Characterizing the Soft Error Vulnerability of Multicores Running Multithreaded Applications

Niranjan Soundararajan, Anand Sivasubramaniam, Vijay Narayanan
Dept. of Computer Science and Engineering, The Pennsylvania State University
University Park, PA, USA
soundar@cse.psu.edu, anand@cse.psu.edu, vijay@cse.psu.edu

ABSTRACT

Multicores have become the platform of choice across all market segments. Cost-effective protection against soft errors is important in these environments, due to the need to move to lower technology generations and the exploding number of transistors on a chip. While multicores offer the flexibility of varying the number of application threads and the number of cores on which they run, the reliability impact of choosing one configuration over another is unclear. Our study reveals that the reliability costs vary dramatically between configurations and being unaware could lead to a sub-optimal choice.

Categories and Subject Descriptors: B.8.1 [**Performance and Reliability**]: Reliability, Testing, and Fault-Tolerance

General Terms: Experimentation, Measurement.

Keywords: Multicore, Soft Errors, FIT rate

1. INTRODUCTION

Chip-Multiprocessors (CMPs) or multicores [2], allow the exploitation of thread-level parallelism (TLP), as opposed to the traditional instruction-level parallelism (ILP) which has reached its limits with regard to meeting application demands. A side-effect of the increasing transistor count, due to additional cores, is the requirement to lower operating voltages for keeping total system power within bounds. However, this leads to transistors becoming more prone to transient faults. Soft errors are an important class of transient faults whose impact is increasing greatly between technology generations, 8% per bit in each generation. Soft errors [7, 1] occur when high energy neutron particles from the atmosphere or alpha particles strike and cause transistor state to flip. Incidents like [6] underscore the importance of increasing circuit robustness to soft errors. While ECCs are an effective protection technique for caches, the computing datapaths are harder to protect since any protection for these involve significant overheads [5].

Soft error vulnerability is captured using Failures in Time (FIT) rates [4], which gives the number of failures in a billion hours of operation. Error masking occurs due to corrupt data values not being read and values getting overwritten before being read. This deration of the raw error rate is captured using a metric called the Architectural Vulnerability Factor (AVF) [4]. AVF gives the fraction of the total bits in a hardware structure that is vulnerable to soft errors. The effective soft error rate, hence, is given by $FIT_{effective} = FIT_{raw} * AVF$. Higher AVF means that there is a greater probability for the soft error in a particular cycle to affect the final output.

SIGMETRICS'10, June 14–18, 2010, New York, New York, USA.
ACM 978-1-4503-0038-4/10/06.

2. MOTIVATION

Analyzing the vulnerability of multicore platforms to soft errors is the first step to developing and implementing efficient protection mechanisms as it provides better insight into the level of protection required. In this work we look at the impact of two specific parameters, the number of cores on which the application is run and the number of application threads. Even for a given platform, the choices made dramatically impacts the soft error vulnerability of the underlying hardware. Where a designer would have required protection mechanisms to meet reliability demands, a change in configuration might provide it at lower power, performance and hardware overheads. To our knowledge, there has been no prior work attempting to study the influence of the impact of varying the number of threads and the cores on which they run on soft error vulnerability of multicores.

3. ANALYSIS

We developed a very detailed AVF analysis framework on the GEMS full system simulator [3] on which we study the vulnerability of multicore platforms. The number of cores and application threads are varied between 2, 4 and 8. Each of the configurations involve a 2-level cache hierarchy (32KB private L1 and 2MB shared L2). Opal has been modified to support a P6-style 2-way out-of-order pipeline. Instructions spend significant percentage of their lifetime in the Reorder Buffer (ROB) and hence it has a large impact on the overall core AVF. Given this and space constraints, we focus our analysis using the ROB as representative of the core, in the rest of this paper.

System AVF and System Vulnerability: While AVF is directly proportional to the soft error rate, see section 1, it is normalized with respect to time. It gives an average per cycle vulnerability of a structure to soft errors. Therefore when analyzing the reliability impact of configurations with varying runtimes, studying the impact on AVF, alone, does not provide the full picture. Hence we also capture the impact on the cumulative vulnerability ($AVF * Time$).

For reliable operation, it is desirable to lower AVF and the cumulative vulnerability.

Given that we vary the number of cores, we use System AVF, AVF sum across all cores, and System Vulnerability to capture the associated reliability tradeoffs. Note that while AVF is a fractional value, System AVF is not. System AVF is used to study the relative per-cycle soft error vulnerability of the different multicore platforms.

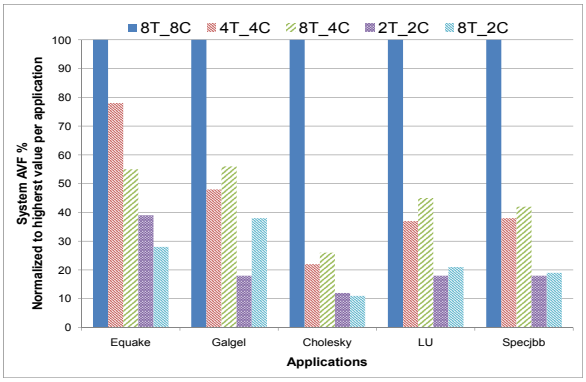

Figure 1: System AVF across different configurations. XT_YC is X threads running on Y cores.

3.1 Impact on System AVF

Figure 1 shows the varying impact on system AVF across configurations. It is apparent that the greater availability of cores increases the System AVF. In certain cases like cholesky, System AVF increases by a factor larger than the scaling with respect to cores. The main reason for this being the mismatch between performance scalability of applications and available hardware. For applications like cholesky and specjbb, we observed that running on 4 cores provided almost 80% of 8-core performance. The lack of throughput increase the instruction occupancy time in the ROB resulting in the rise in System AVF.

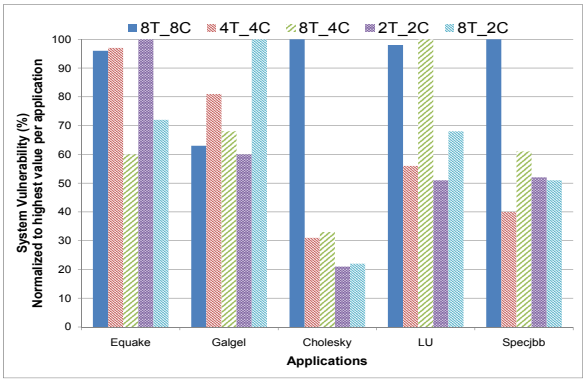

Figure 2: System Vulnerability across different configurations. XT_YC is X threads running on Y cores.

3.2 Impact on System Vulnerability

Figure 2 shows that decreasing the cores does not imply a reduction in the System Vulnerability. While the 8 core run had the highest AVF, in applications like equake, galgel and LU, the 4 or 2 core configurations lead to higher System Vulnerability. Factors like increased number of cache misses, lower ILP compared to the TLP combined with the increase in runtime, increase the System Vulnerability. This emphasizes that the configuration choices with respect to optimizing the reliability parameters (AVF and vulnerability) are not straight-forward. Figures 1,2 show that for a given number of cores, the number of application threads do not seem to categorically impact either parameter in a specific manner.

4. CONCLUSION

This work presented the first study on the impact of soft error vulnerability of a multicore platform running multi-threaded applications. Varying the number of cores and application threads to capture the System AVF and System Vulnerability illustrated that the associated performance-reliability tradeoffs are not straight-forward. Across different applications, we observed that certain configurations led to a dramatic increase in the System AVF not matched by the speedup they offer while other configurations with lesser number of cores had a sub-optimal impact on the System Vulnerability.

Acknowledgments The work was supported by NSF grants 0916887, 0702617, 0615097 and 0621429.

5. REFERENCES

[1] D. Bossen. CMOS Soft Errors and Server Design. In *IEEE Reliability Physics Tutorial Notes*, April 2002.

[2] Intel(R). Intel previews Intel Xeon(R) 'Nehalem-EX' processor. 2009.

[3] M. M. K. Martin, D. J. Sorin, B. M. Beckmann, M. R. Marty, M. Xu, A. R. Alameldeen, K. E. Moore, M. D. Hill, and D. A. Wood. Multifacet's General Execution-driven Multiprocessor Simulator (GEMS) toolset. *SIGARCH Comput. Archit. News*, 33(4):92–99, 2005.

[4] 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 *Proceedings of the International Symposium on Microarchitecture (MICRO)*, pages 29–40, December 2003.

[5] S. Reinhardt and S. Mukherjee. Transient Fault Detection via Simultaneous Multithreading. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 25–36, June 2000.

[6] C. Systems. Cisco 12000 single event upset failures overview and work around summary. 2003.

[7] C. Weaver, J. Emer, S. Mukherjee, and S. Reinhardt. Techniques to Reduce the Soft Error Rate of High-Performance Microprocessor. In *Proceedings of the International Symposium on Computer Architecture (ISCA)*, pages 264–275, June 2004.