DOE ECPI Award DE-FG02-ER25689. Final Technical Report

Dimitrios S. Nikolopoulos
Department of Computer Science, College of William&Mary
dsn@cs.wm.edu, http://www.cs.wm.edu/~dsn

The research conducted until 08/14/06 has led to the implementation of the MELISSES continuous performance profiler. More specifically, we have designed, implemented, robustified and released PACMAN (available at http://www.cs.wm.edu/pacman), an implementation of our continuous profiler which provides accurate hardware event counters on a thread-local basis, at sub-microsecond granularity on Intel Hyperthreaded processors. PACMAN has been used to implement a number of performance and power-related optimizations for multithreaded codes running on layered parallel architectures.

The first successful demonstration of MELISSES capabilities was a profile-driven parallelization scheme for multithreaded codes, in each parallel regions was parallelized individually using either speculative precomputation with helper threads, or non-speculative thread-level parallelization. Regions that exhibit ample instruction-level parallelism with low memory access rates are parallelized with conventional TLP methods, whereas regions with limited instruction-level parallelism and high memory access rates are not parallelized. They are executed instead with speculative precomputation, which preexecutes long-latency memory accesses. MELISSES assists in locating the critical memory accesses that are responsible for most of memory latency and are offloaded for precomputation on helper threads. Runtime mechanisms and schemes for combining TLP with speculative precomputation via the use of MELISSES were presented in publications [6]. Another relevant publication [8] addressed the problem of devising effective speculative precomputation schemes for floating point scientific codes.

The design and implementation of PACMAN is discussed in [1]. Recently, we deployed MELISSES and our continuous monitoring technology to achieve simultaneous optimization of performance and power on layered multicore platforms. The results of this work appear in [3, 2, 4]. The distinguishing aspect of this work is that it is the first to demonstrate concurrent improvement of both power and performance on a high-end computing platform. Using MELISSES and runtime scalability predictors, a technology we developed from scratch to accurately characterize and predict power and performance in phases of multithreaded code using non-linear regression, we have been able to improve performance by 22% on average, while reducing energy consumption by 26% on average, on Intel platforms with up to 8 cores distributed between 4 processors. More specifically, MELISSES isolates phases of multithreaded execution delimited by loops and function calls, characterizes each phase in terms of scaling at all layers of a parallel architecture (including the processor layer, the core layer within processors and the thread layer within cores), and locates an execution *sweet sport* for each phase, in which maximum scalability is retained while the system deactivates threads, cores or entire processors to reduce power consumption.

We have used a module of MELISSES which conducts statistical analysis of memory references, to monitor cache access behavior in the SimICS multiprocessor simulator. This monitoring module has been used to derive dynamic data re-mapping algorithms for large L2 caches [7]. In a continuation of this work, we are using MELISSES on SimICS, to implement speculative precomputation schemes that reduce remote

memory access latency on layered parallel architectures with non-uniform memory access latency both across processors and within the processor memory hierarchy. The intent of this effort is to overcome the limitations of traditional data distribution and dynamic data migration schemes, while these schemes attempt to reduce remote memory accesses to pages shared actively by multiple processors.

MELISSES has stimulated research on architectural and operating system support for hardware performance monitors. In a recent position paper [5], we outline MELISSES and how the ideas therein can be extended towards developing hardware monitors with multiple event accounting contexts and capabilities for multidimensional performance, power and temperature characterization on multicore processors.

We are in the process of porting MELISSES to AMD Opteron Dual-Core Socket-F processors. The AMD Opteron port will enable us to stress-test MELISSES and its capabilities for simultaneous optimization of performance, power and temperature on multicore platforms, using two power-aware clusters currently under construction.

References

[1] M. Curtis-Maury, C. Antonopoulos, and D. Nikolopoulos. PACMAN: A PerformAnce Counters MANager for Intel Hyperthreaded Processors. In *Proc. of the 3rd International Conference on the Quantitative Evaluation of Systems*, Riverside, CA, September 2006.

[2] M. Curtis-Maury, J. Dzierwa, C. Antonopoulos, and D. Nikolopoulos. Online Power-Performance Adaptation of Multithreaded Programs via Event-Based Prediction. In *Proceedings of the 20th ACM International Conference on Supercomputing*, Queensland, Australia, July 2006.

[3] M. Curtis-Maury, J. Dzierwa, C. Antonopoulos, and D. Nikolopoulos. Online Strategies for High-Performance Power-Aware Thread Execution on' Emerging Multiprocessors. In *Proc. of the Second Workshop on High-Performance Power-Aware Computing, held in conjunction with IEEE/ACM International Parallel and Distributed Processing Symposium*, Rhodes, Greece, April 2006.

[4] M. Curtis-Maury, J. Dzierwa, C. Antonopoulos, and D. Nikolopoulos. On the Design of Online Predictors for Autonomic Power-Performance Adaptation of Multithreaded Programs. *Journal of Autonomic and Trusted Computing*, 2007. to appear.

[5] M. Curtis-Maury, D. Nikolopoulos, and C. Antonopoulos. Dynamic Program Stirring on Multiple Cores: How Hardware Performance Monitors can Help Regulate Performance, Power, and Temperature Simultaneously. In *Proc. of the Second Workshop on Functionality of Hardware Performance Monitors (held in conjunction with MICRO-39)*, Orlando, FL, December 2006.

[6] M. Curtis-Maury, T. Wang, C. Antonopoulos, and D. Nikolopoulos. Integrating Multiple Forms of Multithreaded Execution on SMT Processors: A Quantitative Study with Scientific Workloads. In *Proc. of the Second International Conference on the Quantitative Evaluation of Systems (QEST'2005)*, pages 199–208, Torino, Italy, September 2005.

[7] X. Ding, D. Nikolopoulos, S. Jiang, and X. Zhang. MESA: Integrated Static and Runtime Cache Management for Avoiding Conflicts. In *Proc. of the 2006 International Symposium on Performance Analysis of Systems and Software*, pages 189–198, Austin, TX, March 2006.

[8] T. Wang, C. Antonopoulos, and D. Nikolopoulos. smt-SPRINTS: Software Precomputation with Intelligent Streaming for Resource-Constrained SMTs. In *Proc. of the 11th European Conference on Parallel Computing (EuroPar'2005)*, pages 710–719, Lisbon, Portugal, August 2005.