Investigation of DVFS Based Dynamic Reliability Management for Chip Multiprocessors

Milad Ghorbani Moghaddam
Electrical and Computer Engr.
Marquette University
Milwaukee WI, USA
Email: milad.ghorbanimoghaddam@marquette.edu

Alexandre Yamamoto
Electrical and Computer Engr.
Texas A&M University
College Station TX, USA
Email: yama501@tamu.edu

Cristinel Ababei
Electrical and Computer Engr.
Marquette University
Milwaukee WI, USA
Email: cristinel.ababei@marquette.edu

Abstract—**We investigate dynamic voltage and frequency scaling (DVFS) as a mechanism for dynamic reliability management (DRM) of chip multiprocessors (CMPs). The proposed DRM scheme operates as a control technique whose objective is to drive the operation of the CMP such that reliability changes towards a desired target. While the chip multiprocessor is continuously monitored and reliability is estimated in real time, the voltage and frequency of different cores in the CMP are dynamically adjusted such that reliability converges towards the target. When the temperature of cores increases and thus reliability degrades, the proposed DRM scheme throttles selectively the frequency of the cores with the highest temperature. This is turn, leads to a lower power dissipation in those cores whose temperature decreases, thereby improving reliability. We leverage existing simulation and estimation tools to develop the proposed DRM scheme. Simulations results show that the proposed DRM scheme provides an effective way to tradeoff reliability and performance.**

Keywords*—*dynamic reliability management; dynamic voltage and frequency scaling; chip multiprocessors

I. Introduction

Wearout or aging failure mechanisms of CMOS integrated circuits include time-dependent dielectric breakdown (TDDB), negative bias temperature instability (NBTI), electromigration (EM), thermal cycling (TC), and stress migration (SM). The impact of these failure mechanisms has become increasingly adverse due to the increased power densities and system complexity. Faster aging leads to earlier performance degradation with eventual device breakdown and thus system failure due to errors. The shift from singlecore to multicore processors has somewhat alleviated the issue of increasingly large power densities. However, this issue persists especially with the advent of chip multiprocessors (CMPs) that integrate tens and hundreds of cores[1] on the same chip; some cores must be shut off to keep power densities under control, thereby not utilizing fully the available computational power of chip multiprocessors (CMPs). This is commonly referred to as dark silicon problem. Therefore, reliability started to attract more attention and it has become an important design challenge.

Generally, we can classify reliability oriented design methods into two categories. The first category is that of *static approaches*, which address the problem of reliability at design time. Static design methods include guardbanding and fault tolerance techniques. For example, supply voltages are selected high enough to guarantee correct functionality despite variation in threshold voltage or in temperature and supply noise. In this way energy gained from downscaling is sacrificed to combat reliability problems. However, if this sacrifice becomes too large, downscaling may become detrimental [1]. Fault tolerance techniques are based on fault detection and recovery mechanisms, which require energy and area overheads. Previous work employed fault tolerant techniques based on 1) error detection implemented through coarse grained replication or redundancy [2]–[4], 2) failure prediction used to take preventative measures to avoid, or at least mitigate the effects of device failures [5]–[7], and 3) error masking [8]. Simulated annealing is used to optimize both energy and reliability in [9]. A sequential quadratic programming based approach is proposed in [10] to maximize the lifetime of a multiprocessor system considering the electromigration effects in communication links. A wearout aware schedulability analysis technique is introduced in [11] for real-time independent tasks mapped to processor with dynamic voltage and frequency scaling capabilities. A convex optimization based approach is proposed in [12] to maximize the lifetime reliability of the cores of a multiprocessor system subject to electromigration wearout. The study in [13] uses genetic algorithms to identify voltages and frequencies of the cores of a multiprocessor system to maximize the lifetime and minimize the soft-error susceptibility. The main challenge of this category of methods is to reduce the energy and area overheads while reliability is still improved.

The second category of reliability oriented design methods is that of *dynamic approaches*. The main idea of this class of approaches is to dynamically monitor the system during runtime and by using either *reactive* or *proactive* techniques to change the operation of the system such that reliability is improved. Note that these approaches may use support from the first category of static approaches discussed in the previous paragraph. A two phase DRM algorithm to address various aging mechanisms is introduced in [14]. In the first phase, an application is profiled to find the maximum performance that each hardware configuration can run while still maintaining the desired mean time to failure (MTTF). In the second phase, the configuration with the highest performance and satisfying MTTF is selected for the remaining application's run. Dynamic reliability banking is proposed in [15] to address aging due to electromigration. Reliability slack is introduced in [16] and used for dynamic reliability management during periods of high processing demand. The authors of [17] exploit the natural

[1] It is predicted that actually future CMPs will integrate thousands of cores.

variation in workloads to assign jobs to cores in a manner that minimizes the impact of NBTI and TDDB on lifetime reliability. The authors of [18] introduce Facelift, a technique to hide aging through aging-driven application scheduling and to slow it down by applying voltage changes at key times. A DVFS control and look-up table reliability estimation based DRM scheme is introduced in [19] for singlecore processors to address process variation aware oxide breakdown. The impact of job scheduling based power management on reliability is investigated in [20]. A dynamic tile partition algorithm is introduced in [21] to balance workload among active cores while relaxing stressed ones. A system level HW/SW reliability management scheme where a chip dynamically adjusts its own operating frequency and supply voltage over time as the devices age due to NBTI is introduced in [22]. The authors of [23] study a control theoretic approach that uses data from aging sensors to compute the wearout degradation and to maximize the lifetime of homogeneous multicore systems. The same authors introduce a complete software implementation, working on a real mobile hardware platform, of a workload-aware dynamic reliability management technique to address TDDB wearout [24]. A reinforcement learning algorithm is proposed in [25] to optimize the lifetime of a multicore system by controlling the average temperature and thermal cycling. While the majority of previous work focus with their reliability oriented design methods only either on the computational portion of the system (i.e., singlecore or multicores) or on the communication component (i.e., buses or networks-on-chip), the authors of [26] concentrate on the combination of both. They use a neural network based reliability estimator and thread migration for dynamic reliability management of chip multiprocessors.

In this paper, we investigate the use of dynamic voltage and frequency scaling as a mechanism for dynamic reliability management of chip multiprocessors. Because for CMPs either the cores or the network-on-chip may become the reliability bottleneck, we focus on the whole CMP system as the combination of both components. We leverage existing full system simulation and reliability estimation tools to develop the proposed DRM scheme. We use these tools to conduct simulations for different CMP architectures and report results for several PARSEC benchmarks. Simulation results show that the proposed DVFS based DRM scheme provides a mechanism to trade reliability with performance. To the best of our knowledge, no prior work investigated DVFS based dynamic reliability management for CMPs treated as the combination of both computational and communication units.

II. Background

In this section we discuss briefly background information on time-dependent dielectric breakdown (TDDB) and negative bias temperature instability (NBTI) as the two failure mechanisms that we deal with in this paper. We also discuss the concept of dynamic voltage and frequency scaling (DVFS).

A. Wearout Failure Models

1) Time Dependent Dielectric Breakdown: Time dependent dielectric breakdown (TDDB) is a wearout mechanism that is caused by the formation of a conducting path through the gate oxide to substrate inside CMOS transistors due to electron tunneling current [27]. TDDB has become increasingly severe as the thickness of the gate oxide decreased due to continuous technology downscaling. It has been studied a lot in the past and few modeling approaches exist. Here, we use the *TDDB lifetime model* described by the following expression [14]:

$$MTTF_{TDDB} \propto (\frac{1}{V})^{a-bT} \times e^{\frac{X+\frac{Y}{T}+ZT}{kT}} \tag{1}$$

where k is the Boltzmann's constant and a, b, X, Y, and Z are model fitting parameters and are determined from experimental data. In our implementation discussed later on, we use the same values as in [14] $a = 78$, $b = -0.081$, $X = 0.759eV$, $Y = -66.8eVK$, and $Z = -8.37e^{-4}eV/K$ based on the data from [28].

2) Negative Bias Temperature Instability: Negative bias temperature instability (NBTI) is another wearout mechanism that mainly affects PFET CMOS transistors, when they are stressed at large negative gate voltages and high temperatures [29]. NBTI manifests as a gradual increase in the threshold voltage and consequent decrease in drain current and transconductance. The degradation exhibits logarithmic dependence on time. This aging phenomenon has also become more adverse due to technology downscaling. Likewise TDDB, NBTI has been investigated a lot. One of the most popular *NBTI lifetime models* gives for $MTTF_{NBTI}$ at a temperature T, and is described by the following expression [14]:

$$MTTF_{NBTI} \propto [(ln(\frac{A}{1+2e^{\frac{B}{kT}}})-ln(\frac{A}{1+2e^{\frac{B}{kT}}}-C))\times\frac{T}{e^{\frac{D}{kT}}}]^{\frac{1}{\beta}} \tag{2}$$

where A, B, C, D, and β are model fitting parameters. We use the same values as in [14] $A = 1.6328$, $B = 0.07377$, $C = 0.01$, $D = -0.06852$, and $\beta = 0.3$ based on the data from [30].

B. Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) is a design operation mode or design style where partitions of the same system are supplied with different supply voltages and clock frequencies that can be changed dynamically. The objective of this design style is to reduce average power dissipation without degrading performance. These partitions are referred to as voltage-frequency islands (VFI) and the communication between such islands must be done via first input first output (FIFO) queues to buffer the data transmitted between different islands operating at different clock rates. Additional costs incur due to the required voltage level converters.

There has been significant work done on DVFS based design optimization. On one hand, previous studies focus on computational cores. Specifically, chip multiprocessors (CMPs) can especially benefit from DVFS that enables power management while conducting computations under stringent power considerations [31]. Consequently, we find prior works addressing several design aspects including voltage island formation [32], power management techniques [33], or thermal management [34].

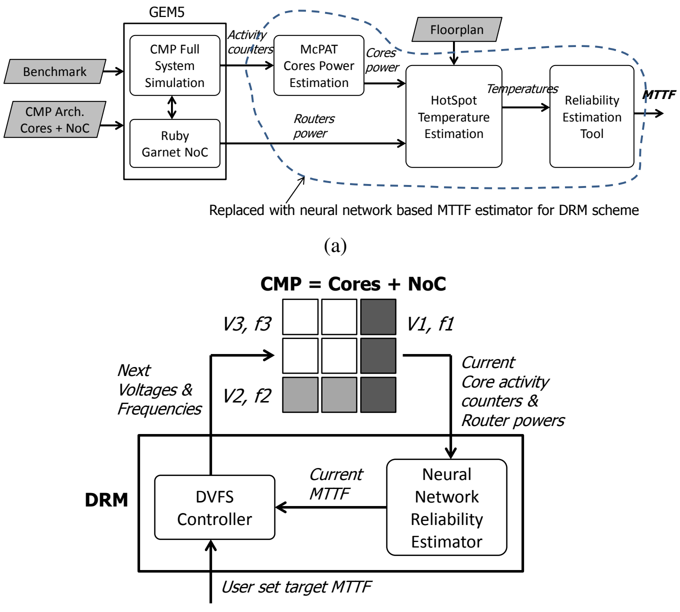
On another hand, previous studied focus on networks-on-chip as the communication component in today's chip multiprocessors or complex multicore systems-on-chip. In this context, previous DVFS techniques are typically applied at either router/link level or cluster of routers level. For example, in the first category, the authors of [35] use DVFS for NoC links. Frequency boosting is used to further improve the link performance [36] while in [37] DVFS is applied to the wire-lines of wireless NoCs. In the second category, many previous studies present methods to partition the NoC into several VFIs and methodologies for runtime energy management [38]. Because the granularity of VFIs is coarser in this case, the potential energy savings are generally less than when VF islanding is done at the router level.

In this paper, we use dynamic voltage and frequency scaling (DVFS) as the primary knob to control the application or benchmark behavior such that the per-tile power dissipation is geared in such a way that the overall CMP lifetime reliability moves towards the desired target value. While there are other techniques to change the power profile of a CMP system, such as core folding or per-core power gating [39], here, we focus on DVFS due to its popularity and support in today's multicore systems and operating systems.

III. DVFS Based Dynamic Reliability Management

In this section, we describe the DVFS based dynamic reliability management scheme that we propose to investigate. The idea of dynamic reliability management is to continuously monitor the CMP system and then periodically make decisions to *update* or *tune* different control knobs with the goal of shifting the system's operation to a mode where lifetime reliability is as close as possible to a desired value that is usually set by user. Such a target lifetime reliability is usually reached after several *control periods* because of the inertia or delay it takes for different portions of the CMP chip to heat-up or cool-off. The challenging aspect of any DRM scheme is to achieve the above goal with minimal performance penalty and hardware overheads. Consequently, even though the idea of dynamic reliability management is relatively new, there have been several attempts proposed to address that challenge. In this paper, our objective is to investigate the use of DVFS as a control knob to dynamically control lifetime reliability of CMPs seen as the unified combination of both cores and networks-on-chip.

There are two very important aspects regarding the construction of the DRM scheme that need to be emphasized. First, in order to be able to use it in real time, the DRM scheme must be very efficient such that its runtime overhead is very small and therefore performance is not significantly affected by the time it takes 1) to estimate current lifetime reliability and 2) to make decisions to update voltages and frequencies. To estimate reliability *statically*, we adopt the lifetime reliability estimation approach proposed in [26] because, as illustrated in Fig.1.a, it treats the CMP system as the combination of both cores and network-on-chip. In other words, this approach does not rule out major components that can become lifetime reliability bottlenecks, thereby minimizing estimation errors of MTTF of the whole CMP system[2]. Obviously, using a reliability estimation approach as illustrated in Fig.1.a *dynamically* is not practical due to the rather long computational runtimes of McPAT power calculator [41], HotSpot temperature calculator [42], and REST lifetime reliability calculator tool [43]. Therefore, similar to [26], in the DRM scheme investigated in this paper, we use a neural network (NN) based *estimator* as shown in Fig.1.b. The NN based lifetime reliability estimator is very efficient because it translates to only the evaluation of a function that takes as input the activity counters and router powers (as indicated in Fig.1.a) as well as specific weights that are computed *statically* during the training process[3].


GEM5
Benchmark
CMP Arch. Cores + NoC
CMP Full System Simulation
Ruby Garnet NoC
Activity counters
McPAT Cores Power Estimation
Cores power
Routers power
Floorplan
HotSpot Temperature Estimation
Temperatures
Reliability Estimation Tool
MTTF
Replaced with neural network based MTTF estimator for DRM scheme

(a)

CMP = Cores + NoC
V3, f3
V1, f1
V2, f2
Next Voltages & Frequencies
Current Core activity counters & Router powers
DRM
DVFS Controller
Current MTTF
Neural Network Reliability Estimator
User set target MTTF

(b)

Fig. 1. (a) Block diagram of complete flow to *statically* estimate lifetime reliability (measured as MTTF) of the whole system as combination of cores plus network-on-chip, (b) Dynamic reliability management scheme uses DVFS controller to set voltages and frequencies of individual tiles in the next control period such that *current* MTTF approaches *target* MTTF. The CMP systems is composed of a number of tiles. A tile is the combination of one core and one NoC router.

The second important aspect regarding the construction of the DRM scheme is that to ensure that lifetime reliability estimations are accurate (and therefore the entire scheme to ultimately be accurate), we must include in such estimations both cores and network-on-chip because they are interdependent components of the same system. This is precisely what we do in our DRM implementation. This is very important because, as reported in [26], disregarding any of the two components during lifetime reliability estimation is prone to errors that can be as high as 60%, thereby significantly misleading any technique that attempts to optimize lifetime reliability.

The block diagram of the DRM scheme investigated in this paper is shown in Fig.1.b. It is essentially implemented as a control algorithm inside our customized Gem5 based full system simulation framework. During a regular simulation of a given benchmark, for a given architecture of the CMP, infor-

[2] Note that the vast majority of previous work focused on either cores as the computational component or on network-on-chip or bus as the communication component.

[3] Note that in real systems, the input to the NN based lifetime reliability estimator would be directly the temperatures collected by sensors placed on the CMP chip. Here, in the context of working with the Gem5 simulator [40] − which allows us to conduct investigative studies on CMP architectures that are not yet available − we must use activity counters and routers power.

mation about the activity counters (i.e., instructions executed by cores) and power values for all routers of the network-on-chip is used as input into the neural network based MTTF estimator. The projected or estimated MTTF is compared to the desired target MTTF by the DVFS controller, which then decides for each core whether the clock frequency must be throttled, increased, or left unchanged. The logic behind the DVFS controller is simple: if the estimated current MTTF is less than the target MTTF, then, throttle the frequency of the core to the next lower frequency from the set of frequencies we work with (and lower its supply voltage too); otherwise, raise the frequency to the next higher frequency (and raise its supply voltage too); if the estimated current MTTF is within the vicinity (dictated though a user set parameter δ) of the target MTTF, then keep the same frequency for the core. The pseudocode of this control algorithm is shown in Fig.2.

```
Algorithm: DRM Scheme
In: Desired $MTTF_{target}$; $\delta$ hysteresis bandwidth; $\gamma$ maximum percentage of updated tiles in a control period; core activity counters and routers power
Out: Frequencies and supply voltages for all tiles for next control period
Use neural network based MTTF estimator to find current MTTF of each tile and of whole CMP
if $MTTF_{CMP} < MTTF_{target} - \delta$ then
    Sort all tiles in increasing order of their MTTF
    for $i \leftarrow 1$ to $\gamma n$ do // n: number of tiles
        if $MTTF_i < MTTF_{target} - \delta$ then
            Switch down frequency and voltage of this tile
        end if
    end for
else if $MTTF_{CMP} > MTTF_{target} + \delta$ then
    Sort all tiles in decreasing order of their MTTF
    for $i \leftarrow 1$ to $\gamma n$ do
        if $MTTF_i > MTTF_{target} + \delta$ then
            Switch up frequency and voltage of this tile
        end if
    end for
end if
```

Fig. 2. Pseudocode of the DVFS based DRM scheme. This control algorithm is implemented as a callable routine inside the Gem5 simulation framework. Parameters δ and γ can be set by user to allow for calibration of how *aggressive* the DRM policy is.

IV. Simulation Results

We leverage existing simulations tools, Gem5 full system simulator [40] and REST reliability estimator [43], to implement the DRM scheme discussed in the previous section. We conduct full system simulations on several Parsec benchmarks [44] to investigate the DRM scheme for two different CMP architectures composed of 4 cores and 16 cores respectively. Each of these architectures use regular mesh NoCs: 2x2 and 4x4. The default architectural configuration parameters utilized in our custom Gem5 based simulations, unless otherwise specified, are shown in Table I.

A. Dynamic Reliability Management

In our simulations, we set as target or desired average MTTF a value that is with 100% longer than what it is when no DRM is applied, which is our reference case. In other words, we are interested in doubling the average lifetime of the investigated CMP architectures. Fig.3 – Fig.6 show the simulation results for *blackscholes*, *canneal*, *bodytrack*, and *dedup* Parsec benchmarks run as applications with 16 threads on a CMP architecture with 4x4 tiles. The plots show only the period of time that covers the so called region of interest (ROI) of the Gem5 simulation. For each simulation shown in these figures, the Gem5 simulator is stopped a number of times during the ROI (this number depends on the actual length of the ROI and the selected *control period* discussed in the previous section) to perform DRM and update the frequencies and voltages of each tile. Each of these stop-times corresponds to a data point out of the *sampling points* shown on the horizontal axis in Fig.3 – Fig.6.

TABLE I. Architectural configuration parameters.

Parameter	Value
Technology node	65nm
Frequencies	2GHz downto 1.4GHz, with 100MHz step
VDDs	1.1V downto 0.95V, with 25mV step
Core	Alpha EV6 21264
Core CPU model	Out of order (Detailed CPU)
Branch predictor	2 bit counter
Reorder buffer	80-entries
L1 ICache	32KB
L1 DCache	64KB
L2	2MB
Network	2D regular mesh, 1 router per core
Tile floorplan	Router to the top of core ALU
Link bandwidth	32 bits
Routing algorithm	XY
Number of virtual channels (VCs)	2

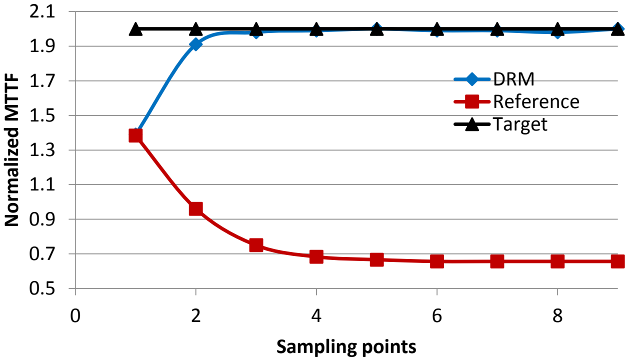


Fig. 3. Gem5 with DVFS based DRM simulation of *blackscholes* benchmark.

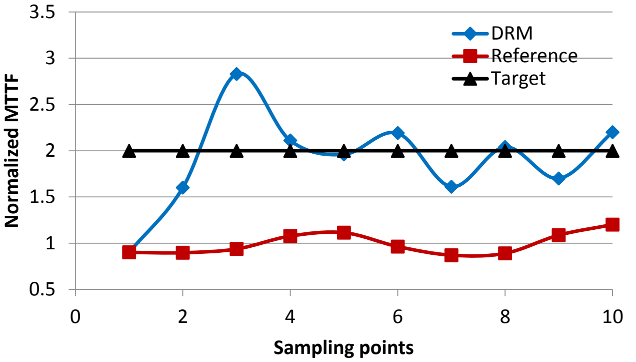


Fig. 4. Gem5 with DVFS based DRM simulation of *canneal* benchmark.

We note that for some benchmarks the MTTF fluctuates around the target MTTF. This is for example the case of *bodytrack* and *dedup* benchmarks. We suspect that this is primarily due to the variation in the workload that each core must do

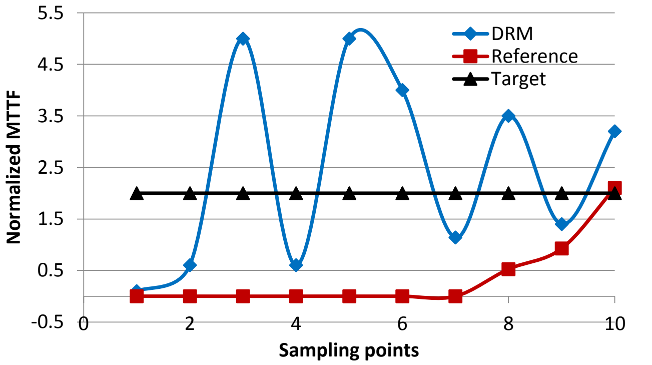


Fig. 5. Gem5 with DVFS based DRM simulation of *bodytrack* benchmark.

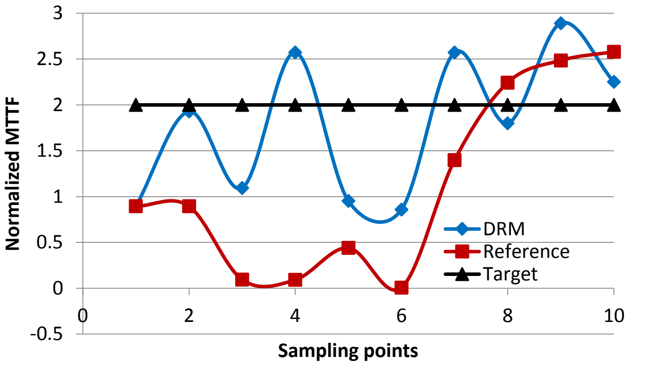


Fig. 6. Gem5 with DVFS based DRM simulation of *dedup* benchmark.

for these particular benchmarks during different control periods inside the ROI. This may also be as a result of the changes in dependencies created by frequency throttling among jobs that are executed on different cores. We noticed that when all cores are loaded with work uniformly throughout the ROI (as is the case of *blackscholes* and *canneal* benchmarks), the overall MTTF is more stable. Better calibration of the proposed DRM algorithm from Fig.2 can help address such fluctuations.

Table II summarizes the information presented in these plots. The performance penalty includes the time spent to perform the reliability estimation as shown in Fig.1.b and to execute the DRM algorithm presented in Fig.2.

TABLE II. SUMMARY OF SIMULATIONS SHOWN IN FIG.3 – FIG.5

Benchmark	Avg. MTTF improv.	Perf. penalty	ROI exec. time (reference run)	Gem5 sim. time
blackscholes	100%	11.8%	69 ms	6 h
canneal	100%	16.96%	103 ms	12 h
bodytrack	100%	9.3%	139 ms	9 h
dedup	100%	15.8%	376 ms	18 h

B. Discussion

The results indicate that lifetime reliability can be effectively improved using DVFS based DRM schemes. However, this improvement is at the expense of some performance penalty, as shown in Table II. When compared to the thread migration based DRM scheme studied in [26][4], we note that the DVFS based DRM scheme is able to improve MTTF more but at larger performance penalties (the largest performance penalty reported in [26] is 9.16%). This suggests that, for applications where performance degradation is not acceptable, a thread migration based DRM scheme may be a better choice. In applications where performance degradation can be tolerated, the proposed DVFS based DRM scheme can be used to trade performance for larger MTTF improvements. Note that, *frequency throttling* can theoretically improve MTTF a lot – at the limit, if cores are completely stopped, MTTF becomes infinity. On the other hand, *thread migration* is limited in its ability to significantly improve MTTF even if it would be acceptable to degrade performance – that is because no matter how much one could shuffle jobs among cores, if the benchmark is computationally intensive and all cores are heavily utilized, temperature profile will be high anyways.

V. CONCLUSION

The main contribution of this paper is the investigation of DVFS as a control technique to develop dynamic reliability management schemes for chip multiprocessors (CMPs) to address TDDB and NBTI aging failure mechanisms. A notable merit of this study is that the online NN based reliability estimation is done in a unified manner: CMP systems are treated as the combination of both cores and network-on-chip, because each of these components can become the weakest link from a lifetime reliability perspective. Simulation results showed that lifetime can be doubled with acceptable performance degradation. This suggests that DVFS offers a better way to trade lifetime with performance degradation than thread migration based techniques.

ACKNOWLEDGMENT

This work was supported by the National Science Foundation (NSF), grant CCF-1360439. Any findings and conclusions or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the NSF.

REFERENCES

[1] A. DeHon, H.M. Quinn, and N.P. Carter, "Vision for cross-layer optimization to address the dual challenges of energy and reliability," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, March 2010.

[2] F. Angiolini, D. Atienza, S. Murali, L. Benini, and G. De Micheli, "Reliability support for on-chip memories using Networks-on-Chip," *IEEE Int. Conf. on Computer Design (ICCD)*, Oct. 2007.

[3] R. Vadlamani, J. Zhao, W. Burleson, and R. Tessier, "Multicore soft error rate stabilization using adaptive dual modular redundancy," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, March 2010.

[4] S. Feng, S. Gupta, A. Ansari, and S. Mahlke, "Shoestring: probabilistic soft error reliability on the cheap," *Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, March 2010.

[4] We consider the DRM scheme from [26] for comparison purposes because it is the only other DRM scheme that considers both cores and NoC in a unified manner. All other previous DRM schemes do not include the NoC component in their optimization, and therefore their reported MTTF values may be off by as much as 60% as reported in [26].

[5] J. Blome, S. Feng, S. Gupta, and S. Mahlke, “Self-calibrating online wearout detection," *Int. Symp. on Microarchitecture (MICRO)*, pp. 109-120, Dec. 2007.

[6] Y. Li, Y.M. Kim, E. Mintarno, D.S. Gardner, and S. Mitra, “Overcoming early-life failure and aging for robust systems," *IEEE Design & Test of Computers*, vol. 26, no. 6, pp. 28-39, 2009.

[7] B. Datta and W. Burleson, “Circuit-level NBTI macro-models for collaborative reliability monitoring," *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, May 2010.

[8] M. Choudhury, V. Chandra, K. Mohanram, and R. Aitken, “TIMBER: time borrowing and error relaying for online timing error resilience," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, March 2010.

[9] L. Huang and Q. Xu, “Energy-efficient task allocation and scheduling for multi-mode MPSoCs under lifetime reliability constraint," *ACM Int. Conference on Design Automation and Test in Europe (DATE)*, 2010.

[10] S. Wang and J.-J. Chen, “Thermal-aware lifetime reliability in multicore systems," *Int. Symp. on Quality Electronic Design (ISQED)*, 2010.

[11] A. Masrur et al., “Schedulability analysis for processors with aging-aware autonomic frequency scaling," *IEEE Int. Conf. on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2012.

[12] A. Das, A. Kumar, and B. Veeravalli, “Reliability-driven task mapping for lifetime extension of networks-on-chip based multiprocessor systems," *ACM Int. Conference on Design Automation and Test in Europe (DATE)*, 2013.

[13] A. Das, A. Kumar, B. Veeravalli, C. Bolchini, and A. Miele, “Combined DVFS and mapping exploration for lifetime and soft-error susceptibility improvement in MPSoCs," *ACM Int. Conference on Design Automation and Test in Europe (DATE)*, 2014.

[14] Jayanth Srinivasan, Lifetime reliability aware microprocessors, Ph.D. Thesis, University of Illinois at Urbana-Champaign, 2006.

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

[16] E. Karl, D. Blaauw, D. Sylvester, and T. Mudge, “Multi-mechanism reliability modeling and management in dynamic systems," *IEEE Trans. on Very Large Scale Integration Systems (TVLSI)*, vol. 16, no. 4, 2008.

[17] 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 (HiPEAC)*, 2010.

[18] A. Tiwari and J. Torrellas, “Facelift: hiding and slowing down aging in multicores," *ACM/IEEE Int. Symp. on Microarchitecture (MICRO)*, 2008.

[19] C. Zhuo, D. Sylvester, and D. Blaauw, “Process variation and temperature-aware reliability management," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, 2010.

[20] A.K. Coskun, R.D. Strong, D.M. Tullsen, and T.S. Rosing, “Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," *SIGMETRICS/Performance*, 2009.

[21] J. Sun, A.K. Kodi, A. Louri, and J.M. Wang, “NBTI aware workload balancing in multi-core systems," *IEEE Int. Symp. on Quality Electronic Design (ISQED)*, 2009.

[22] O. Khan and S. Kundu ,“A self-adaptive system architecture to address transistor aging," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, 2009.

[23] P. Mercati, A. Bartolini, F. Paterna, T.S. Rosing, and L. Benini, “Workload and user experience-aware dynamic reliability management in multicore processors," *ACM Int. Design Automation Conference (DAC)*, 2013.

[24] P. Mercati et al., “A Linux-governor based dynamic reliability manager for android mobile devices," *ACM/IEEE Design Automation and Test in Europe Conf. (DATE)*, 2013.

[25] A. Das, R. A. Shafik, G. V. Merrett, B. M. Al-Hashimi, A. Kumar, and B. Veeravalli, “Reinforcement learning-based inter- and intra-application thermal optimization for lifetime improvement of multicore systems," *ACM Int. Design Automation Conference (DAC)*, 2014.

[26] A.Y. Yamamoto and C. Ababei, “Unified reliability estimation and management of NoC based chip multiprocessors," *Microprocessors and Microsystems*, vol. 38, no. 1, pp. 53-63, Feb. 2014.

[27] J.H. Stathis, “Reliability limits for the gate insulator in CMOS technology," *IBM J. of Research and Development*, vol 46, pp. 265-286, 2002.

[28] E. Wu, J. Sune, W. Lai, E. Nowak, J. McKenna, A. Vayshenker, and D. Harmon, “Interplay of voltage and temperature acceleration of oxide breakdown for ultra-thin gate oxides," *Solid-State Electronics*, vol. 46, no. 11, pp. 1787-1798, 2006.

[29] D.K. Schroder and J.A. Babcock, “Negative bias temperature instability: road to cross in deep submicron silicon semiconductor manufacturing," *J. of Applied Physics*, vol. 98, no. 1, pp. 1-18, 2003.

[30] S. Zafar, B. Lee, J. Stathis, A. Callegar, and T. Ning, “A model for negative bias temperature instability (NBTI) in oxide and high k pFETs," *Int. Symposium on VLSI Technology*, 2004.

[31] D.N. Truong et al., “A 167-processor computational platform in 65 nm CMOS," *IEEE J. of Solid-State Circuits*, vol. 44, no. 4, pp. 1-15, Apr. 2009.

[32] S.S. Majzoub, R.A. Saleh, S.J.E. Wilton, and R.K. Ward, “Energy optimization for many-more platforms: communication and PVT aware voltage-island formation and voltage selection algorithm," *IEEE Trans. on TCAD*, vol. 29, no. 5, pp. 816-829, May 2010.

[33] A.K. Mishra, S. Srikantaiah, M. Kandemir, and C.R. Das, “Coordinated power management of voltage islands in CMPs," *SIGMETRICS*, 2010.

[34] F. Zanini, D. Atienza, L. Benini, and G. De Micheli, “Multicore thermal management with model predictive control," *European Conference on Circuit Theory and Design*, 2009.

[35] L. Shang, L.-S. Peh and N. K. Jha, “Dynamic voltage scaling with links for power optimization of interconnection networks," *IEEE Int. Symp. on High-Performance Computer Architecture (HPCA)*, 2003.

[36] S.E. Lee and N. Bagherzadeh, “A variable frequency link for a power-aware network-on-chip (NoC)," *Integration*, vol. 42, no. 4, pp. 479-485, 2009.

[37] J. Murray, P.P. Pande, and B. Shirazi, “DVFS-enabled sustainable wireless NoC architecture," *IEEE Int. System-on-Chip Conference (SOCC)*, 2012.

[38] U.Y. Ogras, R. Marculescu, D. Marculescu, and E.G. Jung, “Design and management of voltage-frequency island partitioned Networks-on-Chip," *IEEE Trans. on VLSI Syst.*, vol. 17, no. 3, pp. 330-341, 2009.

[39] A. Vega, A. Buyuktosunoglu, H. Hanson, P. Bose, and S. Ramani, “Crank it up or dial it down: coordinated multiprocessor frequency and folding control," *MICRO*, 2013.

[40] N. Binkert, B. Beckmann, G. Black, S.K. Reinhardt, A. Saidi, A. Basu, J. Hestness, D.R. Hower, T. Krishna, S. Sardashti, R. Sen, K. Sewall, M. Shoaib, N. Vaish, M. D. Hill, D.A. Wood, “The gem5 simulator," *ACM SIGARCH Computer Architecture News Archive*, 2011.

[41] S. Li, J.H. Ahn, R.D. Strong, J.B. Brockman, D.M. Tullsen, N.P. Jouppi, “McPAT: an integrated power, area, timing modeling framework for multicore and manycore architectures," *IEEE/ACM Int. Symposium on Microarchitecture (MICRO)*, 2009.

[42] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron and M.R. Stan, “HotSpot: a compact thermal modeling method for CMOS VLSI systems," *IEEE Trans. on Very Large Scale Integration Systems (TVLSI)*, vol. 14, no. 5, 2006.

[43] REST: Reliability ESTimation for chip multiprocessors (CMPs), 2014. [Online]. Available: https://code.google.com/p/reliability-estimator

[44] M. Gebhart, J. Hestness, E. Fatehi, P. Gratz, and S.W. Keckler, “Running PARSEC 2.1 on M5," The University of Texas at Austin, *Technical Report TR-09-32*, Oct. 2009.