

# EM-Reliability System Modeling and Performance Optimization for High-Performance Microprocessors

Zao Liu\*, Xin Huang\*, Tan Yu\*, Valeriy Sukharev<sup>†</sup>, Sheldon X.-D. Tan\*  
\*Department of Electrical Engineering, University of California, Riverside, CA 92521  
<sup>†</sup>Mentor Graphics Corporation, Fremont, CA 94538

**Abstract**—This article presents a new approach for system-level reliability management technique for multi-core microprocessors. In the new approach, the electromigration (EM) induced mean time to failure (MTTF) at the system level is modeled as a resource, which is abstracted from a more physics-based EM model. Based on such resource-based EM models, we propose a novel task migration method to explicitly balance the consumption of EM resources for all the cores. The new method leads to the equal chance of failure of these cores, which will maximize the life time of the multi-core system. Low power mode control is enabled to compensate the excessively consumed life time of all the cores, giving the multi/many core system more flexibility to handle heavy task assignment under a given reliability requirement. Experimental results on a 36-core processor platform shows that the proposed task migration scheme could balance and compensate the life time consumption of all the cores as desired.

## I. INTRODUCTION

Reliability is becoming a limiting constraint in high-performance microprocessor designs due to the high failure rates in deep sub-micron and nanoscale devices. The increase in failure rates is caused by high integration levels and higher power densities, which leads to excessive on-chip temperatures. The introduction of new materials, processes and devices, coupled with voltage scaling limitations and increasing power consumption will impose many new reliability challenges. The semiconductor industry faces the challenges to maintaining reliability such as the continued increase in die size and number of transistors and the constant scaling of transistors for performance [1]. Increasing transistor density and thus power density is causing higher temperatures on chip, resulting in failure acceleration. Scaling to smaller transistors increases failure rates by shrinking the thickness of dielectrics. This has led the International Technology Roadmap for Semiconductor (ITRS) to predict the onset of significant reliability problems in the future, and at a pace that has not been seen in the past [2].

Some initial efforts have been carried out for system level reliability analysis for SoCs (system-on-a-chip). RAMP [3] is the first architecture level tool for modeling the long-term processor reliability of microprocessors at the design stage. The follow-up work by the same authors proposed a dynamic reliability management (DRM) concept by dynamic voltage and frequency scaling (DVFS) [4]. It showed that it is not sufficient to just manage the temperature or power from the reliability perspective. Method in [5] shows that the power/performance and reliability are intrinsically conflicting metrics and have strong interactions on SoC designs, and proposes a joint policy optimization method. Another dynamic reliability management method was proposed in [6], in which a simple PID based run-time control was applied to optimize the performance subject to the long-term reliability constraints. Recently, DVFS techniques considering negative bias temperature instability (NBTI) effects were proposed for microprocessors [7]. A supply voltage scheduling technique was proposed for optimizing energy subject to NBTI constraints [8].

Despite of some early efforts, the research on system or architecture-level reliability analysis and optimization is still in its early stage. For electromigration (EM) related reliability effects, all the early efforts at architecture and system level are based on the simple semi-empirical Black's equation shown below to estimate the mean time to failure (MTTF) of interconnect wires and simplified

series reliability and constant failure rate models [3], [9].

$$MTTF = A j^{-n} \exp\{E_a/kT\} \quad (1)$$

Here,  $j$  is the current density,  $k$  is the Boltzmann's constant;  $T$  is the absolute temperature;  $n$  is the current density exponent;  $E_a$  is the EM activation energy. Typically the values of current density exponent  $n$ , and activation energy  $E_a$  in (1) were obtained at the stressed (accelerated) conditions. However, the two parameters actually depends on temperature and current density [10], [11]. As a result, the Black's equation will not be accurate for normal operational conditions of chips. Second, the Black's equation ignores the impacts of existing thermal stress and residual stress caused by the chip manufacture process and chip-package interactions. Third, existing approaches employed high pessimistic series reliability model to compute the reliability of interconnect wires. For practical mesh-structured power grid networks, which are more susceptible to EM reliability issues, there exists high-level redundancy as the failure of some of wire segments will not be necessary to result in the voltage drops below the critical threshold, which really determine the failure of the power grid networks.

This article presents a new approach for system-level reliability management technique for multi/many core microprocessors. In the new approach, the electromigration induced mean time to failure at the system level is modeled as a resource, which can be spent at different rates by the cores under different power and temperature settings. Based on the resource-based EM model, we propose a novel reliability management scheme to effectively balance and compensate the life time of the multi/many core system. The paper is organized as the followings: section II outlines the problem to solve, section III presents the resource based reliability model, section IV presents the proposed reliability management method, section V presents the experiments result on a 36-core processor, and section VI conclude this paper.

## II. PROBLEM FORMULATION AND RELIABILITY MODELING

As the reliability and performance are intrinsically conflicting factors, one has to consider them jointly at system level optimization as shown in the existing works [3], [5], [6]. For multi-core microprocessors, the optimization could be achieved through proper reliability management of resources and tasks. In this work, we treat MTTF as a reliability resource that could be consumed and controlled during task executions. For optimization purpose, instead of dealing with the difficult trade-off between performance and reliability in a given period, in this work, we target at finding a way to compensate the life time if it is excessively consumed during a certain period when the processor is loaded with heavy tasks.

## III. SYSTEM LEVEL EM-RELIABILITY RESOURCE CONSUMPTION MODEL

EM is a physical phenomenon of the migration of metal atoms along a direction of applied electrical field. Atoms (either lattice atoms or defects/impurities) migrate toward the anode end of metal wire along the trajectory of conducting electrons. Degradation of

the electrical resistance of interconnect segments, caused by void nucleation and growth, can be derived from the solution of kinetics equation describing the time evolution of stress in the interconnect segment [12], [13], [14], [15]. Recently, new stress based EM model for analysis of stress evolution caused by the electrical load in hundreds of millions interconnect segments has been proposed in [16] in which the model of void nucleation time and kinetics of void size evolution is developed to estimate the life time of the power grid.

Given the new physics-based EM model, we now introduce our system level EM-reliability resource consumption model. Our observation is based on the fact that EM-induced stress development process can be viewed as source consumption process, in which, the difference between the current stress and the critical stress, or the stress slack is the source. Once electrical current starts to flow through the wire, the EM process starts to spend the resource at a rate, which is the function of the temperature and current density. We notice that treating the EM as a resource was first introduced in [17]. But this work is still based on the traditional Black's equation.

At the system level, instead of using stress directly, we treat the life time of the processor specified by MTTF as a resource that could be consumed as the chip works. We define the specified MTTF as a nominal value, denoted as  $MTTF_N$ , which is the intended or required life of the chip under a typical temperature and power setting for a core or system. For example, a processor has nominal MTTF of 10 years under temperature of  $70^\circ C$  and working power of 20W as a specification. However, in reality, MTTF varies under different temperature and power settings. Hence, we denote the real MTTF as  $MTTF_R$ .

According to [17], the life time of the chip due to EM could be expressed as

$$life\ time = \frac{1}{(\sum_{k=1}^n (\Delta t_k \frac{1}{MTTF_{R,k}})) / T} \quad (2)$$

where  $MTTF_{R,k}$  is the actual MTTF under the  $k$ -th power and temperature settings for  $\Delta t_k$  period, assuming the chip works through  $n$  different power and temperature settings and  $T = \sum_{k=1}^n \Delta t_k$ . As a result, we propose a new EM-reliability resource consumption model. In this model, we treat the nominal MTTF ( $MTTF_N$ ) as the resource to consume, which actually is the *life time* defined in (2), and we define an average rate to consume  $MTTF_N$  as the amount of life time the working chip consumes during each unit time interval. In the nominal case, the chip is working under its specified temperature and power setting, and it has life time given by  $MTTF_N$ . Hence, the amount of life time consumed by the chip in each second is 1 EM second, that is to say, the nominal average consumption rate is  $r_N = 1$ . In reality, depending on different physical settings, the average consumption rate could be either higher or lower than its nominal rate, and we define average consumption rate as

$$r_R = \frac{MTTF_N}{MTTF_R} \quad (3)$$

in which the life time in real case ( $MTTF_R$ ) could be estimated by the new proposed reliability model in the previous sub-sections. With this definition, we could see that if  $MTTF_R > MTTF_N$ , then  $r_R < r_N$ , which indicates that the chip is consuming its nominal life time at lower rate, and thus the real life time is longer than the nominal one. Conversely, if  $MTTF_R < MTTF_N$ , then  $r_R > r_N$ , which indicates that the chip is consuming its nominal life time at higher rate, and thus the real life time is shorter than the nominal one. Hence, instead of saying MTTF changes, we perceive MTTF as a constant resource, which is given by  $MTTF_N$ , and it is the

consumption rate ( $r_R$ ) of MTTF that determines the real life time of the chip. If the time integration of EM slacks over a period is zero, then the life time or MTTF of the chip during that period will the  $MTTF_N$  as predicted by (2).

#### IV. EM-RELIABILITY MANAGEMENT METHOD

##### A. EM-reliability resource based task migration

First, we present the new task migration method to balance the EM-reliability resources, which is different from the conventional task migration method that targets at improving on-chip temperature profile. According to the definition of average MTTF consumption rate defined by (3), if  $r_R > r_N$  persistently, it will introduce excessive consumption of MTTF, which would possibly lead to early failure of the chip if no compensation is made. In real application, it is common that  $r_R > r_N$  during the period when heavy tasks are assigned to the chip, and the life time is excessively consumed during this period, while on the other hand, when light tasks are assigned to the chip,  $r_R < r_N$ , and less life time is consumed during this period. We define *MTTF resource slack* as the accumulative MTTF consumption difference between real case and nominal case over all different task execution periods, which is calculated through

$$S_d = \sum_{k=0} (r_N - r_R(k)) \Delta T \quad (4)$$

In which  $r_R(k)$  is the average consumption rate during the  $k$ -th execution cycle,  $\Delta T$  is the unit interval (UI) of the execution cycle, and  $k = 0$  indicates that the MTTF resource slack is accumulated from the very beginning when the chip first gets powered-on, and (4) illustrates the followings:

- If  $S_d = 0$ , it indicates the overall consumption of the chip would lead to its intended MTTF. It is easy to verify  $life\ time = MTTF_N$  by using (2) in this case.
- If  $S_d < 0$ , it indicates that the life time is excessively consumed for the past execution periods, and it requires compensations in future to avoid early failure.
- If  $S_d > 0$ , it indicates that the life time is consumed less than its nominal rate for the past execution periods, and it allows increased consumptions in future without causing early failure.

In a multi-core system, for each core  $i$ , we could calculate MTTF resource slack for core  $i$  at the end of each task execution cycle, and denote it as  $S_d(i)$ . We could also characterize the average power of the tasks in the coming execution cycle for each core. Assuming that the multi-core processor has  $N$  cores, and the average power of the tasks on each core are denoted as  $P_1, P_2, \dots, P_N$ , and the MTTF resource slack for each core are denoted as  $S_d(1), S_d(2), \dots, S_d(N)$ . To balance the EM-reliability of all the cores, we sort out the order of power consumptions and that of the MTTF resource slack, and assign the highest power to the core with highest value of  $S_d$ , and assign the second highest power to the core with the second highest value of  $S_d$ , and so on. The overall task migration scheme is shown in Fig.1, in which  $S_d$  is calculated based on (4) and (3), using the estimate  $MTTF_R$  through our proposed reliability model. In this way, the MTTF consumption of different cores could be balanced, which means that all the cores will be targeting at similar length of life time, avoiding early failure of some cores due to continuously heavy load assignment.

##### B. EM-reliability resource based low power control

With the proposed task migration scheme, the MTTF consumption of different cores is balanced so that all the cores would have comparable life time. However, task migration would not be able

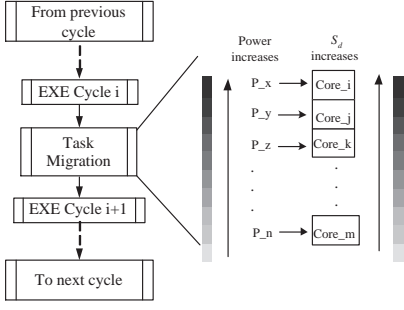


Fig. 1. The proposed reliability resource based task migration scheme

to compensate the excessively consumed MTTF if all the cores are loaded with heavy tasks. Hence, low power mode needs to be enabled to compensate the overly consumed MTTF later on, so that the chip could maintain its intended life time. With MTTF consumption getting balanced across all the cores, low power mode could effectively balance the life time of the cores as will be discussed later in this sub-section. Here, we first introduce the overall concept of the low power mode control to compensate the excessively consumed MTTF of a single core as illustrated in Fig. 2.

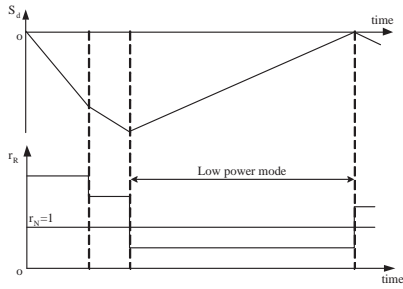


Fig. 2. Low power mode compensation scheme (one core)

From this Fig. 2 and (4), it is clear that  $S_d$  will start negative accumulation when  $r_R > r_N$ , which indicates faster consumption of MTTF comparing with nominal case. However, once the chip switches to low power mode with  $r_R < r_N$ , the excessively consumed MTTF starts to get compensated, and eventually, the consumed MTTF gets compensated to the nominal consumption of MTTF when  $S_d = 0$  over time.

Since the MTTF consumption could be compensated by low power mode with the proposed task migration scheme, Hence, in terms of MTTF resource compensation, we propose the following scheme to trade off heavy task execution and MTTF requirement.

- When working in high performance mode, the multi-core processor continues to keep this mode for  $N$  cycles after over 80% of cores have  $S_d < 0$ , and then switch to low power mode starting from the  $N + 1$  cycles.
- When working in low power mode, the multi-core processor continues to keep this mode for  $M$  cycles after over 80% of cores have  $S_d > 0$ , and then switch to high performance mode starting from the  $M + 1$  cycles.

In reality, the number  $N$  and  $M$  could be specified by the user based on the needs of handling heavy load and compensating life time consumption. In this way, the required MTTF of all the cores could be maintained through low power mode compensation, while the processor has the flexibility to handle heavy task assignment when

needed.

## V. NUMERICAL RESULTS AND DISCUSSIONS

The proposed reliability model is implemented in C++, and the task migration and low power mode control framework is built in Matlab environment. Hotspot [18] is used to build the thermal model based on the configuration of a 36 core processor, and Wattch [19] is used as the architecture level power simulation tool. In this work, we extend its functionality to calculate power under different supply voltage and working frequency. The dynamic workloads from spec2000 benchmarks (ammp, apsi, bzip2, equake, galgel, gcc, lucas, mesa, parser, twolf, vpr, applu, art, crafty, fma3d, gap, gzip, mcf, mgrid, swim vortex) are used as tasks to simulate power traces. We remark that the task migration for reliability balancing management could occur for relatively long period of time (hours, days or even weeks) in practical cases. Hence, in practice, the migration overhead is just negligible, and thus we do not need to consider any types of migration overhead in our test. Also, since it is impractical to run software based testing framework for task execution length over days and years, we have to 'scale down' the time frame in our testing environment. To do this, we scale down the length of task execution interval, and perform task migration at the end of each interval. In this way, we could find out how the MTTF are consumed for each core over several task execution intervals. In our testing environment, we keep each time step to be  $30 \mu s$  for power simulation and thermal simulation, and specify each task execution interval to be 362 time steps. The nominal MTTF for each core is set to 15 years, and the 36 core processor is used as testing vehicle. In our framework, our processor has two different P-states, one is high performance P-state, with 1 G of frequency and 1.4 V of supply voltage; and the other is low power P-state, with 800 M of frequency and 1.12 V of supply voltage.

First, we disable low power mode control and use the proposed task migration method to balance the reliability across all the cores. The experimental results of the MTTF resource slack as reflected by  $S_d$  is shown in Fig. 3, in which the unit of  $S_d$  is normalized to UI (unit interval of the task execution as defined by  $\Delta T$  in (4)), and the unit of time is measured by task execution cycles. We could clearly see that MTTF consumption is balanced across all the cores, and thus, the MTTF is consumed in similar rate, which indicates that all the cores are regulated to have similar life time.

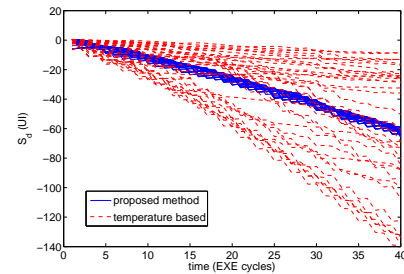


Fig. 3. MTTF resource slack (represented by  $S_d$ ) under different task migration schemes

In addition, as a comparison, we also implement temperature based task migration scheme that migrates the heaviest tasks to the cores with lowest temperature and testify its performance in terms of MTTF resource slack. As demonstrated by Fig. 3, the temperature oriented task migration could not balance MTTF consumption, and we could clearly observe that some of the cores are consuming significantly more MTTF than others, leading to imbalanced MTTF consumption.

Since the result in Fig. 3 has confirmed that the proposed task migration scheme could balance the MTTF consumption across all the cores, making the cores targeting at similar life time, the low power control could now effectively compensate the MTTF consumption by switching the processor to low power mode. In this part of experiment, the low power control mode is setup as the following: In high performance state, if over 80% of the cores have  $S_d < 0$  for over 10 task execution cycles ( $N = 10$ ), the processor switches to low power mode. In low power mode, if over 80% of cores have  $S_d > 0$  for over 1 cycle ( $M = 1$ ), the processor switches back to high performance state. After enabling the low power control, as Fig. 4 shows, the processor starts with high performance mode, in which the MTTF is excessively consumed for all the cores, and  $S_d$  of all the cores are decreasing simultaneously. After 10 task execution cycles after 80% of the cores have  $S_d < 0$ , the processor switches to low power mode. Once the processor switches to low power mode, values of  $S_d$  start to accumulate in positive direction because all the cores are now consuming MTTF at lower rate than the nominal rate, and we could clearly observe that  $S_d$  values calculated from different cores get effectively compensated to around 0 as the task runs under the proposed task migration scheme, which indicates that the overall MTTF consumption is close to the nominal case and the cores are targeting at achieving their required MTTF. The calculated standard deviation of MTTF resource slack by the end of the 40 execution cycle is 2.27 UI, which is converged and would not keep increasing as more tasks get executed.

On the other hand, if we use temperature based migration, low power mode could not be used to compensate the MTTF consumption, as Fig. 4 shows, because the MTTF consumption for different cores are completely different, and the values of  $S_d$  diverge as task runs, which indicates the life time of different cores diverges, and some cores would likely to have early failure if tasks are executed under this scheme. The calculated standard deviation of the MTTF resource slack by the end of the 40 execution cycle is 41.08 UI, which is around 18 times larger than that of the standard deviation using the proposed method. And this standard deviation will increase as more tasks are executed.

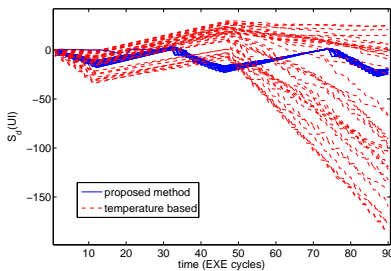


Fig. 4. MTTF resource slack (represented by  $S_d$ ) compensation using low power mode under different task migration schemes

The result from Fig. 4 also suggests that, the processor could be assigned with heavy load for a certain period under the proposed migration scheme. Because, as long as the cores could be balanced to have comparable MTTF, their life time consumption could be effectively compensated by low power mode.

## VI. CONCLUSION

This article presents a new reliability management method to balance and control the life time of multi-core processor chip due to electromigration (EM) process. A more physics-based EM model

was used for more accurate prediction of the MTTF without using empirical solutions. The proposed reliability management treats the MTTF as a resource to consume during task execution, and uses task migration to balance the MTTF consumption across different cores, which leads to comparable life time among different cores to maximize the life time of the whole system. With the balanced MTTF consumption, a low power control mode can effectively compensate the MTTF consumption of all the cores after executing heavy task loads, which gives the processor the flexibility to handle heavy load when needed since the excessively consumed MTTF could be compensated later on. The experimental results confirm that, comparing with the proposed approach, the traditional temperature-based task migration approach could not balance MTTF consumption, and the low power mode could not effectively compensate the life time of all the cores.

## REFERENCES

- [1] "Critical Reliability Challenges for The International Technology Roadmap for Semiconductors (ITRS)." In International Sematech Technology Transfer Document 03024377A-TR, 2003.
- [2] "International technology roadmap for semiconductors (ITRS), 2012 update," 2012. <http://public.itrs.net>.
- [3] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "Ramp: A model for reliability aware microprocessor design," *IBM Research Report*, 2003.
- [4] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "The case for lifetime reliability-aware microprocessors," in *Computer Architecture, 2004. Proceedings. 31st Annual International Symposium on*, pp. 276–287, 2004.
- [5] T. Simunic, K. Mihic, and G. Micheli, *Optimization of Reliability and Power Consumption in Systems on a Chip*, vol. 3728 of *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.
- [6] E. Karl, D. Blaauw, D. Sylvester, and T. Mudge, "Reliability modeling and management in dynamic microprocessor-based systems," in *Design Automation Conference, 2006 43rd ACM/IEEE*, pp. 1057–1060, 2006.
- [7] M. Basoglu, M. Orshansky, and M. Erez, "NBTI-aware DVFS: A new approach to saving energy and increasing processor lifetime," *Low-Power Electronics and Design (ISLPED), 2010 ACM/IEEE International Symposium on*, pp. 253–258, 2010.
- [8] A. Calimera, E. Macii, and M. Poncino, "Energy-optimal SRAM supply voltage scheduling under lifetime and error constraints," in *Design Automation Conference (DAC), 2013 50th ACM / EDAC / IEEE*, pp. 1–6, 2013.
- [9] D. Brooks, R. P. Dick, R. Joseph, and L. Shang, "Power, Thermal, and Reliability Modeling in Nanometer-Scale Microprocessors," *Micro*, 2007.
- [10] M. Ohring, *Reliability and Failure of Electronic Materials and Devices - Milton Ohring - Google Books*. San Diego: Academic Press, 1998.
- [11] M. Hauschildt, C. Hennessy, G. Talut, O. Aubel, M. Gall, K. B. Yeap, and E. Zschech, "Electromigration early failure void nucleation and growth phenomena in Cu and Cu(Mn) interconnects," in *2013 IEEE International Reliability Physics Symposium (IRPS)*, pp. 2C.1.1–2C.1.6, IEEE, 2013.
- [12] M. A. Korhonen, P. Borgesen, K. N. Tu, and C. Y. Li, "Stress evolution due to electromigration in confined metal lines," *Journal of Applied Physics*, vol. 73, no. 8, pp. 3790–3799, 1993.
- [13] J. J. Clement, "Reliability analysis for encapsulated interconnect lines under dc and pulsed dc current using a continuum electromigration transport model," *Journal of Applied Physics*, vol. 82, no. 12, pp. 5991–6000, 1997.
- [14] M. E. Sarychev, Y. V. Zhitnikov, L. Borucki, C.-L. Liu, and T. M. Makhviladze, "General model for mechanical stress evolution during electromigration," *Journal of Applied Physics*, vol. 86, no. 6, pp. 3068–3075, 1999.
- [15] V. Sukharev, A. Kteyan, E. Zschech, and W. D. Nix, "Microstructure Effect on EM-Induced Degradations in Dual Inlaid Copper Interconnects," *IEEE Transactions on Device and Materials Reliability*, vol. 9, no. 1, pp. 87–97, 2009.
- [16] X. Huang, T. Yu, V. Sukharev, and S. X.-D. Tan, "Physics-based Electromigration Assessment for Power Grid Networks," in *Design Automation Conference (DAC), 2014 51th ACM / EDAC / IEEE*, pp. 1–6, 2014.
- [17] Z. Lu, W. Huang, J. Lach, M. Stan, and K. Skadron, "Interconnect lifetime prediction under dynamic stress for reliability-aware design," in *Proc. IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pp. 327–334, IEEE, Nov. 2004.
- [18] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, pp. 2–13, 2003.
- [19] D. Brooks, V. Tiwari, and M. Martonosi, "Watch: A framework for architectural-level power analysis and optimizations," in *Proc. Int. Symp. on Computer Architecture (ISCA)*, pp. 83–94, 2000.