2017 IEEE International Symposium on High Performance Computer Architecture

BRAVO: Balanced Reliability-Aware Voltage Optimization

Karthik Swaminathan*, Nandhini Chandramoorthy$^{\Psi}$, Chen-Yong Cher*,
Ramon Bertran*, Alper Buyuktosunoglu* and Pradip Bose*

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598

Email: *{kvswamin, chenyong, rbertra, alperb, pbose}@us.ibm.com, $^{\Psi}$Nandhini.Chandramoorthy@ibm.com

ABSTRACT

Defining a processor micro-architecture for a targeted product space involves multi-dimensional optimization across performance, power and reliability axes. A key decision in such a definition process is the circuit- and technology-driven parameter of the nominal (voltage, frequency) operating point. This is a challenging task, since optimizing individually or pair-wise amongst these metrics usually results in a design that falls short of the specification in at least one of the three dimensions. Aided by academic research, industry has now adopted early-stage definition methodologies that consider both energy- and performance-related metrics. Reliability-related enhancements, on the other hand, tend to get factored in via a separate thread of activity. This task is typically pursued without thorough pre-silicon quantifications of the energy or even the performance cost. In the late-CMOS design era, reliability needs to move from a post-silicon afterthought or validation-only effort to a pre-silicon definition process. In this paper, we present *BRAVO*, a methodology for such reliability-aware design space exploration. BRAVO is supported by a multi-core simulation framework that integrates performance, power and reliability modeling capability. Errors induced by both soft and hard fault incidence are captured within the reliability models. We introduce the notion of the Balanced Reliability Metric (BRM), that we use to evaluate overall reliability of the processor across soft and hard error incidences. We demonstrate up to 79% improvement in reliability in terms of this metric, for only a 6% drop in overall energy efficiency over design points that maximize energy efficiency. We also demonstrate several real-life use-case applications of BRAVO in an industrial setting.

1. INTRODUCTION

Reliability-aware design is fast becoming the predominant paradigm in industry, for the realization of next- generation processors. The need for fast, energy-efficient processing, coupled with variations in temperature, altitude as well as harsh deployment conditions have made processor reliability a key design parameter alongside energy and performance across a range of processing platforms, from embedded systems to server-class processors and hyperscale systems. In addition, as we approach the limits of technology scaling, the effect of increased power density and reduction in the charge-retaining capacity of transistors, have resulted in significant concerns for processor reliability. In particular, these include an increasing degree of vulnerability to radiation-induced soft errors and hard errors due to aging effects.

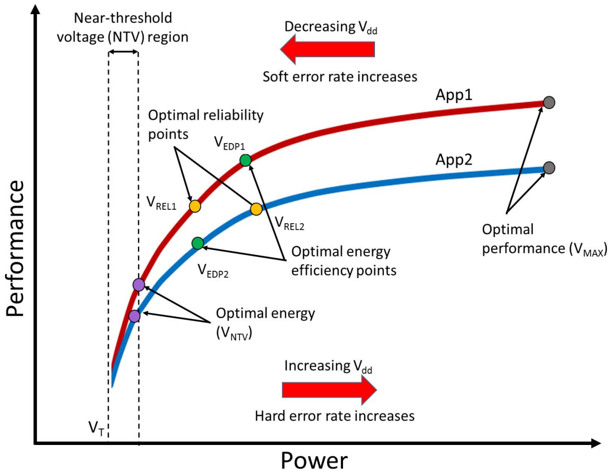


Figure 1: Impact of reliability considerations on the power-performance tradeoff curve. The different power/performance values are obtained by changing the processor supply voltage (V_{dd}). The V_{dd}s that yield the optimal energy (V_{NTV}), energy-delay product (EDP) (V_{EDP1}, V_{EDP2}), performance (V_{MAX}) and reliability (V_{REL1}, V_{REL2}) are shown in the figure. It is clear that operating at maximum energy efficiency may not necessarily result in optimal reliability. For example, $V_{REL1} < V_{EDP1}$ and $V_{REL2} > V_{EDP2}$

Enabling designers to simultaneously study performance, energy and reliability characteristics for different applications helps in making key design decisions such as the type of core, number of cores, optimal operating voltage and frequency early in the processor design stage. Most state-of-the-art design paradigms used in industry and academia today, simply involve optimizing only for performance and/or energy. One such prevalent paradigm is that of *Near- Threshold Computing (NTC)*, where processors are operated close to the threshold voltage V_{th}. It has been demonstrated in works such as [15, 16], that operating the processor in the Near-Threshold Voltage (NTV) region yields the minimum energy. However, in addition to the limitations in frequency, selecting the optimal operating voltage for a processor from the point of view of minimizing energy alone, may result in adverse implications for reliability, particularly due to soft errors. Increase in V_{dd} reduces the soft error rate, but increases hard error rate, and therefore the overall reliability of the processor increases or decreases, depending on which effect is more dominant.

2378-203X/17 $31.00 © 2017 IEEE
DOI 10.1109/HPCA.2017.56

Figure 1 illustrates the tradeoffs involved when designing for reliability, in addition to energy (or power) and performance considerations. It shows the variation in performance versus power with changing V_{dd} for two different applications (*App1* and *App2*). Each voltage corresponds to a fixed frequency of operation for the given processor. As described above, near-threshold operation at minimum energy may result in poor reliability due to increased soft error vulnerability. On the other hand, operating at the peak performance point (V_{MAX}) will not only result in a huge increase in power, but also cause potential failures due to increased temperature and accelerated aging effects. In many cases, the Energy-Delay Product (EDP) has been adopted in industry as the primary optimization metric, as it accounts for both energy and performance. However, this point too, may not be the best choice when reliability is considered, as shown in the figure. In the case of *App1*, the relatively higher impact of aging-related hard errors causes the optimal reliability voltage (V_{REL1}) to be less than the corresponding EDP-optimal voltage (V_{EDP1}), while *App2*, being more vulnerable to soft errors, has a higher optimal reliability point (V_{REL2}) than the corresponding EDP-optimal point (V_{EDP2}).

Based on individual estimates of vulnerability due to each of these phenomena, designers adopt circuit and architectural techniques to mitigate the probability of failures [12, 34]. However, designs that incorporate these techniques to maintain a desired level of reliability usually come with huge performance and energy costs. Hence, it is preferable to implement these techniques after determining the optimal reliability-aware voltage point, so as to minimize these overheads.

Determining the reliability-aware optimal Vdd point at an early stage of the design enables the designers to selectively implement resilience strategies such as checkpoint-restart, latch-hardening or selective duplication mechanisms in conjunction with voltage optimization in order to meet the desired resilience targets. It also helps optimize the extent of voltage guard-band that is applied in order to mitigate runtime errors.

Determination of this optimal point can be a challenging task due to several reasons. The relation between different factors such as soft errors and hard errors, that constitute processor reliability may not be easily formulated mathematically, making it difficult to realize an integrated reliability metric. For example, aging-related effects such as Time-Dependent Dielectric Breakdown (TDDB) are qualified as parts-per-million-defective (ppm) while soft errors are qualified in terms of the critical charge (Q_{crit}) or Failures In Time (FITs), if a use case model is provided [41]. Further, there is a lack of an integrated infrastructure that can jointly study the tradeoffs between performance, power and reliability in terms of hard and soft errors. Particularly in case of reliability, today, hard and soft errors for a given process technology are characterized only at the conservative worst-case design points at the device level, often in different metrics, before being passed on to the designers, without considering the relationship between the application performance and power characteristics.

In light of these constraints, we analyze a range of factors that impact processor reliability and carry out a joint evaluation of performance, power and reliability to determine the globally optimal operating V_{dd} for different processor architectures. The main contributions of this paper can be summarized as follows:

- We analyze different factors that influence reliability, both individually and jointly, with the help of *BRAVO*, our integrated cross-layer infrastructure comprised of industrial-strength simulation tools. We thus reduce the multi- dimensional complexity of these tradeoffs and provide specific design inputs to future processor design across a host of applications. In doing so, we propose the *Balanced Reliability Metric*, or BRM, a composite metric that enables us to jointly analyze the competing trends of the different factors influencing reliability.
- Using *BRAVO*, we estimate the optimal operating voltage for processors that comprise of simple as well as complex cores. We evaluate the performance and power costs that result from adopting a reliability-unaware optimal operating point. We demonstrate that the optimal V_{dd} can vary substantially from existing reliability-agnostic design points.
- By means of this work, we aim to provide new insights that could be useful in the early stage design of next-generation processors, particularly for IBM POWER systems, in terms of incorporating reliability as a design parameter and determining the optimal V_{dd} to be selected.
- Finally, we demonstrate real-life applications in industry from domains such as high-performance and embedded computing that trade off between performance, power and reliability when error mitigation overheads are taken into account. We show how

The rest of the paper is as follows. Section 2 provides an overview of the various sources of vulnerability considered in this paper. Section 3 describes the methodology we propose for evaluating each individual reliability metric as well as techniques to carry out a multi-dimensional analysis of the different sources of processor vulnerability. Section 4 details the architectural configuration of our evaluation platforms and our simulation infrastructure, Section 5 shows the results of our analyses and Section 6 demonstrates application of the presented framework towards real-life use cases in high performance and embedded domains. Section 7 describes the related work and we conclude with Section 8.

2. SOURCES OF VULNERABILITY IN A PROCESSOR

Technology node migration over the past few generations has progressed along a path that has exposed greater challenges for chip yield and in-field reliability. For example, deviation from Dennard scaling has meant slower decrease in supply voltages than before, which has exacerbated failure rates due to factors like oxide breakdown (TDDB). Since device densities and core counts have continued to increase, albeit at a slower rate, processor-level failure rates also continue to increase, even if some component failure rates have remained constant or decreased with scaling. Hence, it is important to incorporate reliability metrics into the processor design targets, in addition to performance and power efficiency. In this work, we quantify processor reliability in terms of its potential sources of vulnerability, which, for the purposes of this paper, we categorize (broadly) into two classes: radiation-induced soft errors and aging-induced hard errors.

2.1 Soft Errors

Radiation-induced soft errors may be caused due to incident neutrons from cosmic radiation sources, or alpha particles radiated from trace elements present in packaging materials. Even a single bit flip due to soft errors can have potentially hazardous implications for the processor and the integrity of the program being executed. There are various mechanisms that offer protection against soft errors, ranging from the circuit to the processor-level abstraction such as Razor circuits [17], ECC and application checkpoint and replay mechanisms [12, 34], to Algorithm Based Fault Tolerance (ABFT) [33]. In addition only a small fraction of the bit-flips that occur in the processor pipeline can impact the output. Consequently, most of the errors are benign or *derated*. This derating factor depends both on the architecture as well as the application being executed.

2.2 Hard Errors

Unlike radiation-induced errors which are transient in nature, failures due to manufacturing defects or those that evolve over a period of continued operation are permanent in nature. The lifetime of a processor can be adversely affected by aging and wearout effects. These phenomena are mainly due to the properties of the materials and wearout due to sustained operation. We consider hard errors to be composed of several failure mechanisms, the most dominant being:

- **Electromigration (EM):** This phenomenon occurs in the metal interconnects due to the movement of conducting electrons. The momentum of the electrons forces the migration of the metal atoms of the conducting wires as well, which can cause irregularities in the distribution of material across the wire. This in turn may result in undesired open and short circuits across the processor. The Mean-Time-To-Failure (MTTF) due to electromigration at a point on an interconnect is proportional to the current density at that point and has an exponential relation to the temperature. The corresponding FIT rate is equal to the reciprocal of the MTTF for exponentially distributed occurrences of failures. It can be expressed mathematically in terms of *Black's equation* [14] as:

$$FIT_{EM} = (A.j^{-n}.e^{\frac{Q}{kT}})^{-1} \qquad (1)$$

  Here, j is the current density, Q is the activation energy in eV, k is the Boltzmann constant, T is the temperature in K, while A and n are empirical constants.

- **Time Dependent Dielectric Breakdown (TDDB):** This phenomenon occurs due to the wear out of the gate dielectric over time, resulting in a conducting path across the dielectric. This would cause the transistor to be permanently shorted. The FIT rate increases with increases in voltage and temperature and is estimated by means of the model proposed in [45]. It is expressed in terms of voltage V_{gs}, temperature T, duty cycle D, and fitting constants A_{TDDB}, a, b, X, Y and Z, as:

$$FIT_{TDDB} = (\frac{1}{D}.A_{TDDB}.V_{gs}^{(-a+bT)}.e^{\frac{X+\frac{Y}{T}+ZT}{kT}})^{-1} \qquad (2)$$

- **Negative Bias Temperature Instability (NBTI):** This occurs primarily in PFETs in CMOS logic. When the gate is biased negatively with respect to the source, positive charges accumulate in the gate oxide, causing a rise in the threshold voltage. This in turn reduces the drive current, causing slower transistor switching, which may eventually result in timing errors. We adopt the model used in [42] for the purpose of estimating the effect of NBTI. In case of EM and TDDB, a single device (or via, in case of EM) can be used as a reference circuit or device to estimate the metrics. In case of NBTI, however, an inverter chain (of N_{inv} stages) is adopted as the reference circuit and its FIT rate is determined as shown:

$$FIT_{NBTI} = 10^9\left(\frac{K}{\Delta V_{T\_ref}}\right)^{\frac{1}{n}}, where \qquad (3)$$

$$K = A_{NBTI}.t_{ox}.\sqrt{C_{ox}.|V_{gs}-V_T|}.e^{\frac{E_{ox}}{E_0}}.e^{-\frac{E_{\alpha\_NBTI}}{kT}}$$

$$\Delta V_{T\_ref} = 0.01.\frac{N_{inv}.(V_{dd}-V_T)}{\alpha}$$

  Here, A_{NBTI} and n are empirical constants, α, t_{ox}, C_{ox}, E_{ox}, E_0, and E_{α_NBTI} are fixed technological parameters, V_T is the threshold voltage, and V_{gs} is assumed to be equivalent to the supply voltage V_{dd}.

Works such as [45] combine the various aspects of lifetime reliability into a single FIT value, using the *Sum-Of-Failure-Rates (SOFR)* model. However, this makes several assumptions such as exponential arrival rates of failures, which may not be practical. In addition, these metrics are not entirely correlated, as will be shown in Section 5. Hence we consider EM, TDDB and NBTI to be components of reliability that are evaluated separately.

In addition to these sources of error, variations in the supply voltage level are observed on account of non-idealities in the Power Delivery Network (PDN), resulting in an *IR drop* and time-varying fluctuations across the network known as *di/dt droop*. Works such as [53] have characterized the effect of this voltage noise, particularly at near-threshold voltage where it is exacerbated. Furthermore, it is important to note that, at every operating voltage and frequency point, there are guard-bands that are added to prevent potential timing violations due to large di/dt droops. *Hence, this paper primarily focuses on radiation-induced soft errors and aging-induced hard errors to be a component of our combined reliability metric and any application-dependent effects of voltage noise are not examined.*

3. METHODOLOGY

In this section, we describe the integrated cycle-accurate simulation platform with plug-in energy and reliability modeling capabilities. The input to our framework comprises of an application (trace) and the output consists of detailed performance, power, temperature and reliability characteristics of the processor for a range of selected architectures. The objective of such a framework is to enable us to determine the optimal operating point from the point of view of all these characteristics for the applications being run.

3.1 Estimating individual reliability metrics

We analyze all applications by running them across the spectrum of permissible processor operating voltages and frequencies for each of the reliability phenomena, namely aging-induced hard errors and radiation-induced soft errors, as described below:

- Analysis of hard errors: This involves estimating errors due to aging-related effects of EM, TDDB and NBTI, which together constitute hard errors. Their characterization requires the thermal and power profiles of representative applications run on the processor, from which the FIT rate is calculated analytically. We then estimate the maximum FIT value across the processor grid for each particular application/supply voltage configuration.

- Analysis of soft errors: Based on the latch-level information for each microarchitecture component, component-level residency statistics, additional derating factors due to the functionality of the component (high derating for speculative instructions, for instance) and application derating characteristics, we can determine the peak SER for different applications and operating voltages.

While the analysis described above would provide us with insights as to which configurations are unacceptable, it is not possible to determine the optimal reliability point from this method by analyzing each reliability metric individually. Therefore, we jointly study all aspects of processor reliability along with performance and power, by determining the *Balanced Reliability Metric* or BRM for different applications running at different operating V_{dd}s.

3.2 Estimating balanced reliability metric

Since the data sets corresponding to each individual reliability metric can be considered to be sets of (partially) correlated observations, we use Principal Component Analysis (PCA) [4] to combine the individual metrics into a unified *Balanced Reliability Metric* (BRM). Each data set is associated with a fixed threshold, which may be a user-defined parameter that sets the individual reliability limits of the desired processor. The threshold quantifies a user-specified tolerance limit on processor vulnerability on account of each soft and hard error metric. The algorithm for determining the *Balanced Reliability Metric* is described in Algorithm 1. Here, the inputs are a $N \times 4$ matrix containing SER, EM, TDDB and NBTI FIT rates for N observations, normalized with respect to the corresponding standard deviation (σ) across all applications and operating voltage configurations, and a 1×4 vector containing user-defined thresholds for each metric. PCA is performed on the centered covariance matrix of the input data to obtain the components of the input data in directions of maximum variance of the data points. We consider only the first few principal components that cumulatively account for *VarMax* proportion of the variance, thereby reducing the dimensions. We also determine the observations from the reduced two-dimensional component matrix, that violate the user-defined thresholds in the projected PCA space. Finally, we compute the L2-norm across the reduced dimensions to obtain a combined BRM, which quantifies the processor vulnerability on account of four factors, namely radiation-induced soft errors and the three aging-related sources of hard error. Note it is also possible to obtain similar results using statistical techniques other than PCA, such as Partial Least Squares (PLS) [3] and Common Factor Analysis (CFA). It is interesting to note that such a reliability-aware optimal V_{dd} may turn out to be different from the one that is only power-performance optimized, as is done in most prior studies.

Algorithm 1 Pseudocode for obtaining composite reliability metric

```
procedure ($Data, Threshold$)
    $RelData \leftarrow Data / stdev(Data)$
    $MeanSubRelData \leftarrow RelData - mean(RelData)$
    $RelThreshold \leftarrow (Threshold / stdev(Data)) - mean(RelData)$
    $[EigenVectors, EigenVals] \leftarrow PCA(MeanSubRelData)$
    $PCAThreshold \leftarrow RelThreshold \times EigenVectors$
    $PCAData \leftarrow MeanSubRelData \times EigenVectors$
    while $(i <= 4)$ do
        $VarCoverage += EigenVals[i] / sum(EigenVals)$
        if $VarCoverage > VarMax$ then
            break;
        else $i++$
    $ViolatingComponents \leftarrow PCAData[find(PCAData \geq PCAThreshold)]$
    $BRM \leftarrow L2Norm(PCAData[:, 1:i])$
    return $BRM, ViolatingComponents$
```

4. DESCRIPTION OF EVALUATED PROCESSOR PLATFORMS

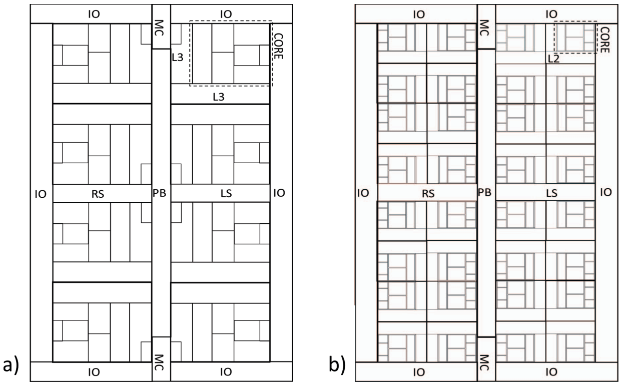


Figure 2: Representative layout of target architectures for a) Complex core and b) Simple core based processors. The processor bus (PB), memory controller (MC), Local and Remote SMP links (LS and RS respectively) and the I/Os are assumed to be similar across both processors

4.1 Architecture Description

Figures 2a) and b) show the simulated target architectures, comprising of a) 8 out-of-order complex cores and b) 32 in-order simple cores, both based on the POWER ISA. The complex core has a 3 level cache hierarchy comprising of a 32KB L1, a 256KB L2 and 4MB private L3 cache per core, while the simple core has a 16KB L1 and a 2MB shared L2 cache per core. The complex and simple cores are set to run at nominal frequencies of 3.7GHz and 2.3 GHz respectively and operate within the same voltage range, V_{MIN} to V_{MAX}, which are the minimum and maximum values attainable by the cores for both types of processors. Even though the voltage range is the same for simple and complex cores, their corresponding nominal frequencies are observed to be different. This can be attributed to the pipeline depths being different across simple and complex cores. The voltages corresponding to the nominal frequencies of the cores are termed as the nominal voltages and lie between V_{MIN} and V_{MAX}. Further details about the respective architectures can be found in [57] and [27]. The interconnect architecture, comprising of a processor bus

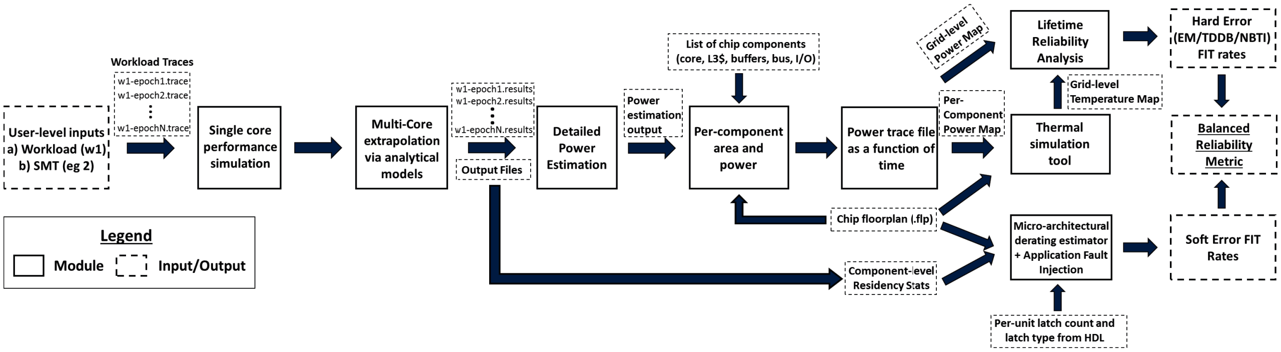

Workload Traces
w1-epoch1.trace
w1-epoch2.trace
w1-epochN.trace
User-level inputs
a) Workload (w1)
b) SMT (eg 2)
Single core performance simulation
Multi-Core extrapolation via analytical models
w1-epoch1.results
w1-epoch2.results
w1-epochN.results
Output Files
Detailed Power Estimation
Power estimation output
List of chip components (core, L3$, buffers, bus, I/O)
Per-component area and power
Power trace file as a function of time
Grid-level Power Map
Per-Component Power Map
Lifetime Reliability Analysis
Grid-level Temperature Map
Thermal simulation tool
Hard Error (EM/TDDB/NBTI) FIT rates
Balanced Reliability Metric
Chip floorplan (.flp)
Component-level Residency Stats
Micro-architectural derating estimator + Application Fault Injection
Per-unit latch count and latch type from HDL
Soft Error FIT Rates
Legend
Module
Input/Output

Figure 3: Simulation flow for integrated framework. The various modules shown in the figure correspond to individual tools used for estimating performance, power and reliability of the processor.

(PB), local and remote SMP links (LS and RS), I/O links and 2 memory controllers (MC), is assumed to be similar for the simple and complex core-based processor and is run at constant voltage, irrespective of the core V_{dd}. Since the area of 4 simple cores is roughly equal to that of 1 complex core, the two processors, henceforth referred to as *SIMPLE* and *COMPLEX*, are assumed to be iso-area, with a difference of less than 5%.

4.2 Simulation Infrastructure

Figure 3 illustrates the toolchain used for determining the performance, power and reliability parameters on the target evaluation platforms. This framework comprises of open source tools along with software-engineered modules used for integration into a single toolset for end to-end simulation of an embedded system and a server-class processor, based on the POWER ISA [57].

For performance estimation, we use the trace-based *SIM_PPC* simulator, which is available as part of the OpenPOWER initiative [1]. This tool provides detailed micro-architecture level statistics which are also used for power and SER estimation. The input traces are composed of *simpointed* subtraces [38], each of 100M instruction length. This tool is integrated with the Detailed Power Model (DPM) tool, used as the reference in [50], that has been validated against actual measurements on IBM POWER systems and circuit simulations to a high degree of accuracy ($<$ 5% absolute error as reported in [26]). DPM has been instrumental in modeling power consumed by the generations of processors from POWER7 and beyond. For simulating the simple core-based processor, we use an architectural simulator validated against WireSpeed [27] and Blue Gene/Q cores [46], which also outputs statistics to enable power and soft error modeling. The application-driven power model was validated against actual measurement (to within 5% error) [46] on the IBM Blue Gene/Q Sequoia supercomputer [25], and adapted to our architectural parameters.

In order to scale the single core simulation results to a multi-core system without the large simulation time overheads associated with most multi-core simulators, we use an in-house high-level analytical model for estimating multi-core contention using performance metrics collected from single-core simulation runs. The model was validated against hardware measurements on the POWER platform for both synthetic stressmarks and SPEC CPU2006 workloads to within 10% error.

The *EinSER* tool is used to estimate the soft error rate (SER) of the target processor by using information across multiple layers of abstraction ranging from the logic to the micro-architecture and the application level. The tool has three main modules: (a) HDL Extraction and Analysis [6], (b) Microarchitectural Derating (MD) estimation [40] and (c) Application Derating Estimation [21]. The first module derives logic-level derating factors based on different latch types and latch counts from the design database. By utilizing this logic-level information and the microarchitecture-level residency statistics for different applications, obtained from the performance simulator, the second module computes the microarchitectural derating factor, defined as the ratio of derated bits to the total bits in the system. In addition, the application itself adds another level of derating, since not all incorrect architectural states will necessarily affect the final program output. The third module is used to calculate this Application-level Derating factor (AD) [11, 21, 40], by means of statistical fault injection during program execution. We model the voltage dependence of SER based on results reported in [37]. The SER model was validated against accelerated proton testing of the Blue Gene/Q processor [10], and its latch classification module against RTL-level error injection [20].

The impact of aging-based hard errors on processor reliability, namely, EM, TDDB and NBTI, is determined analytically as described in Section 2. Our framework inputs grid-level maps of the power and temperature distribution and outputs grid-level FIT rates for both reference processors, for each of the aging phenomena. In order to obtain the thermal map across the processor, we use HotSpot-6.0 [24], with thermal conductivities and the architectural parameters tuned to match the reference processors and validated against real POWER systems [52].

5. RESULTS

Based on the methodology described in Section 3, we detail our experimental evaluations and results. Characterizations were done on kernels from the PERFECT application suite [2], which is a benchmark suite used to assess next-generation processor architectures across the computational spectrum. Our evaluation techniques comprised of using the proposed BRAVO framework to determine the optimal V_{dd} by profiling each application on both *SIMPLE* and *COMPLEX* processors, under various conditions described in this section.

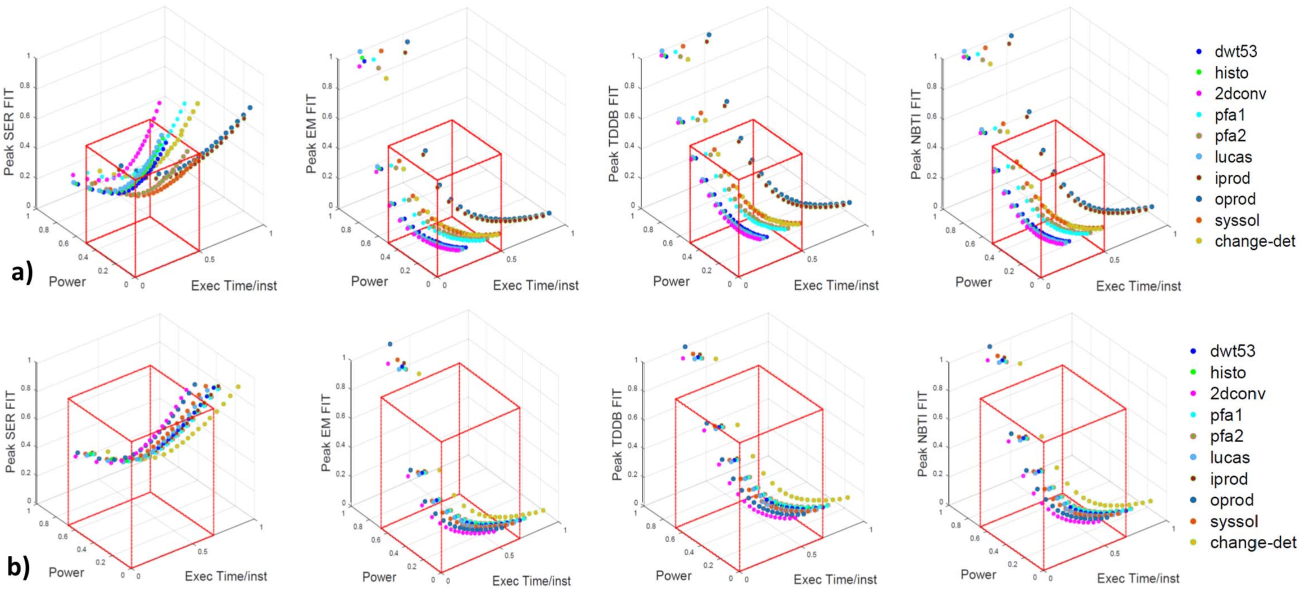
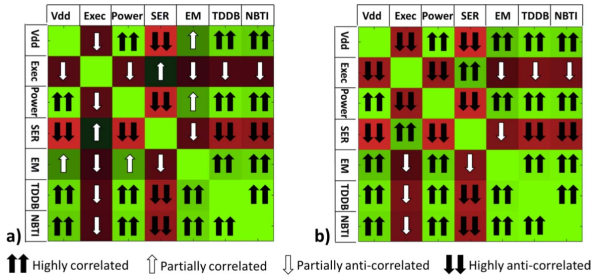


Figure 5: Comparison of FIT rates due to SER, EM, TDDB and NBTI with power and performance for different applications at different processor V_{dd}s for a) *COMPLEX* and b) *SIMPLE*



a)

	Vdd	Exec	Power	SER	EM	TDDB	NBTI
Vdd		⇩	⇈	⇊	⇧	⇈	⇈
Exec	⇩		⇩	⇧	⇩	⇩	⇩
Power	⇈	⇩		⇊	⇧	⇈	⇈
SER	⇊	⇧	⇊		⇩	⇊	⇊
EM	⇧	⇩	⇧	⇩		⇈	⇈
TDDB	⇈	⇩	⇈	⇊	⇈		⇈
NBTI	⇈	⇩	⇈	⇊	⇈	⇈	

b)

	Vdd	Exec	Power	SER	EM	TDDB	NBTI
Vdd		⇊	⇈	⇊	⇈	⇈	⇈
Exec	⇊		⇊	⇈	⇩	⇩	⇩
Power	⇈	⇊		⇊	⇈	⇈	⇈
SER	⇊	⇈	⇊		⇩	⇊	⇊
EM	⇈	⇩	⇈	⇩		⇈	⇈
TDDB	⇈	⇩	⇈	⇊	⇈		⇈
NBTI	⇈	⇩	⇈	⇊	⇈	⇈	

⇈ Highly correlated ⇧ Partially correlated ⇩ Partially anti-correlated ⇊ Highly anti-correlated

Figure 4: Pairwise comparison of input voltage, performance, power and reliability metrics in terms of relative trends and correlation coefficients for a) *COMPLEX* and b) *SIMPLE*. A positive correlation is indicated by a ⇑ in a green box and a negative one by a ⇓ in a red box.

In addition to static assignment of optimal V_{dd} for each application, it is also possible to make this determination at runtime, by varying the voltage across different program phases. Present-day DVFS techniques do not take into account the combined effects of hard and soft errors on processor reliability. These aspects are discussed in further detail in Section 6.3.

5.1 Pairwise comparison of performance, power and reliability metrics

In several application domains, the relative importance attached to a particular design dimension such as performance, power, or reliability might be higher than the others. For instance, it might be essential to meet real-time deadlines, even at the cost of power and reliability. Conversely, in some other cases, the adverse effects of soft and hard errors are far more significant than the need for mitigating power or performance overheads. Figure 4 shows a pairwise comparison of the relative trends and the correlation between the supply voltage, execution time, power, SER, EM, TDDB and NBTI FIT rates, averaged across all applications from the PERFECT suite. The green squares indicate that the two metrics have the same direction of variation with voltage and frequency, while the red squares indicate that the trends are in the opposite direction. We observe a high degree of correlation amongst the different hard error components, while SER varies in the opposite direction. Further, SER and execution time are correlated on account of the SER increasing with increasing residency. In some cases, we observe differing degrees of correlation or anti-correlation between and *COMPLEX* and *SIMPLE*. For instance, the correlation between SER and execution time is observed to be lower in *COMPLEX* than in *SIMPLE*. This is due to the greater degree of instruction-level parallelism (ILP) that is possible in *COMPLEX* due to its out-of-order nature.

5.2 Analysis of individual sources of processor vulnerability

Figure 5a) and b) show the variation of the peak FIT rates due to SER, EM, TDDB and NBTI on *COMPLEX* and *SIMPLE* respectively. Results are normalized to the corresponding worst case value on each axes, for the various applications running at different V_{dd}s, with respect to the corresponding performance (in terms of execution time per instruction) and power. By fixing a user-defined threshold across each coordinate axis, indicated by the red lines, it is possible to deem each of the data points to be *acceptable* in terms of performance, power and reliability for each of the plots. Due to the higher power consumption, performance and temperatures attained by *COMPLEX*,in contrast to *SIMPLE*, we consider tighter constraints to be placed on *COMPLEX*, resulting in a smaller acceptable region covered by the red lines. We observe that while the aging effects increase with increasing V_{dd}, the overall SER trend is in the opposite direction. This is because, increasing the voltage increases the margin between the existing charge and the critical charge (Q_{crit}), which reduces the SER probability.

This figure also enables us to determine the configuration with the best energy efficiency or minimum Energy-Delay Product (EDP). However, this operating point does not take into account the joint effects of soft errors and hard errors. Hence, we use the methodology described in Section 3, to determine the Balanced Reliability Metric.

5.3 Determining the optimal reliability-aware operating point

Figure 6 shows results of reliability analysis based on the BRM. All values have been normalized to the worst-case. The non-monotonicity of the curves clearly show that there is an optimal operating point which is different for different applications, which is determined by the competing trends between soft and hard error rates. Note that while the overall reliability is maximum at this point, it might not necessarily be the most energy-efficient.

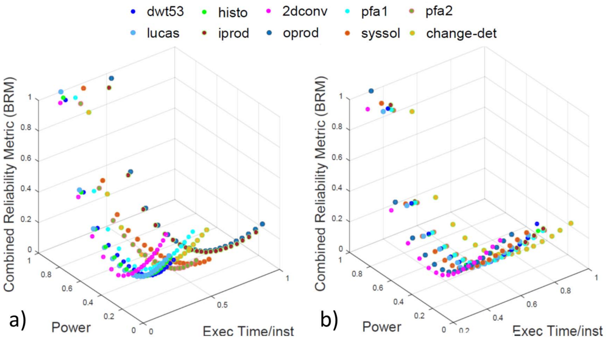


Figure 6: Comparison of Balanced Reliability Metric (BRM) with power and performance at different processor supply voltages for a) *COMPLEX* and b) *SIMPLE*. We can observe that each application has an optimal operating point in this comparison, in contrast to prior comparisons of individual reliability metrics.

5.4 Comparison of individual reliability components with BRM

Figure 7a) shows the variation of each individual reliability component for *COMPLEX*, namely SER, EM, TDDB, NBTI, and that of the combined BRM metric, with voltage for the *pfa1* benchmark from the PERFECT suite, with V_{MAX} being the maximum attainable V_{dd}. We observe that BRM closely follows the SER curve until the reliability-aware optimal point, beyond which the aging effects dominate. The sensitivity of BRM with respect to each metric's variation in voltage is shown in Figure 7b). These figures thus establish the utility of a statistical metric such as BRM in determining a) the optimal V_{dd} (empirically obtained at the cross-over point) across opposing effects of hard and soft errors with varying voltages b) the reliability metric that is the dominant component of the BRM at each operating V_{dd} .

It is also possible to characterize processors based on the ratio of hard and soft errors to the total errors. For instance, in most processors, hard errors may dominate over soft errors in the estimation of the total FIT rate. The ability to specify a target hard-to-soft error ratio would be extremely useful for designers, particularly in the implementation of hardware mechanisms for error mitigation. To characterize the impact of this ratio on overall processor reliability, we demonstrate experimental results that consider different ratios of hard and soft errors in the processor. Figure 8 shows the mode (most frequently appearing) values of the optimal V_{dd}s across all evaluated applications for both processor types for different proportions of hard errors to the total. A ratio of 0 means that the entire processor reliability comprises of exclusively opti-

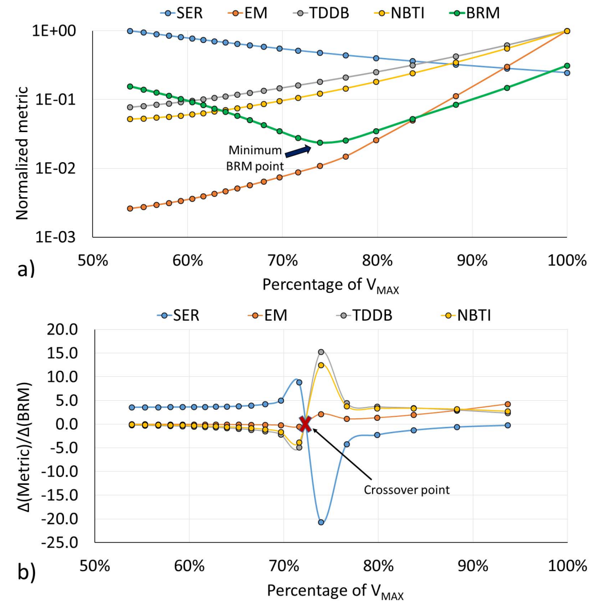


Figure 7: a) Variation of each reliability metric along with the BRM with supply voltage for *pfa1*. b) Ratio of variation of each metric to variation of BRM ($\frac{\Delta(Metric)}{\Delta(BRM)}$), with voltage. The supply voltage is expressed as a percentage of V_{MAX}, the maximum attainable value for the processor. We observe which metric is the dominant component at each operating voltage, with the optimal V_{dd} occurring at 74% of V_{MAX}

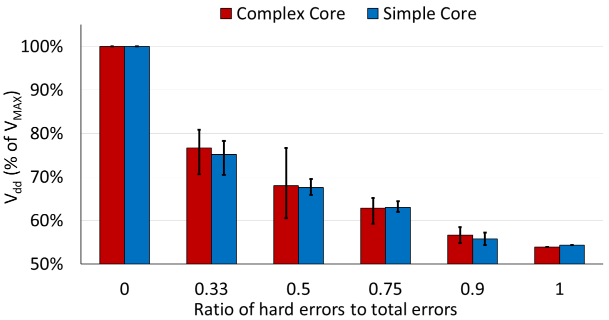


Figure 8: Variation of optimal V_{dd} as a fraction of the maximum attainable voltage (V_{MAX}) when the fraction of hard errors is varied in the processors for both *COMPLEX* and *SIMPLE*. The bars denote the most frequently appearing value (i.e the mode) of optimal voltage across applications for each hard error ratio, while the black lines show the corresponding minimum and maximum values. While the mode value of the optimal V_{dd} is similar across both processors, *COMPLEX* shows much greater variation in optimal V_{dd} across applications

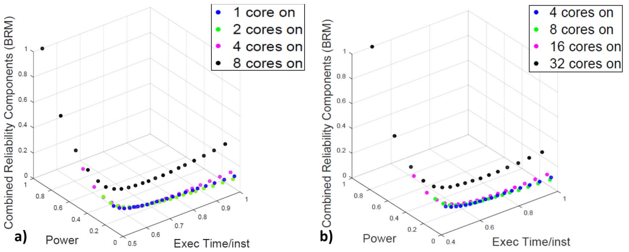


Figure 9: Variation of optimal V_{dd} in *histo*, when a) *COMPLEX* is operated with 1, 2, 4 and 8 cores turned on, and b) *SIMPLE* is operated with 4, 8, 16 and 32 cores turned on. In both cases, the optimal V_{dd} tends towards the minimum operating value (V_{MIN}) when majority of the cores are turned off

mizing soft errors, while a ratio of 1 implies that only hard errors are considered. The minimum and maximum values attained by the frequency and voltage for these applications are also shown. It is evident that increasing the ratio causes a drop in optimal voltage and frequency. Although the mode V_{dd} values are similar across both processors, the deviation across applications between maximum and minimum values of optimal V_{dd} is observed to be much larger in *COMPLEX*, due to its greater degree of application dependence.

5.5 Impact of power gating

We also run experiments to examine the effect of power gating cores and running fewer threads on the processor. This would clearly benefit overall reliability, since hard errors decrease due to reduced power density, and SER also decreases due to fewer operating cores resulting in fewer vulnerable bits. Figures 9a) and b) show this effect when we run copies of *histo* on 1, 2, 4 and 8 cores on *COMPLEX* and 4, 8, 16 and 32 cores on *SIMPLE* respectively. We observe that the optimal V_{dd} drops when fewer cores are running. In the case when fewest cores are on, the BRM increases monotonically with increase in V_{dd}, resulting in the optimal V_{dd} settling at the minimum attainable value (V_{MIN}). The optimal V_{dd} increases when more cores are turned on. This is because the SER component drops linearly with increased power gating of cores, while a more gradual drop is seen in hard errors due to their dependence on processor temperature. Consequently, hard errors dominate in case of fewer operating cores, reducing the optimal V_{dd}.

5.6 Impact of SMT

In addition to single-threaded results, we also estimate the BRM in case of simultaneous multithreading (SMT). Both complex and simple cores are capable of running up to to 4-way SMT. Figure 10 demonstrates the optimal operating voltage for 1, 2 and 4-way SMT for both processor platforms. Both soft and hard errors increase with increasing SMT. The increased resource contention causes the overall residency and utilization to increase, resulting in higher SER. In addition, the increased per-core activity causes a rise in overall temperature which in turn, exacerbates aging effects. Hence, the change in optimal operating point depends on which metric (hard or soft errors) shows a more drastic increase with SMT. For instance, in *change-det* running on *COMPLEX*, the BRM increases sharply on account of increased residency due to SMT, while the relative throughput and power increase marginally. This results in the optimal operating point increasing even further. The reverse phenomenon is observed in applications such as *iprod*, while in others such as *dwt53*, the optimal point remains unchanged.

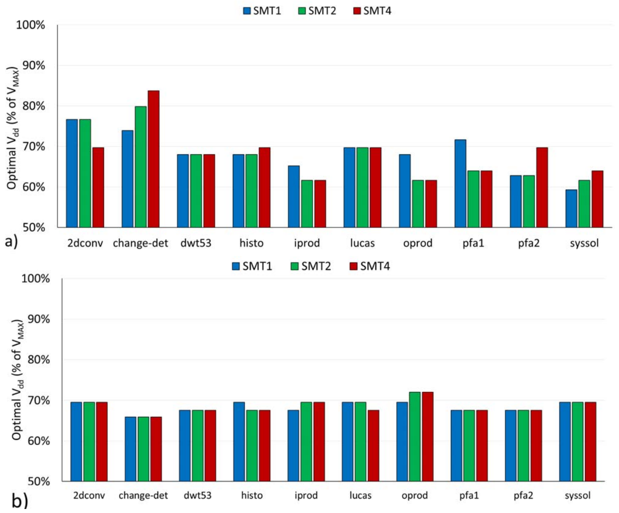


Figure 10: Comparison of optimal V_{dd} with varying SMT for a) *COMPLEX* and b) *SIMPLE*. For applications which show increasing SER with SMT, the optimal V_{dd} goes up, while for applications where temperature and consequently hard errors dominate with increasing SMT, the optimal V_{dd} drops

Table 1: Optimal operating points from the point of view of energy efficiency and reliability for both *COMPLEX* and *SIMPLE*. The EDP column denotes the operating point with minimum EDP, while the BRM column denotes that with minimum BRM

Application	Optimal Voltage (Fraction of V_{MAX})			
	EDP *COMPLEX*	BRM *COMPLEX*	EDP *SIMPLE*	BRM *SIMPLE*
2dconv	0.65	0.77	0.68	0.70
change-det	0.59	0.74	0.64	0.66
dwt53	0.65	0.68	0.68	0.68
histo	0.65	0.68	0.68	0.70
iprod	0.65	0.65	0.68	0.68
lucas	0.65	0.70	0.68	0.70
oprod	0.65	0.68	0.68	0.70
pfa1	0.65	0.74	0.68	0.68
pfa2	0.59	0.63	0.68	0.68
syssol	0.65	0.59	0.68	0.70

5.7 Comparison of reliability and EDP-aware optimal V$_{dd}$

In our previous results, we stated that the reliability-aware optimal may not be the best choice with respect to the EDP, even though its reliability may be significantly higher due to its lower BRM. Table 1 illustrates this point, by showing the EDP and reliability-aware optimal operating voltages for each of the applications. In general, the increase in SER with decreasing voltage is greater than the corresponding decrease in hard error rate. Therefore, the reliability-aware optimal voltage becomes greater than the EDP-aware optimal in these cases. Conversely, in case of *syssol*, the low utilization of LSQ structures due to fewer memory accesses results in the absolute value of SER being much lower. This in turn, leads to lower optimal V_{dd}. In case of *SIMPLE*, the contribution to the overall power of the interconnects and other uncore components is far greater at lower voltages, since no voltage scaling is applied to the uncore. Thus, the temperature and consequently, the hard error rate increases at lower V_{dd}s, causing the reliability-aware optimal voltage to rise. In addition, the inter-application variation of the reliabilty-aware optimal is also less for *SIMPLE*. On the other hand, the variation of the optimal V_{dd} across applications for *COMPLEX* is much more pronounced. This is because the ratio of core power to overall power is much higher in *COMPLEX* than in *SIMPLE*, resulting in voltage-dependent core characteristics playing a larger role in determining the optimal voltage.

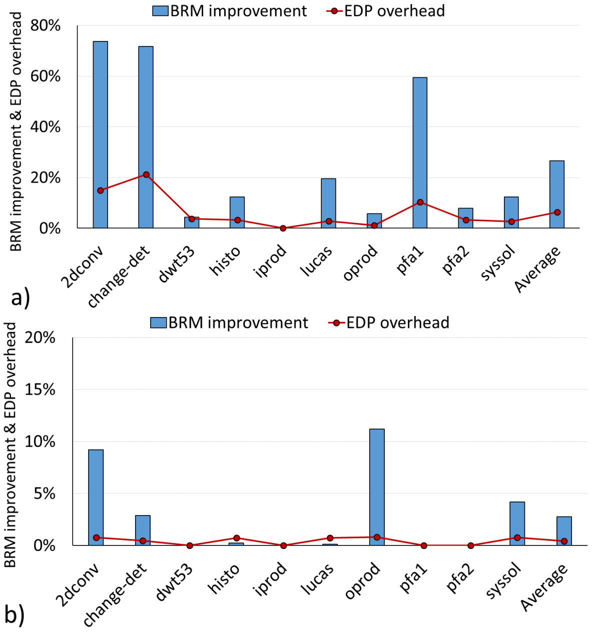


Figure 11: Comparison of the reliability and energy-efficiency tradeoffs for a) *COMPLEX* and b) *SIMPLE*. The blue bars denote the improvement in reliability when operating at the BRM-aware optimal point instead of the optimal EDP point. The red line denotes the corresponding overhead in energy efficiency that is incurred on account of operation at this new optimal V_{dd}

5.8 Tradeoffs in reliability and energy-efficiency

Figure 11 shows the reliability improvement in terms of BRM when each processor is operated at the optimal BRM V_{dd} for each application as opposed to the optimal EDP point. It is evident that this would result in a small drop in energy efficiency (denoted by the red line) due to the increased EDP. We observe an average BRM improvement of 27% for a mere 6% EDP overhead on *COMPLEX*, with a peak improvement of 79%. In case of *SIMPLE*, improvements are more modest since the BRM-aware optimal V_{dd} deviates only marginally from the corresponding EDP-aware optimal, yielding only 3% BRM improvement at negligible EDP overhead ($< 0.5\%$).

6. CASE STUDIES

The BRAVO framework makes it possible to architect a processor for an optimal V_{dd} to attain desired reliability and energy efficiency targets. This, in turn, enables designers to implement optimizations across the system stack, ranging from tuning transistor parameters to designing processor and software-level reliability and power management techniques. We subsequently demonstrate use-cases for the High Performance Computing (HPC) and the embedded systems environments, where we quantify the performance impacts of achieving better reliability using the BRAVO methodology.

6.1 Use Case 1: Applying BRAVO to High Performance Computing (HPC) systems

In the HPC world, where systems consist of millions of cores, long-running tasks rely on checkpoint-restart (CR) [12, 34] to provide robustness against detected failures. While failures induced by both hard and soft errors cost execution time, only hard errors incur the need to replace failed hardware. Hence, it may be desirable to reduce hard errors at the expense of soft errors by decreasing voltage and frequency.

In this example, we show that BRAVO can be used to tune a system for longer lifetime, with optimal performance impact. For simplicity, we assume that computing components are the only source of failures during application runtime, although our methodology can be extended to cover other components such as memory and network components as well. Suppose an application spends 60% of its time in computation on cores, 20% in network communication, 9% in checkpoint cost, 9% in loss-of-work cost due to restart, and 2% restart cost (i.e., 9%+9%+2%=20% CR costs). In [19], the authors show that CR costs vary little between incremental and full CR, and vary between 5% to 35% for real HPC applications; hence, a mean time of 20% is a viable assumption. Only the compute time is affected by the frequency change in cores, and CR time is parameterized by checkpoint interval and Mean-Time-Between-Failures (MTBF). Using the BRAVO methodology, we can explore the relationship between CR costs and frequency change to arrive at the optimal value of the metric of interest.

Figure 12 shows the sensitivity of hard error rate and system execution time with varying voltage/frequencies for the HPC system consisting of complex out-of-order cores, and with the CR cost at maximum frequency (F_{MAX}) varying at 0% and 20% to illustrate our methodology. The left y-axis shows the relative execution time for the line graph, and the right y-axis shows the relative hard error rate for the bar graph. The x-axis shows increasing frequencies from left to right. The F_{MAX} CR cost varies depending on the network and storage configurations. In [12], CR costs are calculated using failure data collected from USENIX Computer Failure Data Repository (CFDR) for a Blue Gene/P machine [56]. In this system, when frequency is decreased, the system loses performance due to the lowered core frequency, but gains by reducing CR costs. The CR costs improve because the hard error rate decreases faster than the soft error rate increases when reducing voltage from V_{MAX}, resulting in overall processor reliability improvement. In the example below, we show detailed calculations for the 20% CR cost case, which can also be applied to systems with different CR costs.

Figure 12 shows the optimal operating point for the above case, where the reduction in BRM, averaged across all PERFECT applications, leads to 2.35× MTBF improvement over

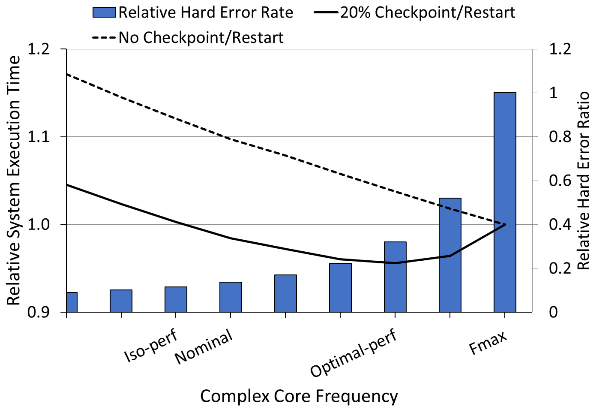


Figure 12: Variation of execution time and relative hard error rate with frequency (varied by adjusting V_{dd}), in the presence and absence of Checkpoint-Restart (CR) overheads. The peak frequency F_{MAX} is the core frequency at V_{MAX} while the *Nominal* point denotes the nominal frequency of execution. Since the hard error rate decreases faster than the soft error rate for lower frequency, the CR overheads also reduce correspondingly. In the presence of 20% CR overhead, execution time at *Iso-perf* frequency is equal to that at F_{MAX}, while it is minimum at *Optimal-perf*. For both with and without checkpoint-restart cases, the execution time values are normalized to corresponding execution times at F_{MAX} for each scheme.

the peak frequency point. In [13, 28], the optimal checkpoint interval is calculated as $\sqrt{2 * MTBF * CheckpointLatency}$. Therefore, the relative change of the optimal checkpoint interval from F_{MAX} to *Optimal-perf* is $\sqrt{2.35}$, because other unchanged terms are canceled out. The checkpoint cost thus reduces by $\sqrt{2.35}\times$, due to less frequent checkpoints. The loss-of-work cost which is computed as interval/MTBF, also reduces by $\sqrt{2.35}\times$. The restart cost, which involves loading the checkpoint across the network, reduces by 2.35×. The relative speedup on account of switching F_{MAX} to *Optimal-perf*, is computed as: *60% compute * 1.05 + 20% network + 6% checkpoint* * $\sqrt{1/2.35}$ + *12% loss-of-work* * $\sqrt{1/2.35}$ + *2% restart * 1/2.35 = 0.956*, which is 4.4% faster.

From Figure 12, we also find the iso-performance frequency, where the system achieves 8.7× better lifetime reliability (and 2.1× power savings) at no performance loss than the F_{MAX} system. Although the finding in our example may not be universal, we show that BRAVO can optimize for lifetime while balancing between CR costs and core frequency at the HPC system level.

In summary, operating at a reliability-unaware voltage can significantly compromise the system reliability. This in turn, can result in additional overheads in performance and power, on account of additional techniques to maintain this reliability, such as checkpoint-restart (CR).

6.2 Use Case 2: Reliability-aware embedded system design

Low-end embedded system SoCs, unlike HPC systems are rarely designed for life-cycles beyond 3-5 years. In such a scenario, the impact of hard errors due to aging phenomena is reduced significantly. Due to the stringent power and energy constraints that are placed on these systems, operation at low, or near-threshold voltages might seem an attractive proposition. However, the increased likelihood of soft errors makes this challenging. Due to their high performance and power overheads, techniques such as Checkpoint-Restart (CR) that are applied in HPC systems for reducing SERs may not be feasible for embedded systems with tight power and throughput constraints, leading to techniques such as selective duplication and re-execution of particular hardware modules [31].

Figure 13 demonstrates the comparison between the SER improvement obtained by a) Selective duplication of the micro-architecture components most vulnerable to soft errors, while operating at near-threshold voltage and b) Operating at a higher voltage as dictated by BRAVO, that consumes the same energy as a processor with selective duplication, as in a). We observe that the reduction in SER due to BRAVO is 14% higher than that due to selective duplication. Note that this does not consider the energy overheads due to re-execution when an error is detected, or the area and costs incurred in case of selective duplication, which would result in the benefits due to BRAVO being even more pronounced.

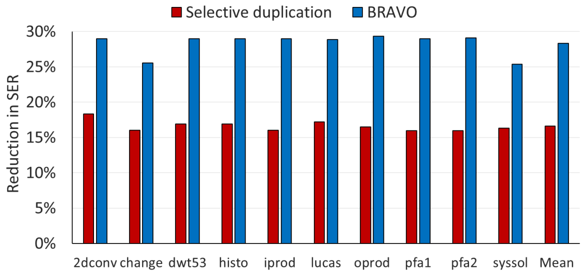


Figure 13: Comparison of SER reduction due to selective duplication of the most vulnerable component and voltage optimization using BRAVO within the same energy budget. We observe the BRAVO-based technique yields 14% lower SER than selective duplication.

6.3 Discussion: Future research directions

It has been demonstrated that BRAVO can be utilized for making key design decisions related to the operating voltage and frequency of next-generation processors. However, it can also be used for finer-grained voltage optimizations at runtime, depending on the variation across application phases.

Dynamic voltage-frequency scaling (DVFS) techniques that trade off power and performance [8, 23]. However, with reliability considerations assuming increased importance with each technology node, it has become essential for these schemes to also take reliability into account. This gives rise to several new challenges such as:

- Need for on-chip sensors or proxies to measure soft and hard error components at runtime. This is integral to building an end-to-end validation framework for the presented methodology. While in this work, each step in the methodology was individually validated against measurements on actual silicon, there remains the challenge of integrating these sensor outputs into our overall infrastructure.

- Techniques for effectively predicting these reliability components depending on application phase behavior.

- Dynamic management algorithms that can intelligently combine several of these reliability components into one common metric to ease the tradeoff between power, performance and reliability.
- Extending reliability-aware design to other micro-architecture level explorations. In addition to the optimizing voltage, one could also extend the BRAVO methodology to analyzing various other aspects of the processor micro-architecture, such as the optimal pipeline depth, issue width, cache configuration etc. Determining these micro-architectural parameters, along with the operating voltage, while taking reliability into account, is a challenge that would definitely be of interest to the research community.

It is hoped that addressing these challenges would spawn new research directions towards building reliability-aware computing systems and applications.

7. RELATED WORK

7.1 Reliability modeling and analysis

Several prior works have demonstrated detailed estimation methodologies for each individual phenomenon affecting reliability. Our work carries out a joint analysis of all aspects related to reliability, including radiation-induced soft errors, and aging-induced hard errors. In [35, 39], the authors compute the *Architectural Vulnerability Factor* (AVF) for each processor unit and then add unit-level Soft Error Rates (SER) to get an overall value. Faultsim [36] is a configurable memory-reliability simulation tool for 2D- and 3D-stacked memories that accelerates evaluation of schemes such as Chipkill. In [47], the authors estimate the overheads of recovery mechanisms that cater to (single/multi-bit) soft errors in the processor. In [9], the authors propose a range of cross-layer techniques to drastically reduce the overall SER. However, they primarily focus on examining techniques exclusively related to soft error mitigation and not at hard errors or voltage optimization schemes. [32] present a range of energy-efficient protection techniques to mitigate and recover from variation-induced timing errors at runtime and soft errors, unlike our work which jointly examines the effects of both hard and soft errors with a view to determine the reliability-aware optimal operating voltage.

7.2 Multi-dimensional optimization in processors

In [51], Yetim *et al.* propose heterogeneous architectures with tradeoffs between reliable, high power reliable and unreliable, low power cores. Zhang *et al.* [54] demonstrate techniques to carry out reliability-aware power management, while in [55], the authors jointly examine reliability and energy efficiency by means of DVFS techniques. Foutris *et al* [18] focus on the performance impact of hard faults in processors through fault injection techniques. In [49, 48], the authors carry out multi-dimensional optimizations of power, performance and soft error reliability for different application workflows. However, unlike our work, these papers simply consider the reliability metric to depend either on soft or hard errors alone, without examining a composite reliability metric, or determine a reliability-aware optimal V_{dd}. Lee *et al.* propose DVFS and power gating techniques for power- and thermally-constrained processors, while examining the effects of die-level process variations in [30]. However, this work does not focus on soft errors or errors due to aging effects, or the interaction between them. In [5], the authors propose a cache design capable of efficient low-voltage operation, however, this analysis does not extend to the entire processor as is the case in our work.

In [7], the authors propose joint circuit-architecture optimizations to determine the configurations that yield near-optimal energy-efficiency for a large region of the energy-performance space. In [29], Lee explores various statistical techniques ranging from regression models to artificial neural networks that can be used for optimizing power and performance across a host of microarchitectural characteristics. However, unlike this paper, neither work focuses on reliability as a parameter in the cross-dimensional optimizations. Instead, they focus on microarchitectural design knobs such as pipeline depth, issue width, cache and register file size etc, while we primarily focus on determining the optimal V_{dd} for a given architecture. In [22], the authors carry out a reliability-aware thermal optimization of processors, focusing on electromigration effects. In [43, 44], the authors examine the tradeoffs between performance, power, temperature and lifetime reliability. However, in all these works the study is restricted to lifetime reliability, without looking at SER effects.

8. CONCLUSION

In this paper, we proposed *BRAVO*, a feasible and practical methodology for determining the optimal operating voltage in conjoint consideration of performance, power and reliability, during the pre-silicon definition of the processor micro-architecture. We identified the main factors that influence processor reliability and provided estimation techniques for each factor for two architectural multicore platforms comprising of complex out-of-order and simple in-order cores. We proposed the *Balanced Reliability Metric (BRM)*, a composite metric that can be used to evaluate processor reliability as a whole. Our techniques enable us to achieve up to 79% improvement in overall reliability, in terms of the BRM, for only 6% reduction in overall energy efficiency with respect to the reliability-unaware EDP point. Finally, we discuss real-life use-cases from industry where this methodology can be used along with existing techniques for reliability-aware optimization of throughput and power in the target system.

Acknowledgment

This work is sponsored by Defense Advanced Research Projects Agency, Microsystems Technology Office (MTO), under contract no. HR0011-13-C-0022. The views expressed are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

9. REFERENCES

[1] "SDK for Linux on POWER" https://www-304.ibm.com/webapp/set2/sas/f/lopdiags/sdkdownload.html.

[2] "Test, Assessment and Verification Effort (TAV) for the DARPA PERFECT Program -" http://hpc.pnl.gov/PERFECT/.

[3] H. Abdi, "Partial least squares (PLS) regression"

[4] H. Abdi and L. Williams, "Principal component analysis" in *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[5] A. Alameldeen *et al.*, "Adaptive cache design to enable reliable low-voltage operation" in *IEEE Trans. Computers*, vol. 60, no. 1, pp. 50–63, 2011.

[6] E. Arbel, S. Koyfman, P. Kudva, and S. Moran, "Automated detection and verification of parity-protected memory elements" ICCAD 2014.

[7] O. Azizi *et al.*, "Energy-performance tradeoffs in processor architecture and circuit design: a marginal cost analysis" in Proceedings of *ISCA*, pp. 26–36, 2010.

[8] R. Bergamaschi *et al.*, "Exploring power management in multi-core systems" in Proceedings of *ASP-DAC*, pp. 708–13, 2008.

[9] E. Cheng *et al.*, "Clear: Cross-layer exploration for architecting resilience - combining hardware and software techniques to tolerate soft errors in processor cores" DAC 2016.

[10] C.-Y. Cher *et al.*, "Soft error resiliency characterization and improvement on IBM BlueGene/Q processor using accelerated proton irradiation" in Proceedings of *ITC*, pp. 1–6, 2014.

[11] C.-Y. Cher *et al.*, "Understanding soft error resiliency of blue gene/q compute chip through hardware proton irradiation and software fault injection" in Proceedings of *High Performance Computing, Networking, Storage and Analysis, SC14*, pp. 587–96, 2014.

[12] C. Costa *et al.*, "A system software approach to proactive memory-error avoidance" in Proceedings of *High Performance Computing, Networking, Storage and Analysis*, pp. 707–18, 2014.

[13] J. Daly, "A model for predicting the optimum checkpoint interval for restart dumps" in Proceedings of *International Conference on Computational Science ICCS*, pp. 3–12, 2003.

[14] F. d'Heurle, "Electromigration and failure in electronics: An introduction" 1971.

[15] R. Dreslinski *et al.*, "Near-threshold computing: Reclaiming Moore's Law through energy efficient integrated circuits" in *Proc. of the IEEE*, vol. 98, no. 2, pp. 253–266, 2010.

[16] R. Dreslinski *et al.*, "Centip3De: A Many-core Prototype Exploring 3D Integration and Near-threshold Computing" in *Commun. ACM*, vol. 56, no. 11, pp. 97–104, 2013.

[17] D. Ernst *et al.*, "Razor: a low-power pipeline based on circuit-level timing speculation" in Proceedings of *MICRO*, pp. 7–18, 2003.

[18] N. Foutris, D. Gizopoulos, J. Kalamatianos, and V. Sridharan, "Assessing the impact of hard faults in performance components of modern microprocessors" in Proceedings of *ICCD*, pp. 207–14, 2013.

[19] R. Gioiosa, J. C. Sancho, S. Jiang, and F. Petrini, "Transparent, incremental checkpointing at kernel level: a foundation for fault tolerance for parallel computers" in Proceedings of *Conference on High Performance Networking and Computing*, p. 9, 2005.

[20] M. Gschwind, V. Salapura, C. Trammell, and S. McKee, "Softbeam: Precise tracking of transient faults and vulnerability analysis at processor design time" in Proceedings of *ICCD*, pp. 404–10, 2011.

[21] M. Gupta, J. Rivers, L. Wang, and P. Bose, "Cross-layer system resilience at affordable power" IRPS 2014.

[22] V. Hanumaiah and S. Vrudhula, "Reliability-aware thermal management for hard real-time applications on multi-core processors" in Proceedings of *DATE*, pp. 1–6, 2011.

[23] V. Hanumaiah and S. Vrudhula, "Energy-efficient operation of multicore processors by DVFS, task migration, and active cooling" in *IEEE Trans. Computers*, vol. 63, no. 2, pp. 349–60, 2014.

[24] W. Huang *et al.*, "Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model" in *IEEE Trans. Computers*, vol. 57, no. 9, pp. 1277–88, 2008.

[25] IBM Blue Gene team, "Design of the IBM Blue Gene/Q Compute chip" in *IBM J. of Res. and Dev.*, vol. 57, no. 1/2, pp. 1:1–13, 2013.

[26] H. Jacobson *et al.*, "Abstraction and microarchitecture scaling in early-stage power modeling" HPCA 2011.

[27] C. Johnson *et al.*, "A wire-speed POWER^{TM} processor: 2.3GHz 45nm SOI with 16 cores and 64 threads" ISSCC 2010.

[28] W. Jones, J. Daly, and N. DeBardeleben, "Impact of sub-optimal checkpoint intervals on application efficiency in computational clusters" in Proceedings of *HPDC*, pp. 276–9, 2010.

[29] B. Lee, "Applied statistical inference for system design and management" in Proceedings of *ICCD*, pp. 188–191, 2015.

[30] J. Lee and N. S. Kim, "Analyzing Potential Throughput Improvement of Power- and Thermal-Constrained Multicore Processors by Exploiting DVFS and PCPG" in *IEEE TVLSI*, vol. 20, no. 2, pp. 225–235, 2012.

[31] M. Maniatakos, P. Kudva, B. Fleischer, and Y. Makris, "Low-cost concurrent error detection for floating-point unit (FPU) controllers" in *IEEE Trans. Computers*, vol. 62, no. 7, pp. 1376–88, 2013.

[32] T. Miller, N. Surapaneni, and R. Teodorescu, "Flexible error protection for energy efficient reliable architectures" SBAC-PAD 2010.

[33] S. Mitra *et al.*, "The resilience wall: Cross-layer solution strategies" in Proceedings of *VLSI-DAT*, pp. 1–11, 2014.

[34] A. Moody *et al.*, "Design, modeling, and evaluation of a scalable multi-level checkpointing system" in Proceedings of *High Performance Computing, Networking, Storage and Analysis*, pp. 1–11, 2010.

[35] S. Mukherjee *et al.*, "Architectural vulnerability factors for a high-performance microprocessor" MICRO 2003.

[36] P. Nair, D. Roberts, and M. Qureshi, "Faultsim: A fast, configurable memory-reliability simulator for conventional and 3d-stacked systems" in *ACM TACO*, vol. 12, no. 4, pp. 44:1–24, 2015.

[37] P. Oldiges *et al.*, "SOI finFET soft error upset susceptibility and analysis" in Proceedings of *IRPS*, p. 4, 2015.

[38] E. Perelman, G. Hamerly, and B. Calder, "Picking statistically valid and early simulation points" PACT 2003.

[39] S. Raasch *et al.*, "A fast and accurate analytical technique to compute the AVF of sequential bits in a processor" MICRO 2015.

[40] J. Rivers *et al.*, "Phaser: Phased methodology for modeling the system-level effects of soft errors" in *IBM J. of Res. and Dev.*, vol. 52, no. 3, pp. 293–306, 2008.

[41] N. Seifert *et al.*, "Soft error susceptibilities of 22 nm tri-gate devices" in *IEEE Transactions on Nuclear Science*, vol. 59, no. 6, pp. 2666–73, 2012.

[42] J. Shin *et al.*, "A framework for architecture-level lifetime reliability modeling" in Proceedings of *DSN*, pp. 534–543, 2007.

[43] W. Song, S. Mukhopadhyay, and S. Yalamanchili, "Architectural reliability: Lifetime reliability characterization and management of many-core processors" in *IEEE CAL*, vol. 14, no. 2, pp. 103–6, 2015.

[44] W. Song, S. Mukhopadhyay, and S. Yalamanchili, "Managing performance-reliability tradeoffs in multicore processors" in Proceedings of *IRPS*, pp. 3C.1–7, 2015.

[45] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The case for lifetime reliability-aware microprocessors" in Proceedings of *ISCA*, pp. 276–287, 2004.

[46] K. Sugavanam *et al.*, "Design for low power and power management in IBM Blue Gene/Q" in *IBM J. of Res. and Dev.*, vol. 57, no. 1/2, pp. 3:1–11, 2013.

[47] L. Szafaryn, B. Meyer, and K. Skadron, "Evaluating overheads of multibit soft-error protection in the processor core" in *IEEE Micro*, vol. 33, no. 4, pp. 56–65, 2013.

[48] L. Wang *et al.*, "Power-efficient embedded processing with resilience and real-time constraints." ISLPED 2015.

[49] L. Wang *et al.*, "Resilience and real-time constrained energy optimization in embedded processor systems" SELSE 2014.

[50] S. Xi *et al.*, "Quantifying sources of error in McPAT and potential impacts on architectural studies" in Proceedings of *HPCA*, pp. 577–589, 2015.

[51] Y. Yetim, S. Malik, and M. Martonosi, "EPROF: An energy/performance/reliability optimization framework for streaming applications" in Proceedings of *ASP-DAC*, pp. 769–774, 2012.

[52] R. Zhang, M. Stan, and K. Skadron, "Hotspot 6.0: Validation, acceleration and extension" 2015.

[53] X. Zhang *et al.*, "Characterizing and evaluating voltage noise in multi-core near-threshold processors" in Proceedings of *ISLPED*, pp. 82–87, 2013.

[54] Y. Zhang and K. Chakrabarty, "A unified approach for fault tolerance and dynamic power management in fixed-priority real-time embedded systems" in *IEEE TCADICS*, vol. 25, no. 1, pp. 111–25, 2006.

[55] B. Zhao, H. Aydin, and D. Zhu, "Reliability-aware dynamic voltage scaling for energy-constrained real-time embedded systems" in Proceedings of *ICCD*, pp. 633–9, 2008.

[56] Z. Zheng *et al.*, "Co-analysis of RAS Log and Job Log on Blue Gene/P" in Proceedings of *IPDPS*, pp. 840–51, 2011.

[57] V. Zyuban *et al.*, "IBM POWER7+ design for higher frequency at fixed power" in *IBM J. Res. Dev.*, vol. 57, no. 6, p. 2:1, 2013.