

# Olay: Combat the Signs of Aging with Introspective Reliability Management

Shuguang Feng      Shantanu Gupta      Scott Mahlke

Advanced Computer Architecture Laboratory  
University of Michigan  
Ann Arbor, MI 48109  
{shoe, shangupt, mahlke}@umich.edu

## ABSTRACT

Aggressive technology scaling has historically been the driving force behind dramatic performance gains in the microprocessor industry. However, as CMOS feature sizes venture deep into the nanometer regime, reliability is emerging as a first-order design constraint. Well recognized wearout mechanisms including negative-bias temperature instability and time-dependent dielectric breakdown will plague future processors, and if left unchecked, can severely reduce operating life and performance. This paper presents an introspective reliability management system for future chip multiprocessors (CMPs), Olay, positioned to meet these reliability challenges head-on.

Olay employs low-level sensors to monitor the condition of a CMP as it ages. Sensor feedback is continuously synthesized to maintain an accurate model of damage inside the CMP, a result of both process variation and wearout degradation. Leveraging this real-time assessment of CMP health, Olay can identify reliability-aware job assignments. By exploiting the natural variation in workloads, jobs can be intelligently assigned to cores in a manner that minimizes the impact on lifetime reliability. These reliability-aware job schedules result in CMPs that can perform on average over 20% more useful work before succumbing to failures than those that rely on naive round-robin assignment policies.

## 1. INTRODUCTION

In recent years computer architects have accepted the fact that transistors become less reliable with each new technology node [10, 7]. With an exponential dependence on temperature, the frequency of faults due to failure mechanisms like negative-bias temperature instability (NBTI) and time-dependent dielectric breakdown (TDDB) will result in ever-shrinking device lifetimes as technology scaling results in higher densities and increasing operating temperatures. Furthermore, as process variation (random and systematic) and wearout become more prominent in future technology nodes, fundamental design assumptions will no longer remain valid. The characteristics of a core on one part of a chip multiprocessor (CMP) may, due to manufacturing defects, only loosely resemble an identical core on a different part of the CMP [43, 31]. Even the behavior of the same core can be expected change over time as a result of age-dependent degradation.

In view of this uncertain landscape, researchers have proposed dynamic thermal and reliability management (DTM and DRM) techniques. Such techniques hope to glean the same performance and life-expectancy, that consumers have come to expect from processors, while hiding a processor's inherent susceptibility to failures and hotspots. Recent proposals rely on a combination of thread scheduling and DVFS to recover performance lost to process varia-

tion [41, 43], and implement intelligent thermal management policies that can extend processor life and alleviate hotspots [24, 13, 14]. There have also been efforts to design sophisticated circuits that tolerate faults [8] and adaptive pipelines with flexible timing constraints [16, 42]. However, most of these existing approaches only react to issues as they manifest [41, 42]. For instance, [43] generates the best job scheduling for a CMP that is already heavily degraded.

On the other hand, introspective reliability management (IRM), the approach championed in this work, promotes proactive reliability management. Rather than waiting for reliability problems to emerge and then recovering, IRM techniques actively try to dictate the rate at which these problems appear within a system. In the case of age-induced failures which are the focus of this work, the IRM solution presented in this paper, Olay, proactively throttles the degradation process with wearout-aware job scheduling.

Left unchecked, manufacturing defects evolve over time leading to the wearout-induced failure of individual cores in a CMP system, and eventually the entire CMP. However, since process variation causes some microarchitectural modules within a core to be weaker than others [42, 41], there is an advantage to assigning jobs to cores such that cores are not executing workloads that apply excessive reliability stress to their weakest modules. Given the diversity in workload behavior<sup>1</sup> an optimal versus suboptimal job assignment can have dramatically different reliability implications. Furthermore, since it has been shown that the rate at which damage accumulates is often a function of the amount of damage already present [37] the optimal schedule, for a given set of jobs, when the CMP is first manufactured may not resemble at all the optimal schedule for the same workload after the CMP has aged.

Despite not being able to reduce the presence of manufacturing defects or the computational demands placed upon a CMP, Olay can recognize (through profiling and low-level sensors) and exploit the heterogeneity in cores and workloads to maximize lifetime reliability enhancement through introspective, wearout-aware job scheduling. Although some previous DRM techniques are also proactive, approaches like [24] that apply DVFS to meet thermal targets, and indirectly reliability requirements, also alter CMP performance. Olay, on the other hand, merely dictates where a job executes on the underlying CMP, with no impact on performance. In fact, techniques like [24] and Olay are orthogonal solutions.

The main contributions of this paper include:

- A simulation framework that enables Failure Aware CMP Emulation (FACE) for lifetime reliability studies

<sup>1</sup>Characteristics like resource utilization, performance requirements, and temperature/power profiles vary substantially between applications.

- An IRM system, Olay, that leverages low-level sensors to improve lifetime reliability
- An evaluation of different reliability-aware job scheduling algorithms
- A discussion of the design trade-offs between sensor accuracy, algorithm complexity, and reliability enhancement

## 2. BACKGROUND

A large body of work exists in the literature on characterizing the behavior of common wearout mechanisms such as NBTI and TDDB, two of the more relevant mechanisms for future technologies and the focus of this paper. Research into the physical effects of wearout has shown that many of these mechanisms are progressive in nature [4, 23, 45, 11]. Unlike soft-errors that can occur suddenly and without warning, wearout-related faults are typically more gradual manifesting as small defects that eventually evolve into hard faults. This property of wearout suggests that before age-induced degradation can cause permanent failures in a CMP, monitoring the accumulation of damage can actually be used to dynamically project the life-expectancy of individual cores. The remainder of this section discusses the reliability models used in this work and also surveys some recent research into low-level reliability sensors. The mean time to failure (MTTF) models presented in this section are used by the FACE infrastructure to generate representative CMPs for Monte Carlo simulations as well as to characterize the reliability impact of CMP workloads (see Section 4).

### 2.1 NBTI

Negative bias temperature instability is a phenomenon that leads to an increase in the threshold voltage of PFET devices. When the gate is negatively biased with respect to the source and drain (i.e., when the PFET is “on”), Si-H bonds are broken and the diffusion of  $H^+$  leaves behind interfacial traps. The accumulation of this charge leads to threshold voltage shifts. Initially the performance of the transistor is impaired since the increased threshold leads to a reduced overdrive voltage, but correct (albeit slower) functionality is maintained. Eventually when NBTI results in a sufficiently large increase in threshold voltage, the PFET ceases to switch and experiences a stuck-at fault [32].

Recent work has also called attention to the property of NBTI that allows devices to heal when the applied stress is removed (i.e., when the gate is “high”) and the potential for designing self-healing circuits [1, 20]. Although some amount of damage can be undone, over time the net effect of stress and recovery cycles ultimately still leads to device failure. The model for NBTI used in this paper is based on work by Li et al. [22]. Equation 1 describes the mean time to failure with respect to NBTI,  $MTTF_{NBTI}$ , expected for a device given a set of expected operating conditions. Note the heavy dependence on temperature and voltage.

$$MTTF_{NBTI} \propto \left(\frac{1}{V}\right)^\gamma e^{\frac{E_{a,NBTI}}{\kappa T}} \quad (1)$$

where,

- $V$  = voltage
- $T$  = temperature
- $\gamma$  = voltage acceleration factor (6~8)
- $E_{a,NBTI}$  = activation energy (0.9~1.2eV)
- $\kappa$  = Boltzmann’s constant

### 2.2 TDDB

Time dependent dielectric breakdown, sometimes referred to as gate oxide breakdown, is caused by the formation of a conductive path through the gate oxide of a CMOS transistor. The exact physics of this path formation has actually been widely debated in the literature. Many theories have been proposed ranging from a field-driven, thermochemical model (E-model) [26, 27] to the Anode Hole Injection (AHI) model [12, 33]. Earlier work suggested that TDDB exhibits two distinct failure modes, namely soft and hard breakdown [15, 6, 35]. However, recently it has become increasingly more common to refer to the TDDB process as progressive rather than “hard” or “soft” [38, 29, 23].

For the purposes of this paper, TDDB can be viewed as a process that begins when a small conductive path is formed in the gate oxide and begins to evolve (grows in size and magnitude of leakage conducted) until the path supports enough current that the actual device is rendered unresponsive. The empirical model for TDDB employed in this paper is derived from a high-level equation (Equation 2) presented by Srinivasan et al. [36], which is based on experimental data collected at IBM [44]. Although  $MTTF_{TDDB}$  is affected by many factors, note that as with  $MTTF_{NBTI}$ , it has a strong dependence on operating voltage and temperature.

$$MTTF_{TDDB} \propto \left(\frac{1}{V}\right)^{(a-bT)} e^{\frac{(X+\frac{Y}{T}+ZT)}{\kappa T}} \quad (2)$$

where,

- $V$  = voltage
- $T$  = temperature
- $a, b, X, Y,$  and  $Z$  are all fitting parameters based on [44]
- $\kappa$  = Boltzmann’s constant

### 2.3 Wearout Sensors

Wearout monitoring and detection for on-chip devices is a challenging problem and has been an active area of research. Circuit-level designs have been proposed for in-situ sensors that detect the progress of various wearout mechanisms with reasonable accuracy [21, 28]. These sensors have been designed with area efficiency as a primary design criteria, allowing a large number of them to be deployed throughout the chip for monitoring overall system health. Most circuit-level sensors target the measurement of a physical quantity that is correlated with the extent of damage present in a device or set of devices. For example, work presented in [20] suggests that standby circuit leakage, IDDQ, can be used to detect and characterize temporal NBTI degradation in digital circuits ranging from ALUs to SRAM arrays. Given the large body of work on the design of accurate and area efficient IDDQ sensors [39], an IDDQ-based NBTI sensor makes an ideal candidate for system monitoring.

A different approach to sensor design has been to examine the health of on-chip resources at a coarser granularity. Research has involved simple temperature sensors, two dozen on the POWER6 [18], to more complex designs such as the wearout detection unit (WDU) presented in [9]. Although temperature provides higher-level feedback, the WDU tracks the degradation of timing and with the appropriate extensions, can approximate the useful life remaining in a microarchitectural module.

## 3. OLAY

Having addressed the relevant reliability background, this section discusses how the existence of wearout sensors can be used to enhance existing, and even enable new, DRM techniques. Combine sensors, intelligent algorithms, and DRM and the result is

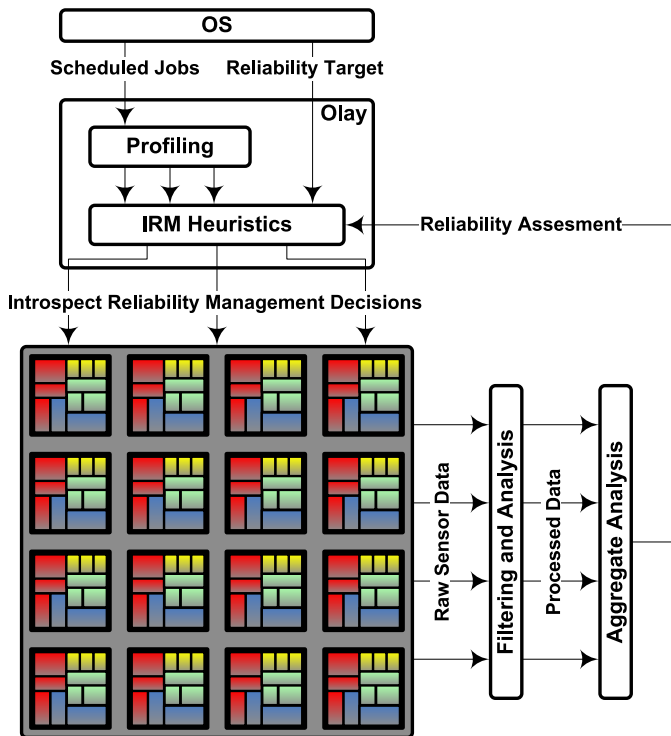


Figure 1: High-level block diagram of the Olay introspective reliability management system. Olay serves as a virtual layer between the OS and the underlying hardware. By analyzing sensor feedback Olay can enhance reliability by identifying wearout-aware job mappings.

IRM, an approach concerned with optimally meeting reliability targets placed on hardware systems through the intelligent application of DRM. The introspective nature of IRM arises from the online analysis of low-level sensor feedback, which enables IRM to dynamically tailor DRM decisions to the specific needs of individual processors in the field. The following provides the motivation for, and the intuition behind, IRM. The remainder of this section also presents Olay, the framework proposed in this paper to investigate one example of IRM, the intelligent, reliability-aware scheduling of jobs on a many-core CMP.

### 3.1 Introspective Reliability Management

Introspective reliability management, as used in this work, builds upon the idea that DRM is an effective tool to combat reliability challenges. Furthermore, the intuition behind IRM is that a DRM system augmented with low-level sensor feedback from the underlying hardware is even more effective. DRM schemes that rely solely upon statistical models of failure mechanisms, although effective when considering the reliability gains provided across an entire population of processors, can result in suboptimal solutions for any one part within that population. Without any form of feedback these DRM approaches must optimize for the common case and cannot adapt to the exceptional, or just *less* common, case. Furthermore, other DRM techniques that rely upon high-level information (e.g., instruction count and performance counters) or manufacturing-time characterization [40] neglect the effects of process variation in the former case and fails to account for wearout in the latter.

In contrast, IRM proposes continuous monitoring of the underlying CMP, and is able to overcome many of the shortcomings of con-

ventional DRM solutions. Figure 1 illustrates the high-level IRM vision. The system begins by collecting raw data streams from an array of sensors (Section 2.3). Statistical filtering and trend analysis converts these streams into descriptions of different system characteristics (e.g., delay profiles, leakage currents, etc.). These individual channels of information are then processed to generate a comprehensive high-level reliability appraisal of different parts of the CMP. Leveraging this real-time health assessment, IRM can meet reliability challenges like thermal hotspots and timing variation by performing a wide variety of traditional DRM tasks ranging from thread migration to dynamic voltage and frequency scaling. In addition, microarchitecture-specific information can also facilitate optimal job to core bindings. This particular application of IRM, intelligent job assignment, is the focus of Olay and the remainder of this work. As described below, insight from low-level circuit sensors enables Olay to apply wearout-aware job scheduling to retard the degradation of silicon<sup>2</sup>. Although ideal sensors are assumed in this paper, results in Section 5 suggests more realistic sensors with as much as +/-20% noise can still sustain respectable reliability gains.

### 3.2 Wearout-aware Job Scheduling

Olay relies on two fundamental principles, 1) CMP workloads are diverse and 2) process variation in future CMPs will result in not only heterogeneous performance capabilities but also varied reliability characteristics. With low-level sensor data that identifies the extent of damage present in individual cores, Olay can classify cores based on their reliability profile—e.g., Core  $i$  has a strong ALU but a weak FPU and Core  $j$  has a strong FPU but a weak LDSTQ. Coupling this knowledge with profiling that detects the inherent variability in workloads [34], Olay is able to schedule jobs on cores where they will leave behind the smallest reliability footprint (i.e., cause the least amount of wear).

This section briefly describes the scheduling policies used to illustrate the potential for reliability enhancement through wearout-aware job binding. Here, job scheduling refers to the act of mapping threads to cores and only occurs when the OS supplies Olay with a set of new jobs to execute. Existing jobs that have already been assigned do not migrate between cores. Techniques that perform thread migration for thermal management purposes are orthogonal and can also be incorporated into the Olay system for additional benefits. Note also that there are potentially other, more intricate, policies not presented here that could perform better. However, those that follow were chosen because despite their simplicity they yielded convincing results. Identifying and assessing more sophisticated algorithms is left for future work.

**Round-Robin:** This is the baseline policy used to evaluate the claims made by Olay. It binds jobs to cores in a conventional round-robin fashion, which, because it is oblivious to the condition of each core, results in an essentially random mapping.

**GreedyE:** This greedy policy attempts to maintain the maximum number of live cores *early* in life. This approach binds heavyweight jobs identified through run-time profiling to the strongest cores, those with less variation/wearout related damage. Less taxing workloads are reserved for the weaker cores. A distinction is also made between workloads with different resource requirements to avoid, for example, assigning a floating-point heavy application to a core with a weak FPU. In general, an effort is made to equalize the damage locally

<sup>2</sup>Olay can be thought of as an anti-aging treatment for silicon

within a core as well as globally across the CMP. In practice, cores that may have survived longer actually sacrifice some of their lifetime in order to lighten the burden on their weaker counterparts.

**GreedyL:** This version of the greedy policy aims to maximize the number of cores alive in *later* years, toward the end of life. Under this scheme, the heaviest jobs are actually assigned to the weaker cores. In essence, this policy culls the herd, victimizing the weak so that the strong can remain alive longer. It’s important to note that an attempt is still made to equalize damage locally by assigning jobs based on their resource requirements. By enabling the strongest cores, those with the least amount of initial damage and projected to have the longest lifetimes, to survive longer, the CMP remains functional (at least partially) for a greater period of time. Although counterintuitive at first, the GreedyL policy actually allows a CMP to execute more useful work in systems that are under-utilized (see Section 4.2), where having more cores around early in life only translates to more idle cores.

**GreedyA:** The final policy evaluated is a hybrid of GreedyE and GreedyL and *adapts* to the needs of the system. Early in the life of a CMP, when the cores are likely to be underutilized, GreedyA emulates a GreedyL scheduling policy. This prevents weak cores from surviving incrementally longer, but dying before they did anything other than sit idle, at the expense of stronger cores that could have been performing useful work further out in the future. As cores begin to fail and the number of cores falls below what’s needed to accommodate a nominal load (estimated from past history), GreedyA begins emulating a GreedyE policy. The intuition is that although it may be unnecessary to maximize live cores when a system is underutilized, as cores fail and system utilization approaches 100% (with respect to the number of functional cores) then a GreedyE approach is better at prolonging the life of the remaining cores<sup>3</sup>

## 4. SIMULATION INFRASTRUCTURE

In order to evaluate the merits of Olay, a framework had to be developed capable of conducting detailed lifetime reliability simulations. FACE (Failure Aware CMP Emulation) must perform three main tasks, 1) maintain a detailed reliability model of the CMP, 2) model workload execution and 3) simulate the impact, over an entire CMP lifecycle, reliability management policies have on the evolution of wearout (i.e., accurately simulate years of use in a fraction of the time). Each of the components that make up the FACE infrastructure (Figure 2) are addressed in the following text.

### 4.1 CMP Modeling

FACE is fundamentally different than the many simulation packages typically found in the architecture community. Unlike SimpleScalar [5] and SIMICS [25] which are concerned with cycle-accuracy and the impact of microarchitectural changes on program execution, a reliability simulator is more interested in the impact program execution has on the underlying microarchitecture. Figure 2b depicts the hierarchical design of the CMP model used by FACE. Each level of the hierarchy will be addressed in turn.

**CMP:** The CMP is modeled as a collection of  $n$  cores, each based on the DEC Alpha 21264 [2]. The cores are tiled in

<sup>3</sup>This is based on the principle of multiplicative degradation (See Section 4.1).

a grid pattern with the L1 caches serving as boundaries between adjacent cores. Similar floorplans are used in [43, 14], allowing for the simplifying assumption that the caches act as thermal barriers preventing modules from directly influencing the temperatures of neighboring cores (except through the heatsink). A CMP is considered alive and capable of executing jobs as long as one of its  $n$  cores remains alive.

**Core:** Individual cores within the CMP are modeled as a collection of microarchitectural modules. As the core executes an application these microarchitectural modules experience different operating temperatures and dissipate varying amounts of power depending on their activity. The thermal interaction between neighboring modules within the same core is modeled with HotSpot [34]. A core is considered dead and unable to receive job assignments when any of its modules dies.

**Module:** Microarchitectural modules, as with the CMP and cores are modeled as a collection of smaller components, a set of  $N$  transistors. Transistors are distributed between the modules proportionally based on their area. All devices within the same architectural module are assumed to experience the same operating conditions (i.e., temperature) at any given point in time. A module is considered functional as long as none of its constituent transistors is dead.

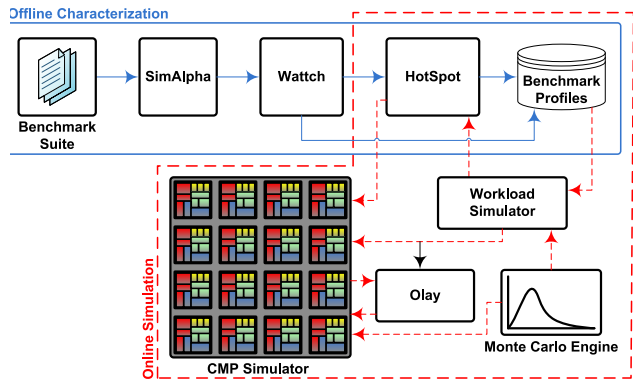
**Transistor:** As discussed in Section 2 many wearout mechanisms, particularly NBTI and TDDB, manifest as small defects which slowly develop into full-fledged hard faults. In light of this, aging-related degradation is modeled at the transistor level as the accumulation of damage. The evolution of different wearout mechanisms within a given transistor is assumed to be independent and the first mechanism to exceed its damage threshold causes the transistor to die.

**Mechanism:** The modeling of mechanism-specific damage accumulation obeys the multiplicative degradation principle [30] and is consistent with the models used by others [37]. In brief, this principle states that the amount of damage incurred in any time interval is a function of the amount of damage that existed in the previous time interval (Equation 3).

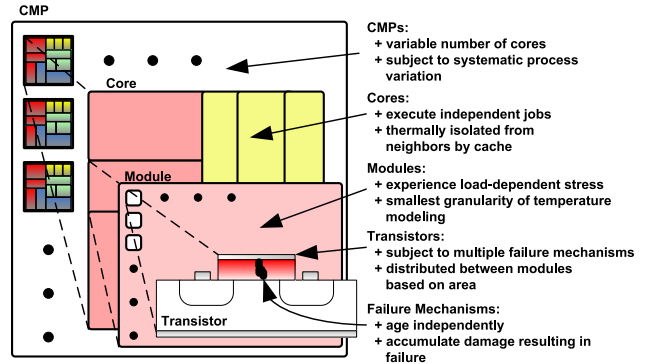
$$\begin{aligned} D_n &= (1 + \alpha_n)D_{n-1} \\ &= [\prod_{i=0}^n (1 + \alpha_i)]D_0 \end{aligned} \quad (3)$$

where  $\alpha_i$  is the degradation rate at time  $i$  and  $D_0$  is the amount of damage that was present when the device was manufactured.

The rate at which damage accumulates at time  $i$ , or  $\alpha_i$ , is determined by the instantaneous MTTF for that particular mechanism at time  $i$ ,  $MTTF_i$ . The instantaneous MTTF is what the MTTF for a given mechanism would be had the CMP been reliability qualified assuming the set of operating conditions present at time  $i$ . The ratio between  $MTTF_i$  and the reliability qualified MTTF,  $MTTF_{qual}$  describes the relative stress the device is exposed to at time  $i$  and consequently the instantaneous rate of damage accumulation,  $\alpha_i$ . The mechanism-specific expected lifetime, or time to failure (TTF), of each transistor is used to determine the amount of damage that exists at manufacture time ( $D_0$ ). The TTFs of each transistor in the CMP are generated from a Weibull distribution by the Monte Carlo engine separately for each simulation run. The mean of the distribution is the the mechanism-specific MTTF (i.e.,  $MTTF_{NBTI}$  and  $MTTF_{TDDB}$ ) of



(a) Block diagram of the FACE framework. Lifetime reliability simulations consist of two stages, 1) offline characterization and 2) online simulation.



(b) Hierarchical design of the CMP simulator.

Figure 2: FACE framework and the CMP simulation hierarchy

the module in which the transistor resides, calculated using the operating temperature for which the module was qualified. Note that since all transistors within a module are assumed to experience the same operating conditions the mechanism-specific MTTFs are common to all devices within a module while TTFs are unique to every transistor.

Modeling transistor damage in this manner ensures that under worst-case operating conditions device  $n$  will have developed a hard fault at  $TTF_{n,Min}$ , where  $TTF_{n,Min}$  is the minimum time to failure across all failure mechanisms for device  $n$ . This is consistent with industry practices, where a given part is designed (not including margins) to operate at a qualification temperature for a fixed number of years. If the actual operating conditions differ from those at which the part was qualified then the part's lifetime will also change accordingly.

Given this hardware model, at every time step FACE updates the damage information for each transistor in the system based on the temperature and power profiles of the jobs that are currently assigned to cores. This information is then propagated all the way up the hierarchy. The process repeats itself at every time step until the last core in the CMP system succumbs to failure.

## 4.2 Workload Modeling

Since FACE is concerned with the reliability stress placed on the underlying CMP when jobs are executed, workloads are abstracted from the typical stream of instructions to a higher-level, time-dependent, trace of power and temperature profiles. Figure 2a illustrates how this characterization is performed. A benchmark suite<sup>4</sup> is simulated with a tool-chain consisting of SimAlpha, Wattch, and HotSpot. Initially SimAlpha and Wattch are used to generate a database of per-module power traces for each benchmark. This information is then used at run-time by HotSpot to calculate module temperatures across the CMP.

In addition to characterizing individual benchmarks, FACE also simulates time dependent variation in CMP utilization from a systems perspective. This flexibility allows it to model different application domains, from embedded systems to high performance server farms. Previous research into data center design has shown

<sup>4</sup>For the preliminary results presented in this work synthetic benchmarks are used. However, the simulation framework can accommodate any arbitrary benchmark suite (i.e., SPEC2000).

that servers experience highly variable utilization. Since designers build data centers to accommodate peak loads, it's not surprising that they are often over-provisioned for the common case. Some reports claim average utilization is as low as 20% of peak [3]. On the other hand, the utilization of embedded processors in mobile devices is characterized by short periods of maximum utilization followed by longer idle periods of virtually no utilization.

To support these different scenarios FACE, uses a statistical model that emulates the OS scheduler. It generates a randomly selected set of live threads (benchmarks) that are active every time slice, where the number of threads varies over time depending upon the mean and standard deviation of the expected system utilization. This combination of the benchmark characterization and utilization modeling provides for a manageable yet accurate workload model for lifetime reliability simulations.

## 4.3 Lifetime Simulation

A Monte Carlo based approach is used to drive lifetime simulations under the FACE framework. Given that CMPs have lifespans on the order of years (3-5 years in future computer systems [17]) detailed lifetime reliability simulations on a many-core CMP is a computationally intensive task, especially when large numbers of Monte Carlo runs have to be conducted to generate statistically significant results. A set of additional simplifying assumptions help to facilitate simulation in a manageable amount of time. The following discusses two of the main assumptions and how each helps to reduce simulation effort without compromising on accuracy.

**Adaptive Fast-Forwarding:** With the understanding that wearout damage takes years to reach critical mass, FACE implements an *adaptive* fast-forwarding (AFF) scheme. Short periods of detailed simulation (DS) are used to record the progression of CMP aging. During the DS phase, degradation information is calculated and recorded at all levels of the CMP hierarchy (Section 4.1). This information is then used to rapidly advance the simulation process by essentially reproducing the effects of the DS period. To minimize the error incurred as a result of AFF, the amount of fast-forwarding, the fast-forwarding factor (FFF), is determined by the amount of damage accumulated during the DS interval. This dynamically adjusted FFF ensures that fast-forwarding slows down toward the end of life, where small

changes in damage can have large implications<sup>5</sup>. This cycle of DS followed by AFF is repeated for the duration of each simulation run. The DS phase can be viewed as a representative snapshot of how the CMP responds to different workloads and IRM policies, which for the purposes of this paper are limited to the job assignment decisions provided by Olay, during the longer AFF phase.

**Two-tiered Temperature Modeling:** Since running detailed HotSpot simulations online to determine module temperatures is prohibitively expensive, FACE uses a two tiered approach to temperature modeling. First, all benchmarks are characterized offline using detailed HotSpot simulations. This information is used to populate a database with  $\Delta T_{ij}$  values for each microarchitectural module, where  $\Delta T_{ij}$  is defined as the difference between the module temperature and the temperature of the CMP heatsink at time  $i$  for benchmark  $j$ . The second part of the two-tiered approach involves heatsink simulations with HotSpot at run-time. Modeling just the heatsink at run-time allows HotSpot to perform a much simpler and far more tractable simulation. Although this does introduce inaccuracy into the modeling, empirical studies suggested that the error is negligible.

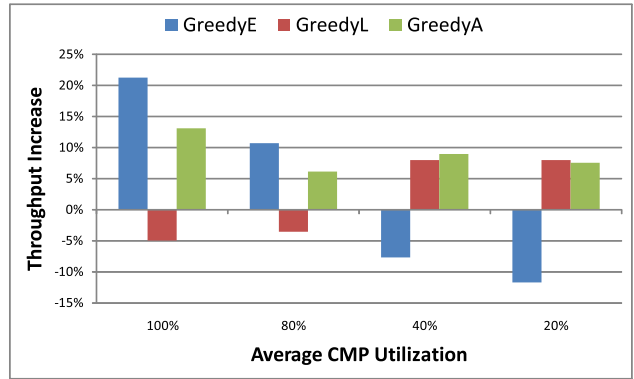
## 5. EVALUATION AND ANALYSIS

This section analyzes the performance of Olay and discusses the source of some of its reliability improvements. As described in Section 4.1, Monte Carlo experiments were conducted using a variable size CMP running synthetically generated benchmarks. The effectiveness of each wearout-aware scheduling policy (see Section 3.2) is measured in terms of the cumulative throughput — the number of cycles spent executing active threads, summed across all cores. Throughput, as used in this work, is effectively the amount of useful work performed by the CMP prior to all the cores failing. Results show more than 20% average throughput improvements by applying introspective job scheduling over the baseline round-robin scheduler (Figure 3). This section will discuss the impact of job scheduling heuristics and system configuration on the results.

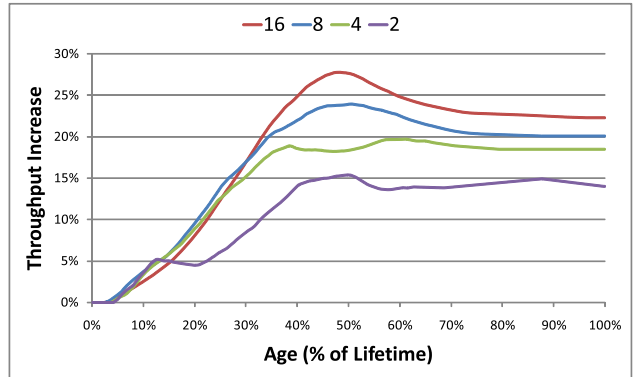
**CMP Utilization:** As mentioned previously (Section 4.2), the utilization of computer systems is highly variable, both within the same domain (e.g., variability inside data centers) and across domains. Figure 3a depicts the performance of the Olay scheduling policies given the average expected utilization of the system<sup>6</sup>. Note that the relative performance of the policies changes with different utilizations. This is expected given the heuristics applied by each policy (see Section 3.2). As the utilization rises, the need of uniformly aging the cores on the CMP also increases. On the other hand, with lower utilizations it is more beneficial to maximize the life of a subset of cores while allowing excess cores to die off. Note also that the ability of the greedy GreedyA policy to emulate an ideal policy for a given utilization level is a function of how well it can predict utilization patterns. The better the heuristic, the quicker GreedyA can respond to changes.

<sup>5</sup>A CMP later in life is likely to contain more damage and therefore, will experience more degradation for the same stress (i.e., workload) than it would have earlier in its life. Therefore, the FFF is much greater early in the simulation process than later.

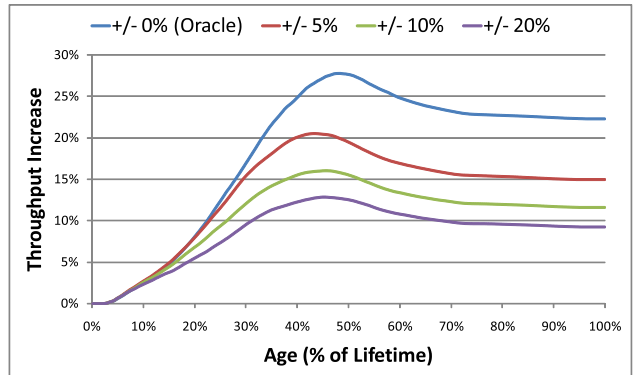
<sup>6</sup>Although mean utilization is fixed, actual CMP utilization over the course of a simulation run is constantly changing. Furthermore, the average *effective* utilization is also changing as cores on the CMP begin to fail.



(a) Impact of varying CMP utilization. Results are for a 16-core CMP with ideal sensors.



(b) Impact of varying CMP size (number of cores). Results are for full average utilization, ideal sensors, and a GreedyE policy.



(c) Impact of varying sensor noise. Results are for a 16-core CMP, full average utilization, and a GreedyE policy.

Figure 3: Impact of various CMP configurations on the performance of different wearout-aware scheduling policies. Throughput improvements are normalized to a naive round-robin scheduler.

**CMP Size:** As the number of cores on a CMP continues to grow [19], the benefits of applying Olay will also increase. A massively multicore CMP will possess even greater heterogeneity (due to process variation and wearout) and offers more opportunities for introspective techniques like Olay to exploit. More cores means more decisions that must be made, which will serve to magnify the growing gap between

naive and intelligent introspective policies. Figure 3b shows a difference of 10% in improved throughput (GreedyE normalized to a naive round-robin policy) when scaling from a 2-core to a 16-core CMP system. The throughput improvements appear to peak around 50% of the lifetime and slowly taper down toward the end of life. This is expected. Just as no throughput improvement is possible early on in life, when all cores in the CMP are still alive, toward the end of life when only a few cores remain alive (despite the best efforts of the introspective policies) there is only incremental room for improvement. Throughout the rest of the lifetime, however, introspective policies are able to achieve substantial improvements in throughput by intelligently distributing work between the cores.

**Error-prone Sensors:** Since ideal sensors were assumed for this work, it was necessary to evaluate Olay’s sensitivity to sensor error. Figure 3c illustrates how more realistic, error-prone sensors would impact performance. Although the introduction of systematic error, which is studied in Figure 3c, does reduce some of the reliability gains, the presence of random noise (more common for circuit-level sensors) is accounted for and mitigated by the statistical filtering and trend analysis component of Olay. Yet despite +/-20% of systematic error Olay still achieves a 9% improvement in throughput. Note that for the same reasons as Figure 3b, Figure 3c also shows a peak in throughput improvement around 50% of the CMP lifetime.

These preliminary results show that wearout-aware scheduling is indeed capable of achieving lifetime reliability gains, but more interesting is the fact that there exists additional opportunities that can be further explored with other policies. Two main areas of future work will involve 1) identifying better algorithms and 2) defining other, perhaps domain-specific, metrics by which to measure reliability enhancement.

First, although IRM techniques like Olay should intuitively produce better results than their naive counterparts, fully exploiting much of this improvement relies upon the quality of the heuristics and algorithms used. For example, the strong dependence on utilization suggests that enhancing Olay with better utilization predictors could dramatically improve the performance of the hybrid GreedyA policy. Moreover, better thread-level profiling would benefit all wearout-aware policies alike and allow the modeling of more realistic benchmarks, and their attendant complexities.

Secondly, perhaps more important is the development and refinement of appropriate metrics. The simple metric of cumulative work done used in these preliminary studies may not be appropriate for all domains. In a high-performance computing setting, mostly concerned with instantaneous throughput, having CMPs struggling along with only a small fraction of their cores functioning may be of little use. However, in domains with less frequent hardware turnover this type of extended, graceful degradation may be acceptable. Equally as likely may be the scenario where the user has a fixed reliability lifetime beyond which point systems will be preemptively replaced. In such an environment Olay’s policies could be modified to account for this *hard* reliability target, allowing them to avoid decisions that would extend core lifetimes beyond the target in favor of those that allow cores to die closer to the target lifetime, using the extra reliability “slack” to preserve weaker cores earlier on.

## 6. CONCLUSION

As large CMP systems continue to grow in popularity and technology scaling continues to exacerbate lifetime reliability challenges, the research community must develop innovative ways for systems to dynamically adapt. Although issues like process variation are the source of design and validation nightmares, this inherent heterogeneity in future systems is embraced by the IRM philosophy. Techniques like Olay recognize that although emerging reliability obstacles cannot be ignored, with the appropriate monitoring, they can be overcome. Despite focusing on just the benefits of wearout-aware job scheduling, Olay is still able to achieve more than 20% lifetime reliability enhancement. More comprehensive IRM approaches that leverage sensor feedback to improve upon other traditional DRM mechanisms (e.g., DVFS) should demonstrate still more potential, perhaps enough to encourage researchers pursuing work in low-level sensors to press on with even greater zeal.

## 7. REFERENCES

- [1] J. Abella, X. Vera, and A. Gonzalez. Penelope: The nbt-aware processor. In *Proc. of the 40th Annual International Symposium on Microarchitecture*, pages 85–96, Dec. 2007.
- [2] Alpha. 21364 family, 2001. <http://www.alphaprocessors.com/21364.htm>.
- [3] A. Andrzejak, M. Arlitt, and J. Rolia. Bounding the resource savings of utility computing models, Dec. 2002. HP Laboratories, <http://www.hpl.hp.com/techreports/2002/HPL-2002-339.html>.
- [4] J. S. S. T. Association. Failure mechanisms and models for semiconductor devices. Technical Report JEP122C, JEDEC Solid State Technology Association, Mar. 2006.
- [5] T. Austin, E. Larson, and D. Ernst. SimpleScalar: An infrastructure for computer system modeling. *IEEE Transactions on Computers*, 35(2):59–67, Feb. 2002.
- [6] A. Avellan and W. H. Krautschneider. Impact of soft and hard breakdown on analog and digital circuits. *IEEE Transactions on Device and Materials Reliability*, 4(4):676–680, Dec. 2004.
- [7] K. Bernstein. Nano-meter scale cmos devices (tutorial presentation), 2004.
- [8] D. Blaauw, S. Kalaiselvan, K. Lai, W.-H. Ma, S. Pant, C. Tokunaga, S. Das, and D. Bull. Razor II: In-situ error detection and correction for PVT and SER tolerance. In *IEEE International Solid-State Circuits Conference*, Feb. 2008.
- [9] J. Blome, S. Feng, S. Gupta, and S. Mahlke. Self-calibrating online wearout detection. In *Proc. of the 40th Annual International Symposium on Microarchitecture*, pages 109–120, 2007.
- [10] S. Borkar. Designing reliable systems from unreliable components: The challenges of transistor variability and degradation. *IEEE Micro*, 25(6):10–16, 2005.
- [11] J. Carter, S. Ozev, and D. Sorin. Circuit-level modeling for concurrent testing of operational defects due to gate oxide breakdown. In *Proc. of the 2005 Design, Automation and Test in Europe*, pages 300–305, June 2005.
- [12] I. C. Chen, S. E. Holland, K. K. Young, C. Chang, and C. Hu. Substrate hole current and oxide breakdown. *Applied Physics Letters*, 49(11):669–671, 1986.
- [13] A. K. Coskun et al. Analysis and optimization of mpsoc reliability. *Journal of Low Power Electronics*, 2(1):56–69, Apr. 2006.
- [14] J. Donald and M. Martonosi. Techniques for multicore

- thermal management: Classification and new exploration. In *Proc. of the 33rd Annual International Symposium on Computer Architecture*, June 2006.
- [15] D. Dumin. *Oxide Reliability: A Summary of Silicon Oxide Wearout, Breakdown, and Reliability*. World Scientific Publishing Co. Pte. Ltd., 2002.
- [16] D. Ernst, S. Das, S. Lee, D. Blaauw, T. Austin, T. Mudge, N. S. Kim, and K. Flautner. Razor: Circuit-level correction of timing errors for low-power operation. In *Proc. of the 37th Annual International Symposium on Microarchitecture*, pages 10–20, 2004.
- [17] C. Evangs-Pughe. Live fast, die young [nanometer-scale ic life expectancy]. *IEE Review*, 50(7):34–37, 2004.
- [18] J. Friedrich et al. Desing of the power6 microprocessor, Feb. 2007. In *Proc. of ISSCC*.
- [19] Intel. Tera-scale computing research program, 2008.
- [20] K. Kang, K. Kim, A. E. Islam, M. A. Alam, and K. Roy. Characterization and estimation of circuit reliability degradation under nbtI using on-line iddq measurement. In *Proc. of the 44th Design Automation Conference*, June 2007.
- [21] E. Karl, P. Singh, D. Blaauw, and D. Sylvester. Compact in situ sensors for monitoring nbtI and oxide degradation. In *2008 IEEE International Solid-State Circuits Conference*, Feb. 2008.
- [22] X. Li, B. Huang, J. Qin, X. Zhang, M. Talmor, Z. Gur, and J. B. Bernstein. Deep submicron cmos integrated circuit reliability simulation with spice. In *Proc. of the 2005 International Symposium on Quality of Electronic Design*, pages 382–389, Mar. 2005.
- [23] B. P. Linder and J. H. Stathis. Statistics of progressive breakdown in ultra-thin oxides. *Microelectronic Engineering*, 72(1-4):24–28, 2004.
- [24] Z. Lu, J. Lach, M. R. Stan, and K. Skadron. Improved thermal management with reliability banking. *IEEE Micro*, 25(6):40–49, Nov. 2005.
- [25] P. S. Magnusson et al. Simics: A full system simulation platform. *IEEE Computer*, 35(2):50–58, Feb. 2002.
- [26] J. McPherson and R. Khamankar. Molecular model for intrinsic time-dependent dielectric breakdown in  $\text{SiO}_2$  dielectrics and the reliability implications for hyper-thin gate oxide. *Semiconductor Science and Technology*, 15(5):462–470, May 2000.
- [27] J. McPherson and H. Mogul. Underlying physics of the thermochemical e model in describing low-field time-dependent dielectric breakdown in  $\text{SiO}_2$  thin films. *Journal of Applied Physics*, 84(3):1513–1523, Aug. 1998.
- [28] S. Mishra and M. P. adn Douglas L. Goodman. In-situ sensors for product reliability monitoring, 2006. <http://www.ridgetop-group.com/>.
- [29] F. Monsieur, E. Vincent, D. Roy, S. Bruyere, J. C. Vildeuil, G. Pananakakis, and G. Ghibaudo. A thorough investigation of progressive breakdown in ultra-thin oxides. physical understanding and application for industrial reliability assessment. In *Proc. of the 2002 International Reliability Physics Symposium*, pages 45–54, Apr. 2002.
- [30] NIST. Assessing product reliability, chapter 8, nist/sematech e-handbook of statistical methods. <http://www.itl.nist.gov/div898/handbook/>.
- [31] D. Roberts, R. Dreslinski, E. Karl, T. Mudge, D. Sylvester, and D. Blaauw. When homogeneous becomes heterogeneous: Wearout aware task scheduling for streaming applications. In *Proc. of the Workshop on Operating System Support for Heterogeneous Multicore Architectures*, Sept. 2007.
- [32] D. K. Schroder and J. A. Babcock. Negative bias temperature instability: Road to cross in deep submicron silicon semiconductor manufacturing. *Journal of Applied Physics*, 94(1):1–18, July 2003.
- [33] C. Schuegraf and c. Hu. Metal-oxide-semiconductor field-effect-transistor substrate current during fowler-nordheim tunneling stress and silicon dioxide reliability. *Journal of Applied Physics*, 76(6):3695–3700, Sept. 1994.
- [34] K. Skadron, M. R. Stan, K. Sankaranarayanan, W. Huang, S. Velusamy, and D. Tarjan. Temperature-aware microarchitecture: Modeling and implementation. *ACM Transactions on Architecture and Code Optimization*, 1(1):94–125, 2004.
- [35] P. Solomon. Breakdown in silicon oxide - a review. *Journal of Vacuum Science and Technology*, 14(5):1122–1130, Sept. 1977.
- [36] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. The case for lifetime reliability-aware microprocessors. In *Proc. of the 31st Annual International Symposium on Computer Architecture*, pages 276–287, June 2004.
- [37] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers. Exploiting structural duplication for lifetime reliability enhancement. In *Proc. of the 32nd Annual International Symposium on Computer Architecture*, pages 520–531, June 2005.
- [38] J. Sune and E. Wu. From oxide breakdown to device failure: an overview of post-breakdown phenomena in ultrathin gate oxides. In *International Conference on Integrated Circuit Design and Technology*, pages 1–6, May 2006.
- [39] J.-J. Tang, K.-J. Lee, and B.-D. Liu. A practical current sensing technique for iddq testing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 3(2):302–310, June 1995.
- [40] R. Teodorescu, J. Nakano, A. Tiwari, and J. Torrellas. Mitigating parameter variation with dynamic fine-grain body biasing. In *Proc. of the 40th Annual International Symposium on Microarchitecture*, Dec. 2007.
- [41] R. Teodorescu and J. Torrellas. Variation-aware application scheduling and power management for chip multiprocessors. In *Proc. of the 35th Annual International Symposium on Computer Architecture*, pages 363–374, June 2008.
- [42] A. Tiwari, S. Sarangi, and J. Torrellas. Recycle: Pipeline adaptation to tolerate process variation. In *Proc. of the 34th Annual International Symposium on Computer Architecture*, pages 323–334, June 2007.
- [43] J. Winter and D. Albonesi. Scheduling algorithms for unpredictably heterogeneous cmp architectures. In *Proc. of the 2008 International Conference on Dependable Systems and Networks*, page To appear, June 2008.
- [44] E. Wu et al. Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides. *Solid-State Electronics*, 46:1787–1798, 2002.
- [45] X. Yang, E. Weglarz, and K. Saluja. On nbtI degradation process in digital logic circuits. In *Proc. of the 2007 International Conference on VLSI Design*, pages 723–730, Jan. 2007.