTI models from the work of Kim et al. [17] and White and Bernstein [33]. Table 4 summarizes the failure models used in our study.

6.2 Modeling of Lifetime Reliability

In our analysis, we use an exponential distribution to simplify the models [27, 28, 33]. The failure rate of exponential distribution is expressed as $\lambda = 1/\text{MTTF}$. The total failure rate is calculated as the sum of failure rates (SOFR) of wear mechanisms; $\lambda = \lambda_{HCI} + \lambda_{NBTI}$. We assume that failure rate is proportional to area when stress conditions are identical, and thus the failure rate of a big core (λ_b) is r times greater than that of a small core (λ_s); $\lambda_b = r \times \lambda_s$. Reliability characteristics strongly depend on temperature as shown in Eq. (17) and (18). The thermal state of a processor differs by core composition (e.g., homogeneous or heterogeneous), execution mode (e.g., serial or parallel), and scheduling method (e.g., maximum or dynamic scheduling in the heterogeneous processor). In this section, we present how the energy models in Section 3 are translated to the thermal states and eventually failure rates of heterogeneous cores across different execution phases.

6.3 Homogeneous Processor of Simple Cores

We use the reliability state (i.e., failure rates) of a homogeneous processor of simple cores with 100% parallel execution as a *baseline* in our analysis. It is assumed that each simple core has the normalized power of 1 and failure rate of λ_s. The total failure rate of the processor is calculated as Eq. (19), and MTTF is expressed as $1/\lambda_{hom:s}$. Any small cores on the die can be selected to execute the serial part $1-f$ of a workload. The long-term reliability impact of the core executing sequential operations is divided by the number of cores n. It is assumed that unused cores are power-gated and have no increase of failure rates in the mean time. $\lambda_{s:seq}$ is the failure rate of a small core in serial phases at lower operating temperature. Eq. (14) enables us to estimate the temperature difference of the active core executing a serial thread with respect to the baseline. The estimated thermal difference ($x'_j - \Delta x'_j$) is applied to Eq. (17) and (18) to calculate the changes in failure rates and resulting MTTF. During parallel executions, the performance improvement (f/n) is offset by correspondingly larger total failure rate ($n \times \lambda_{s:par}$) according to the SOFR. $\lambda_{s:par}$ is the baseline failure rate normalized to 1 in our analysis.

$$\lambda_{hom:s} = (1-f)\frac{\lambda_{s:seq}}{n} + \frac{f}{n} \times n \times \lambda_{s:par} \tag{19}$$

6.4 Homogeneous Processor of Complex Cores

In a homogeneous processor consisting of complex cores under the same area as the one of simple cores, there can be n/r number of big cores. Applying $b = n/r$ to Eq. (15), $\Delta\bar{u}_b$ becomes $\delta(p_b - 1)$. When $p/r < 1$, it results in lower power density than a simple core and hence reduces the failure rate per unit area because of lower operating temperature. The failure rate of a big core is calculated as $\lambda_b = \sum \lambda_j$ for all j belonging to the big core area. For each λ_j, thermal difference to the baseline is estimated by using Eq. (14). The total failure rate of the homogeneous processor with complex cores is calculated as in Eq. (20).

$$\lambda_{hom:b} = \frac{1-f}{s} \times \frac{r}{n} \times \lambda_{b:seq} + \frac{f}{s \times n/r} \times \frac{n}{r} \times \lambda_{b:par} \tag{20}$$

In this equation, the serial part $(1-f)$ can be executed by any big cores. $\lambda_{b:seq}$ is the failure rate of a big core during sequential phases. Since any complex core on the die can be chosen to handle serial operations, long-term reliability impact is divided by the number of cores (n/r). The performance increase of parallel executions ($s \times n/r$) is offset by the sum of failure rates of n/r big cores.

6.5 Heterogeneous Processor with Maximum Scheduling

Complex cores in a heterogeneous processor with maximum scheduling are the busiest computing units. One of the complex cores has to execute the serial part of an application, and they also participate in executing parallel threads to maximize performance. With b number of big cores in the heterogeneous processor, the total failure rate is calculated as in Eq. (21).

$$\lambda_{het:ms} = \frac{1-f}{s} \times \frac{\lambda_{b:seq}}{b} + \frac{f}{b \times s + (n - b \times r)} \times \left\{ b \times \lambda_{b:par} + (n - b \times r)\lambda_{s:par} \right\} \tag{21}$$

Any one of the complex cores in the heterogeneous processor can execute the serial part $1-f$ of a workload. The long-term reliability impact of the big core ($\lambda_{b:seq}$) in sequential phases is reduced b fold. During parallel executions, the total failure rate is calculated as $b \times \lambda_{b:par} + (n - b \times r)\lambda_{s:par}$, where the failure rate of each core type is multiplied by the core count of corresponding type. When a processor failure happens, the probability that the fault is caused by big cores is calculated as the failure rate of big cores over the total failure rate ($\lambda_{het:ms}$) as shown in Eq. (22). The reliability of big cores becomes more critical for small b or f since more stresses are put on them.

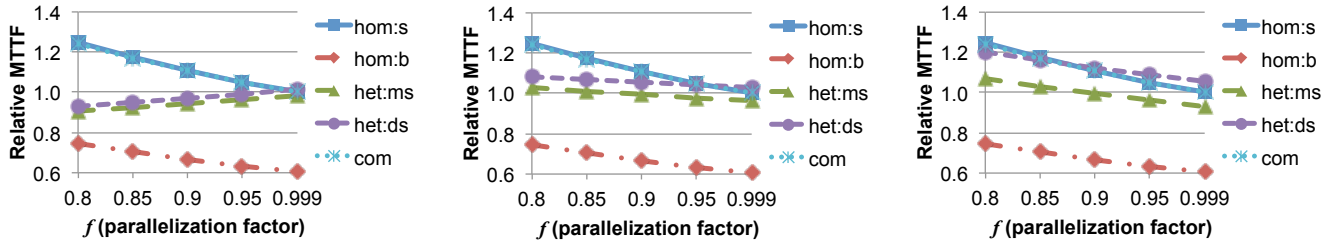
$$Prob_b = \frac{\frac{1-f}{s} \times \frac{\lambda_{b:seq}}{b} + \frac{f \times b \times \lambda_{b:par}}{b \times s + (n - b \times r)}}{\lambda_{het:ms}} \tag{22}$$

6.6 Heterogeneous Processor with Dynamic Scheduling

In a heterogeneous processor with dynamic scheduling, distinct types of cores are used to handle different phases of applications. By turning off unused cores, this scheduling policy benefits from improved reliability that is traded with performance degradation. The total failure rate of the processor is expressed as Eq. (23). The first term in this equation reflects the reliability impact of big cores during sequential operations, and the second term is the failure rate of small cores in parallel phases.

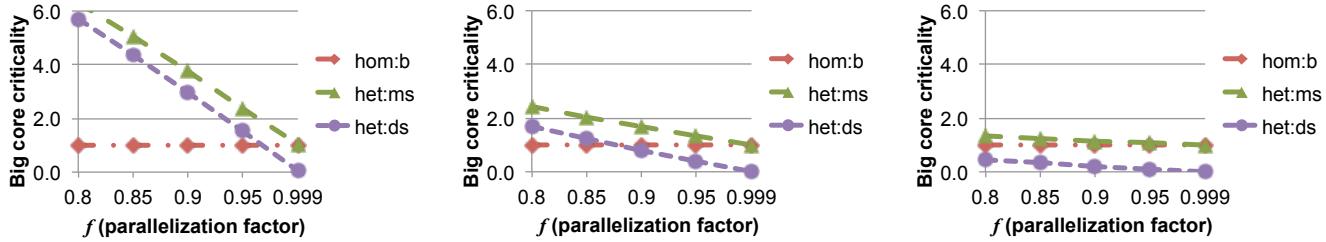
$$\lambda_{het:ds} = \frac{1-f}{s} \times \frac{\lambda_{b:seq}}{b} + \frac{f}{n - b \times r}(n - b \times r)\lambda_{s:par} \tag{23}$$

The probability that a processor failure is caused by big cores is calculated as Eq. (24) that is the failure rate of big cores over the total failure rate ($\lambda_{het:ds}$). Since any one of



(a) $n = 64$, $b = 1$ (b) $n = 64$, $b = 2$ (c) $n = 64$, $b = 4$

Figure 6: Relative lifetime (MTTF) of various multicore models for $n = 64$ with parallelization fraction scaled between $f = 0.8$ and 0.999, and varying number of big cores (b) in the heterogeneous processors.



(a) $n = 64$, $b = 1$ (b) $n = 64$, $b = 2$ (c) $n = 64$, $b = 4$

Figure 7: Relative reliability criticality of big cores in the heterogeneous multicores for $n = 64$ with parallelization factor between $f = 0.8$ and 0.999. The number of big cores (b) is varied in the sub-plots.

big cores can be used to execute serial threads, the reliability criticality is significantly reduced by increasing b.

$$Prob_b = \frac{\frac{1-f}{s} \times \frac{\lambda_{b:seq}}{b}}{\lambda_{het:ds}} \tag{24}$$

6.7 Composed Processor of Simple Cores

Composed processor is technically a homogeneous processor comprised of simple cores. The only difference to the conventional homogeneous model is that multiple small cores are grouped to speed up the sequential operations. The total failure rate of the processor is shown in Eq. (25). Although this processor utilizes multiple cores in the serial phase of $1-f$, the reliability impact is minor because any r number of small cores among n can be chosen. For parallel executions, the failure rate is calculated in the same way as the homogeneous processor of simple cores.

$$\lambda_{com} = \frac{1-f}{s} \times \frac{r \times \lambda_{s:seq}}{n} + \frac{f}{n} \times n \times \lambda_{s:par} \tag{25}$$

7. EVALUATING LIFETIME RELIABILITY SCALING OF MULTICORE MODELS

In this section, we evaluate the lifetime reliability models of multicore processors presented in the previous section. Figure 6 shows the relative MTTF of multicore models ($n = 64$) with Amdahl's scaling factor varied between $f = 0.8$ and 0.999. The number of big cores in the heterogeneous processors is changed by $b = 1$, 2, and 4 in the sub-plots. The baseline MTTF ($= 1.0$) is when the homogeneous processor of small cores is operating at 100% parallel executions. The MTTF curves of the homogeneous processor with simple cores are located above the MTTF $= 1.0$ line because of the serial part $1-f$ that exercises only one simple core. Activating one core also produces better thermal behaviors, so the failure rate of a core in serial phases is lower than that during parallel executions. More importantly, any cores in the homogeneous processor can be selected to perform serial operations (i.e., load sharing effect in the long term), so the serial phases are insignificant from the reliability perspective. The similar phenomena happen in the composed processor. For the same operating conditions, the homogeneous processor of big cores exhibits worse reliability since it spends relatively longer time on parallel phases that turn on all cores.

When the heterogeneous processor includes only one complex core, MTTF decreases with smaller f (i.e., more serial operations) as shown in Figure 6.(a) as opposed to the reliability behaviors of homogeneous processors. The reason can be explained with Figure 7.(a) that plots the failure rate ratio between a big and small core per unit area. Since the only complex core in the heterogeneous processor has to handle all the sequential executions, increasing serial fraction of $1-f$ puts more stresses on the complex core. The problem is especially worse with the maximum scheduling in that the big core has to execute both serial and a part of parallel operations. In particular, the big core in the heterogeneous processor with maximum scheduling at $f = 0.8$ is $6.3 \times r$ times more likely to fail than a small core; for $r = 3$, it has about $19\times$ greater failure rate. Although a big core takes only a small portion of the total area when $n = 64$, uneven failure rate distribution between core types limits the overall lifetime of the heterogeneous processor.

If the heterogeneous processor includes multiple big cores, the reliability criticality of big cores is significantly reduced because of load sharing effect as shown in Figure 7.(b). For instance, the heterogeneous processor with maximum schedul-

Table 5: Simulation Setup

Configurations	Description	
Core type	Complex core	Simple core
Issue width	6	1
ROB size	128 entries	N/A
L1 cache size per core	32KB, 4-way assoc.	
L2 cache size per core	256KB, 8-way assoc.	
Clock frequency	2.0GHz	
Memory bandwidth	25GB/s	

ing shows 13% improvement in lifetime and 17% for dynamic scheduling at $f = 0.8$ by adding one more complex core, by comparing Figure 6.(a) and (b). However, further increasing the number of big cores, $b = 4$ for example, rather diminishes the lifetime of heterogeneous processor with maximum scheduling for highly parallel workloads (e.g., $f = 0.999$) as shown in Figure 6.(c). Increasing the big core count decreases peak parallel throughput, and it causes the processor to operate longer time in parallel phases. As the stresses shift from the big cores to simple cores by increasing b, including many big cores has a negative impact on reliability. The reliability of heterogeneous processor with dynamic scheduling continues to benefit from increasing the number of complex cores, but the similar effect happens when large number of complex cores are incorporated. Consequently, including a few number of big cores helps improve processor lifetime, but adding large number of big cores has an adverse impact on reliability.

8. APPLICATION TO LIFETIME RELIABILITY, PERFORMANCE, AND ENERGY EFFICIENCY TRADEOFFS

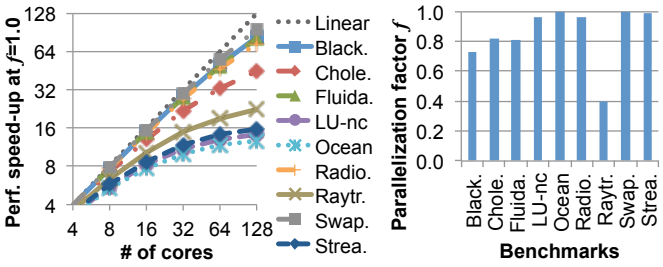
In this section, we apply the performance, energy, and reliability models presented in the previous sections to the simulation results of real benchmarks. We used Manifold microarchitecture simulator [32] to collect performance counters of PARSEC and SPLASH-2 benchmarks [2] and McPAT [20] to estimate the power of exemplary complex and simple core models. The simulation outputs are extrapolated to construct hypothetical multicores, and the results are discussed. Table 5 summarizes the simulation setup used in the experiment.

8.1 Realistic Performance and Energy Models

We consider a more realistic performance model adopted from the work of Esmaeilzadeh et al. [6, 8] instead of maximum performance speed-up assumed in Section 3. Eq. (26) shows that the performance scaling of a multicore processor is bounded by core throughput or memory bandwidth. In this equation, N is throughput-equivalent core count in unit of small cores, and CPI_{active} is instruction latency during active periods. η represents core utilization factor to keep the core pipeline busy without stalls. BW_{mem} is maximum memory bandwidth, and γ_{mem} is the rate of memory instructions with cache miss rates $\prod m$ that require off-chip memory access of data width d_{mem}.

$$Perf = min\left(N\frac{freq}{CPI_{active}}\eta, \frac{BW_{mem}}{\gamma_{mem}\times\prod m\times d_{mem}}\right) \quad (26)$$

Figure 8.(a) plots the estimated performance speed-up of individual benchmarks when assuming that Amdahl's scaling



(a) (b)

Figure 8: Benchmark characterization: (a) performance speed-up with increasing number of cores when assuming 100% parallel executions and (b) actual parallelization fraction f of individual benchmarks obtained from microarchitectural simulations.

factor is $f = 1.0$ (i.e., perfectly parallelizable); performance speed-up is normalized to the throughput of each benchmark at $n = 1$. Figure 8.(b) shows actual Amdahl's scaling factor f of individual benchmarks. From the results in Figure 8, benchmarks can be categorized into the following classes:

- Class I: Blackscholes and Fluidanimate are highly scalable (i.e., good performance speed-up with increasing number of cores) but poorly parallelizable applications (i.e., relatively small parallelization fraction f).
- Class II: Ocean-nc and Streamcluter benchmarks are less scalable (i.e., saturating performance with increasing number of cores) but highly parallelizable (i.e., $f \approx 1.0$).
- Class III: Swaptions benchmark shows great multicore scalability and is also highly parallelizable.
- Class IV: Raytrace benchmark represents poorly scaled and highly serial applications.
- Class V: Other benchmarks show intermediate features.

From the simulation results, area ratio between exemplary complex and small cores was estimated around $r = 3.2$. On average, we obtained performance ratio between two core types around $s = 0.96\times\sqrt{r}$ compared to Pollack's Rule [25] and power ratio as $p = 0.94\times\sqrt{r}^{\alpha}$ with respect to Chung's model [4]. When calculating energy efficiency, we adjusted the dynamic power of cores with the utilization factor η in Eq. (26). Leakage power was repeatedly calculated based on the compact thermal estimation until they converge.

8.2 Application to the performance, energy efficiency, and lifetime reliability models

Figure 9.(a) shows the performance of various multicore processors with different applications. Processor size is fixed at $n = 64$, and two big cores ($b = 2$) are included in the heterogeneous designs. Amdahl's scaling factor f is variant across benchmarks. For each benchmark, the results of multicore processors are normalized to that of the composed processor option. In most cases, the heterogeneous and composed processors (1.0 line in the graph) outperform homogeneous implementations, producing higher throughput for both sequential and parallel executions except for the Class III applications. The Class III workloads are highly scalable and

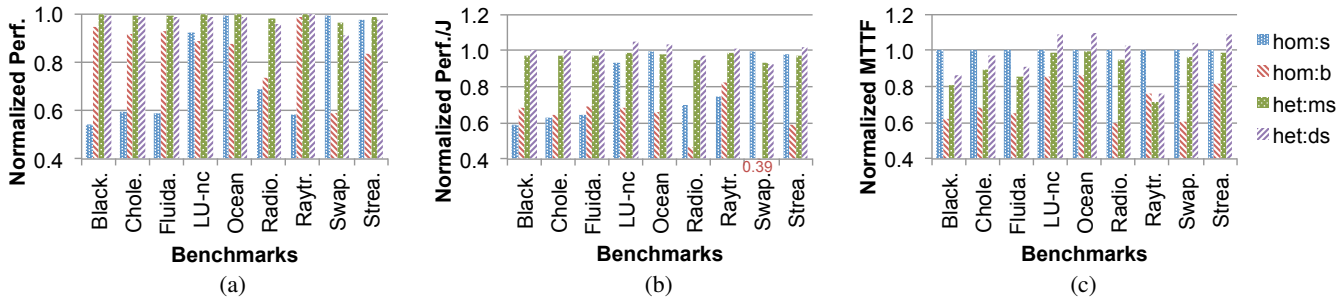


Figure 9: (a) Normalized performance, (b) energy efficiency, and (c) lifetime reliability of multicore processors for $n = 64$, $b = 2$, and $f = varying$. The result of each benchmark is normalized to the composed processor option.

parallelizable, so they favor the homogeneous processor of small cores. Incorporating more big cores in the heterogeneous processors reduces peak parallel throughput and thus diminishes overall performance, compared to the composed processor made of simple cores. In contrast, the performance of the Class IV workloads is dominated by sequential executions. The Class III and IV show exactly opposite performance behaviors with the homogeneous processors. For the Class I type of applications (i.e., highly scalable but less parallelized), overall performance speed-up is more governed by single-thread executions. The homogeneous processor comprised of small cores in this case shows inferior performance to other multicore configurations. The Class II benchmarks have large Amdahl's scaling factor ($f \approx 1.0$) but low multicore scalability. The homogeneous processor of big cores shows 13-17% lower performance than other multicore processors, but the difference is limited because the applications do not scale well with large number of cores.

The energy efficiency of multicore processors with different classes of benchmarks is shown in Figure 9.(b). The heterogeneous and composed processors have superior energy efficiency to the homogeneous configurations except for the Class III applications. The heterogeneous processor with maximum scheduling in overall shows 2-8% lower energy efficiency than the composed processor. The difference is caused by big cores that exhibit lower efficiency. The dynamic scheduling shows higher energy efficiency than the maximum scheduling. It particularly performs well with the Class II workloads (i.e., saturating performance with increasing number of cores) since turning off complex cores during parallel executions has a minor impact on performance when $n = 64$. The homogeneous processor of complex cores shows especially low energy efficiency, but it can be better than the other homogeneous option if workloads consist of large fragment of serial executions such as the Class I and IV. In sum, the heterogeneous processors produce similar performance speed-up and energy efficiency across different classes of applications to the composed processor that represents an ideal implementation.

Figure 9.(c) shows the normalized lifetime reliability of various multicore options. The homogeneous processor made of complex cores exhibits inferior lifetime reliability to other multicores since it spends relatively longer time on executing parallel threads that require activating all cores in the processor. Notably, the heterogeneous processors have a serious reliability drawback when workloads are dominated by serial operations (i.e., Class I and IV). For these type of applications, stresses are excessively biased to the big cores. When the heterogeneous processor with maximum scheduling includes only one complex core instead of two, there is about 10% decrease in processor lifetime for the Class I benchmarks. The decrease becomes greater if applications are more bounded by serial executions. Thus, the simulation results demonstrate that the lifetime reliability of heterogeneous processors can be limited by complex cores.

9. CONCLUSION

Microarchitectural heterogeneity has drawn attentions to enhance performance and energy efficiency. In this paper, we presented theoretical models to understand the lifetime reliability of heterogeneous multicores based on Amdahl's Law, extending prior work for performance and energy models. Importantly, the performance, energy efficiency, and lifetime reliability of heterogeneous processors are correlated as a function of processor size (n in unit of small cores), big core count (b), and Amdahl's scaling factor (f). The following summarizes key insights obtained from the analysis:

- *When b/n ratio is small (e.g., one complex core and many small cores), this puts biased stresses on complex cores and thus is unfavorable for lifetime reliability especially when $f \ll 1.0$.*
- *Increasing processor size n (but fixed b) or decreasing Amdahl's scaling factor f shifts stresses from small cores to big cores in heterogeneous processors and causes the biased stress problem.*
- *When n is sufficiently large, adding a few big cores to a heterogeneous processor increases b/n, but the change is limited and therefore has a minor impact on performance and energy efficiency. In this case, increased b significantly reduces the reliability criticality of big cores and improves processor lifetime.*
- *However, further increasing the b/n ratio (i.e., more area dedicated to big cores) reduces peak parallel throughput, and extended execution time diminishes energy efficiency as well as reliability especially when $f \rightarrow 1.0$.*
- *For highly parallelizable workloads ($f \approx 1.0$), minimizing b/n ratio (e.g., only one complex core in the heterogeneous processor) becomes a preferable option.*

10. REFERENCES

[1] G. Amdahl, "Validity of the single processor approach to achieving large scale computing capability," *AFIPS Spring Joint Computer Conf.*, pp. 483-485, 1967.

[2] C. Bienia, S. Kumar, and K. Li, "PARSEC vs SPLASH-2: a quantitative comparison of two multithreaded benchmark suites on chip-multiprocessors," *IEEE Int. Symp. on Workload Characterization*, pp. 47-56, Sep. 2008.

[3] T. Cao, S. Blackburn, T. Gao, and K. McKinley, "The yin and yang of power and performance for asymmetric hardware and managed software," *Int. Symp. on Computer Architecture*, pp. 225-236, Jun. 2012.

[4] E. Chung, P. Milder, J. Hoe, and K. Mai, "Single-chip heterogeneous computing: does the future include custom logic, FPGAs, and GPGPUs?," *IEEE/ACM Int. Symp. on Microarchitecture*, pp. 225-236, Dec. 2010.

[5] A. Coskun, R. Strong, D. Tullsen, and T. Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," *Int. Conf. on Measurement and Modeling of Computer Systems*, pp. 169-180, Jun. 2009.

[6] H. Esmaeilzadeh, E. Blem, R. Amant, K. Sankaralingam, and D. Burger, "Dark silicon and the end of multicore scaling," *Int. Symp. on Computer Architecture*, pp. 365-376, Jun. 2011.

[7] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Maestro: orchestrating lifetime reliability in chip multiprocessors," *Int. Conf. on High Performance Embedded Architectures and Compilers*, pp. 186-200, Jan. 2010.

[8] Z. Guz, E. Bolotin, I. Keidar, A. Kolodny, A. Mendelson, and U. Weiser, "Many-core vs many-thread machines: stay away from the valley," *IEEE Computer Architecture Lett.*, vol. 8, no. 1, pp. 25-28, Jan. 2009.

[9] Y. Han, I. Koren, and C. Krishna, "TILTS: a fast architecture-level transient thermal simulation method," *Journal of Low Power Electronics*, vol. 3, no. 1, pp. 13-21, Apr. 2007.

[10] R. Haring, M. Ohmacht, T. Fox, M. Gschwind, D. Satterfield, K. Sugavanam, P. Coteus, P. Heidelberger, M. Blumrich, R. Wisniewski, A. Gara, G. Chiu, P. Boyle, N. Christ, and C. Kim, "The IBM Blue Gene/Q compute chip," *IEEE Micro*, vol. 32, no. 2, pp. 48-60, Dec. 2011.

[11] M. Hill and M. Marty, "Amdahl's Law in the multicore era," *The Computer*, vol. 41, no. 7, pp. 33-38, Jul. 2008.

[12] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotSpot: a compact thermal modeling methodology for early-stage VLSI design," *IEEE Trans. on Very Large Scale Integration Systems*, vol. 14, no. 5, pp. 501-513, Nov. 2006.

[13] L. Huang, F. Yuan, and Q. Xu, "Lifetime reliability-aware task allocation and scheduling for MPSoC Platforms," *Design, Automation & Test in Europe Conf. & Exhibit.*, pp. 51-56, Apr. 2009.

[14] L. Huang and Q. Xu, "Characterizing the lifetime reliability of manycore processors with core-level redundancy," *IEEE/ACM Int. Conf. on Computer-Aided Design*, pp. 680-685, Nov. 2010.

[15] J. Joao, M. Suleman, O. Mutlu, and Y. Patt, "Utility-based acceleration of multithreaded applications on asymmetric CMPs," *Int. Symp. on Computer Architecture*, pp. 154-165, Jun. 2013.

[16] R. Kalla, B. Sinharoy, W. Starke, and M. Floyd, "POWER7: IBM's next-generation server processor," *IEEE Micro*, vol. 30, no. 2, pp. 7-15, Apr. 2010.

[17] H. Kim, A. Vitkovskiy, P. Gratz, and V. Soteriou, "Use it or lose it: wear-out and lifetime in future chip multiprocessors," *IEEE/ACM Int. Symp. on Microarchitecture*, pp. 136-147, Dec. 2013.

[18] D. Koufaty, D. Reddy, and S. Hahn, "Bias scheduling in heterogeneous multi-core architectures," *European Conf. on Computer Systems*, pp. 125-138, Apr. 2010.

[19] R. Kumar, D. Tullsen, N. Jouppi, and P. Ranganathan, "Heterogeneous chip multiprocessors," *IEEE Micro*, vol. 38, no. 11, pp. 32-38, Nov. 2005.

[20] S. Li, J. Ahn, R. Strong, J. Brockman, D. Tullsen, and N. Jouppi, "McPAT: Integrated power, area, timing modeling framework for multicore architectures," *IEEE/ACM Int. Symp. on Microarchitecture*, pp. 469-480, Dec. 2009.

[21] Z. Lu, J. Lach, M. Stan, and K. Skadron, "Improved thermal management with reliability banking," *IEEE Micro*, vol. 25, no.6, pp. 40-49, Dec. 2005.

[22] P. Mercati, A. Bartolini, F. Paterna, T. Rosing, and L. Benini, "Workload and user experience-aware dynamic reliability management in multicore processors," *Design Automation Conf.*, pp. 1-6, Jun. 2013.

[23] T. Morad, U. Weiser, A. Kolodny, M. Valero, and E. Ayguade, "Performance, power efficiency and scalability of asymmetric cluster chip multiprocessors," *IEEE Computer Architecture Lett.*, vol. 5, no. 1, pp. 14-17, Jan. 2006.

[24] O. Mutlu, J. Stark, C. Wilkerson, and Y. Patt, "Runahead execution: an alternative to very large instruction window for out-of-order processors," *Int. Symp. on High Performance Computer Architecture*, pp. 129-140, Feb. 2003.

[25] F. Pollack, "New microarchitecture challenges in the coming generations of CMOS process technologies," *ACM/IEEE Int. Symp. on Microarchitecture*, 1999.

[26] W. Song, S. Mukhopadhyay, and S. Yalamanchili, "Architectural reliability: lifetime reliability characterization and management of many-core processors," *IEEE Computer Architecture Lett.*, Jul. 2014.

[27] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The case for lifetime reliability-aware microprocessors," *Int. Symp. on Computer Architecture*, pp. 276-287, Jun. 2004.

[28] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "Lifetime reliability: toward an architectural solution," *IEEE Micro*, vol. 25, no. 3, pp. 70-80, May 2005.

[29] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "Exploiting structural duplication for lifetime reliability enhancement," *Int. Symp. on Computer Architecture*, pp. 520-531, Jun. 2005.

[30] M. Suleman, O. Mutlu, M. Qureshi, and Y. Patt, "Accelerating critical section execution with asymmetric multi-core architectures," *Int. Conf. on Architectural Support for Programming Languages and Operating Systems*, pp. 253-264, Mar. 2009.

[31] E. Totoni, B. Behzad, S. Ghike, and J. Torrellas, "Comparing the power and performance of Intel's SCC to state-of-the-art CPUs and GPUs," *IEEE Int. Symp. on Performance Analysis of Systems and Software*, pp. 78-87, Mar. 2012.

[32] W. Wang, J. Beu, R. Bheda, T. Conte, Z. Dong, C. Kersey, M. Rasquinha, G. Riley, W. Song, H. Xiao, P. Xu, and S. Yalamanchili, "Manifold: a parallel simulation framework for multicore systems," *IEEE Int. Symp. on Performance Analysis of Systems and Software*, pp. 106-115, Mar. 2014.

[33] M. White and J. Bernstein, "Microelectronics Reliability: Physics-of-failure based modeling and lifetime evaluation," JPL Publication 08-5 2/08, NASA Jet Propulsion Laboratory, 2008.

[34] D. Woo and H. Lee, "Extending Amdahl's Law for energy-efficient computing in the many-core era," *The Computer*, vol. 41, no. 12, pp. 24-31, Dec. 2008.

[35] J. Yu, W. Zhou, Y. Yang, X. Zhang, and Z. Yu, "Many-core processors granularity evaluation by considering performance, yield, and lifetime reliability," *IEEE Trans. on Very Large Scale Integration Systems*, Oct. 2014.

[36] F. Zanini, D. Atienza, A. Coskun, and G. Micheli, "Optimal multi-processor SoC thermal simulation via adaptive differential equation solvers," *IFIP Int. Conf. on Very Large Scale Integration*, pp. 139-146, Oct. 2009.

[37] V. Zyuban, S. Taylor, B. Christensen, A. Hall, C. Gonzalez, J. Friedrich, F. Clougherty, Tetzloff, and R. Rao, "IBM POWER7+ design for higher frequency at fixed power," *IBM Journal of Research and Development*, vol. 57, no. 6, pp. 1:1-1:18, Dec. 2013.