Two-Level Soft Error Vulnerability Prediction on SMT/CMP Architectures

Lide Duan[1], Lu Peng[1], and Bin Li[2]
[1]Department of Electrical and Computer Engineering, [2]Department of Experimental Statistics
Louisiana State University, Baton Rouge, LA 70803
{lduan1, bli, lpeng}@lsu.edu

Architectural Vulnerability Factor (AVF) [3] quantifies the probability that a raw soft error finally produces a visible error in the program output. It is often used by computer designers as an important reliability metric at the architectural level. However, the AVF measurement is extremely expensive in terms of hardware and computation. In this paper, we characterize and predict a program's AVF under resource contention and sharing with other programs running on *Simultaneous Multithreading* (SMT) and *Chip-Multiprocessor* (CMP) architectures.

Background. First, inter-thread resource contention and sharing significantly and non-uniformly affect the AVF of processor structures. Based on our experiments, a processor structure's AVF shows strong variation when a benchmark (e.g. *gcc*) is co-scheduled with different benchmarks on this processor. Second, the prediction complexity and scope are significantly enlarged in the context of multi-threading. Walcott et al. [4] performed the prediction across SPEC CPU 2000 benchmarks on a fixed machine configuration; Duan et al. [1] extended their prediction to be across a very small set of configurations. Nevertheless, both of their works were restricted to single-threaded processors with certain simplifications. In contrast, this work correlates two important (but complex) problems: the processor configuration being from a statistically large design space, and the prediction effectiveness across different multi-programmed workloads.

Our proposal: two-level predictive modeling. In this work, we propose a scalable two-level predictive mechanism for predictions on SMT/CMP architectures. At the first level, a cross-program model is trained to predict the contention-free AVF on a single-threaded processor. The inputs to the first level model include a few important performance measurements (e.g. structure occupancies, cache miss rates) from the contention-free execution and the corresponding configuration parameters. The output of the first level model, along with key processor structures' occupancies measured when the program runs against other program(s) on a multi-threaded processor, are inputted to the second level model, which finally predicts the program's AVF under resource contention with others.

Essentially, the first level model uses key parameters and simple performance measurements to characterize hardware configuration and the software program, respectively; while the second level model captures the inter-thread resource contention and sharing. By employing the proposed two-level prediction, we can obtain an accurate estimate of a program's AVF when it is co-scheduled with different programs on an arbitrarily configured SMT/CMP whose configuration is from a large design space.

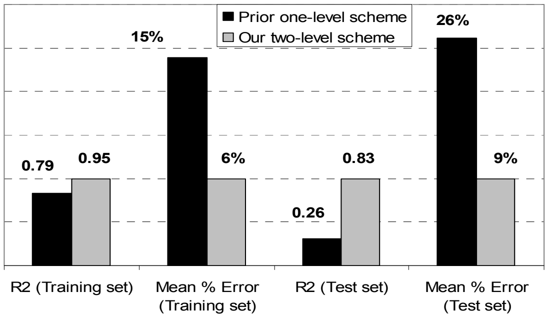

Figure 1. Comparison between the proposed two-level scheme and prior one-level scheme.

Why two levels? Simply applying prior one-level approach [4][1][2] fails to make accurate predictions on multi-threaded processors. For demonstration, we trained a one-level model to predict SMT processors' register file AVF. We used the same prediction technique as in [1] since it demonstrated better performance than linear regression used in [4]; the inputs to this model are the various performance measurements when different threads are run simultaneously. Figure 1 compares the prediction accuracy between the two models in terms of R-square (higher is better with 1 as the maximum) and mean percentage error (lower is better with 0% as the minimum). We can see that the one-level scheme shows more than two times higher error rates than the two-level scheme. Especially for the test set, the mean percentage error reaches a very high value (26%) while the R-square is unacceptably low (0.26). Essentially, decoupling the prediction into two levels reduces the model complexity at each level and also improves prediction accuracy.

Contributions. In summary, the main contributions of this paper are as follows:

- **Universal prediction of the AVF on single-threaded processors**: the first level model accurately predicts the AVF without contention on any given processor configuration in the design space.
- **Universal prediction of the AVF under contention across multi-programmed workloads**: the second level model takes the knowledge of the contention-free AVF from the first level model, and performs an accurate prediction of the AVF under resource contention for any program combination in analysis.
- **A case study of soft error resilient thread-to-core scheduling**: the proposed AVF prediction can be used to identify the optimal thread-to-core assignment that minimizes the AVF of a Chip Multi-Threaded Processor.

References

[1] L. Duan, B. Li and L. Peng, "Versatile Prediction and Fast Estimation of Architectural Vulnerability Factor from Processor Performance Metrics," in *HPCA* 2009.

[2] L. Duan, Y. Zhang, B. Li, and L. Peng, "Universal Rules Guided Design Parameter Selection for Soft Error Resilient Processors," In *ISPASS* 2011.

[3] S. Mukherjee, C. Weaver, J. Emer, S. Reinhardt, and T. Austin, "A Systematic Methodology to Compute the Architectural Vulnerability Factors for a High-Performance Microprocessor," In *MICRO* 2003.

[4] K. Walcott, G. Humphreys, and S. Gurumurthi, "Dynamic Prediction of Architectural Vulnerability from Microarchitectural State," In *ISCA* 2007.