Final Report for DOE ECPI Grant ER25751

Dimitrios S. Nikolopoulos
Department of Computer Science, Virginia Tech
dsn@cs.vt.edu, http://www.cs.vt.edu/˜dsn

In the following sections, we summarize the contributions made through support from this DOE ECPI award to research and training in advanced computing systems.

1 Dynamic scheduling of layered parallelism on emerging multi-core processors and many-core clusters

We have developed several schedulers for dynamic multi-grain parallelization on the Cell Broadband Engine. The Cell processor presents a new paradigm for parallel computing on multicore platforms, by combining conventional processor cores with customized accelerators and by offering an explicitly managed memory hierarchy to programmers, for tighter control of locality and performance. Parallel computation on the Cell is accomplished by off-loading compute-intensive and data-intensive code from the conventional cores to the vector SIMD accelerators. Heterogeneous multi-core architectures such as the Cell represent a design point in computer architecture which holds greater promise for sustaining high performance and power-efficiency than conventional, homogeneous multi-core architectures. Cell is also the processor of choice for Roadrunner, a Petaflop-capable supercomputer currently in the development phase by IBM. Due to these reasons, we believe that the research conducted on Cell with support from the DOE ECPI award is timely, relevant and in line with DOE missions.

The first of the novel schedulers developed in this activity, named MGPS-SLED (for Multi-grain Parallelism Scheduling using Slack Minimizing Event-Drive execution), exploits effectively thread-level and data-level (SIMD) parallelism at runtime, without prior knowledge of the application or input from the programmer. MGPS-SLED follows an event-driven execution model for scheduling tasks and data parallelism of varying granularity, on the synergistic processing elements (SPE) of the Cell. MGPS-SLED provides a novel mechanism for deciding between task-level, loop-level and data-level parallelization on the fly, based on runtime workload characterization and observable utilization metrics on the SPEs. As part of the MGPS-SLED effort, we have ported the MELISSES hardware monitor on the Cell PPE and SPE —the conventional power processing element and the synergistic processing elements of the processor respectively—, to collect continuous data on SPE and PPE utilization and drive the multi-grain decomposition and scheduling processes. More specifically, MELISSES enabled us to collect a historical profile of task execution on the SPE, which in conjunction with program phase analysis, enabled MGPS-SLED to adaptively select the layers and degrees of parallelism to activate in any phase of the program. We emphasize the major contribution of MGPS-SLED, namely phase-aware optimization of the scheduling process, which would have been impossible without leveraging the MELISSES performance monitoring framework. Phase-aware program control in MELISSES has enabled unprecedented performance and power optimizations in parallel programs. We view this result as one of the major contributions of this effort.

MGPS-SLED was initially tested with RAxML, a computational phylogeny toolkit that uses algorithms based on the Maximum Likelihood (ML) method and rapid bootstrapping. The results of this work originally appeared in a publication in the 11th ACM SIGPLAN Symposium on Principles and Practice of Parallel

Programming [6]. This publication has been awarded with the Best Paper Award of the conference, and has since then received extensive attention and a high number of citations from the community. Three more papers exploring different scheduling policies for dynamic multi-grain parallelization and using the MGPS-SLED framework to accelerate applications that compute large-scale phylogenies appeared later [7, 8, 21]. We have also explored new optimizations of codes with multi-grain parallelism on the Cell BE. This exploration resulted in the development of what appears to be the fastest computational phylogeny code ever to run on a single chip. The results of this effort appeared in [5], a contribution which won the first prize in the annual Virginia Tech High End Computing Challenge[1].

We have also explored the development of parallel computation models for accelerator-based architectures, focusing on multi-processors and clusters based on heterogeneous multi-core processors. We developed an extension of the LogP model of parallel computation, which factors heterogeneity in computation and communication substrates in the estimates of execution time. The model that we developed is used as a rapid prototyping tool for exploring the design space of programming models and mappings of parallel computation onto accelerator-based architectures. Our earlier experience with MGPS-SLED suggested that the parallelization and mapping tasks can be arduous for programmers, as parallel architectures introduce more layers of parallelism. We strived to tame some of this complexity via analytic modeling and complemented the developed model with the runtime phase analysis and optimization modules that we developed earlier in this activity. A paper discussing our model of multi-grain parallelism was presented in the HiPEAC'2008 conference [4]. A scheduler based on our modeling was designed, implemented and evaluated on small- and medium-scale clusters of nodes with computational accelerators based on the Cell processor and the related work was presented in a paper at the CCGrid'08 conference [3].

The MGPS-SLED scheduler formed the basis for further work on supporting specific programming models and forms of parallelism on computational accelerator-based architectures based on the Cell processor. Based on MGPS we developed support for implementing the MapReduce programming model on clusters based on Cell blades [20], as well as for computational biology codes using tiled wave-front parallelism [1].

2 The MELISSES continuous hardware monitor

The MELISSES continuous hardware monitor has been designed, developed, and evaluated with support from this DOE ECPI award. The key novelty of the MELISSES monitor is its lightweight nature, which provides new capabilities for dynamic online optimization of parallel applications. The MELISSES monitor has been used extensively in designing effective predictors for dynamic performance prediction and optimization in parallel programs.

More specifically, the monitor was deployed for prediction-based program adaptation. In this work, we focused on designing phase-aware predictors of performance and scalability, which can be used to pinpoint scalability and performance bottlenecks in parallel code running on multi-core processors, as well as opportunities for improving performance or sustaining performance while reducing the dynamic power consumption of the processor. The latter are available through concurrency throttling, a technique which adjusts on-the-fly the degree of active concurrency in the program, so that the program uses the minimum number of cores necessary to sustain the highest level of performance possible. We have presented results on a study of the design space of on-line and off-line predictors for dynamic, phase-aware program adaptation in several conference and journal papers [12, 11, 10, 18, 16, 17]. This research was facilitated through a collaboration between the PI and Lawrence Livermore National Laboratory. Matthew Curtis-Maury, a recently graudated Ph.D. student of the PI, worked as an intern in LLNL during the summer months of 2007

[1] The VT High End Computing Challenge is an annual competition organized similarly to the Gordon Bell Award Competition held annually at the SC conference. Entries are judged by peer reviewers based on full paper submissions and results from actual supercomputers which can be reproduced remotely.

and collaborated with Dr. Bronis de Supinski and Dr. Martin Schulz, of the Center for Applied Scientific Computing, in this research.

We have ported the MELISSES to AMD Opteron Quad-Core Socket-F processors. The AMD Opteron port enabled us to stress-test MELISSES and its capabilities for simultaneous optimization of performance, power and temperature on multicore platforms with NUMA architecture, using two power-aware clusters at Virginia Tech. We have also investigated virtualization of the MELISSES infrastructure, using the Xen hypervisor. A preliminary publication on the integration of the MELISSES hardware event monitor in virtualized environments appeared in [2].

3 Earlier contributions

We hereby provide a summary of earlier contributions form this research, all of which have been documented in earlier reports:

We have robustified and released PACMAN (http://www.cs.wm.edu/pacman), an implementation of our continuous profiler which provides accurate hardware event counters on a thread-local basis, at submicrosecond 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 [15]. Another relevant publication [22] addressed the problem of devising effective speculative precomputation schemes for floating point scientific codes.

The design and implementation of PACMAN is discussed in [9]. We deployed MELISSES and our continuous monitoring technology to achieve simultaneous optimization of performance and power on several layered multicore platforms. The results of this work appeared in [12, 11, 13]. 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 spot* 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 [19]. In a continuation of this work, we used MELISSES on the SimICS full-system simulation platform, 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 was 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 position paper [14], we outlined 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.

4 Training Activities

The DOE ECPI award supported fully or partially as graduate research assistants three graduate students of the Computer Science Program at Virginia Tech, specifically, Mr. Matthew Curtis-Maury (graduated with a Ph.D. in Computer Science in May 2008), Mr. Filip Blagojevic (graduating with a Ph.D. in Computer Science in December 2008), and Mr. Jae-seung Yeom (Ph.D. in progress). One completed Ph.D. thesis (Matthew Curtis-Maury, on "Improving the Efficiency of Parallel Applications on Multithreaded and Multicore Systems", defended March 19, 2008) was fully based on and supported by this award. We also developed extensive course material originating from the research supported by this award and used this material in undergraduate and graduate courses on Parallel Computation, Computer Architecture, and Operating Systems.

References

[1] Ashwin M. Aji, Wu chun Feng, Filip Blagojevic, and Dimitrios S. Nikolopoulos. Cell-swat: modeling and scheduling wavefront computations on the cell broadband engine. In *Conf. Computing Frontiers*, pages 13–22, 2008.

[2] G. Back and D. Nikolopoulos. Application-Specific System Customization on Many-Core Platforms: The VT-ASOS Framework. In *Proc. of the Second Workshop on Software and Tools for Multi-core Systems*, San Jose, CA, March 2007.

[3] F. Blagojevic, M. Curtis-Maury, J. Yeom, S. Schneider, and D. Nikolopoulos. Scheduling Asymmteric Parallelism on a PlayStation3 Cluster. In *Proc. of the 8th International Symposium on Cluster Computing and the Grid*, pages 146–153, Lyon, France, May 2008.

[4] F. Blagojevic, X. Feng, K. Cameron, and D. Nikolopoulos. Modeling Modeling Multigrain Parallelism on Heterogeneous Multi-core Processors: A Case Study of the Cell BE. In *Proc. of the Fourth International Conference on High-Performance Embedded Architectures and Compilers*, pages 38–52, Göteborg, Sweden, January 2008.

[5] F. Blagojevic and D. Nikolopoulos. Exploring Programming Models and Optimizations for the Cell Broadband Engine using RAxML. In *Proc. of the 2006 Virginia Tech High-End Computing Challenge*, 2006. (to appear).

[6] F. Blagojevic, D. Nikolopoulos, A. Stamatakis, and C. Antonopoulos. Dynamic Multigrain Parallelization on the Cell Broadband Engine. In *Proc. of the 2007 ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, pages 90–100, San Jose, CA, March 2007.

[7] F. Blagojevic, A. Stamatakis, C. Antonopoulos, and D. Nikolopoulos. RAxML-Cell: Parallel Phylogenetic Tree Inference on the Cell Broadband Engine. In *Proc. of the 21st International Parallel and Distributed Processing Symposium*, Long Beach, CA, March 2007.

[8] Filip Blagojevic, Dimitrios S. Nikolopoulos, Alexandros Stamatakis, Christos D. Antonopoulos, and Matthew Curtis-Maury. Runtime scheduling of dynamic parallelism on accelerator-based multi-core systems. *Parallel Computing*, 33(10-11):700–719, 2007.

[9] 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.

[10] M. Curtis-Maury, C. Antonopoulos, and D. Nikolopoulos. A Comparison of Online and Offline Strategies for Program Adaptation. In *Proceedings of the 2007 ACM SouthEast Conference*, pages 162–167, Winston-Salem, NC, March 2007.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Matthew Curtis-Maury, Filip Blagojevic, Christos D. Antonopoulos, and Dimitrios S. Nikolopoulos. Prediction-based power-performance adaptation of multithreaded scientific codes. *IEEE Trans. Parallel Distrib. Syst.*, 19(10):1396–1410, 2008.

[17] Matthew Curtis-Maury, Ankur Shah, Filip Blagojevic, Dimitrios S. Nikolopoulos, Bronis R. de Supinski, and Martin Schulz. Prediction models for multi-dimensional power-performance optimization on many cores. In *Proc. of the 17th International Conference on Parallel Architectures and Compilation Techniques*, pages 250–259, Toronto, Canada, 2008.

[18] Matthew Curtis-Maury, Karan Singh, Sally A. McKee, Filip Blagojevic, Dimitrios S. Nikolopoulos, Bronis R. de Supinski, and Martin Schulz. Identifying energy-efficient concurrency levels using machine learning. In *Proc. of the 2007 IEEE International Conference on Cluster Computing*, pages 488–495, Austin, TX, 2007.

[19] 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.

[20] M. Mustafa Rafique, Ali Raza Butt, and Dimitrios S. Nikolopoulos. Dma-based prefetching for i/o-intensive workloads on the cell architecture. In *Proc. of the 5th ACM International Conference on Computing Frontiers*, pages 23–32, 2008.

[21] Alexandros Stamatakis, Filip Blagojevic, Dimitrios S. Nikolopoulos, and Christos D. Antonopoulos. Exploring new search algorithms and hardware for phylogenetics: Raxml meets the ibm cell. *VLSI Signal Processing*, 48(3):271–286, 2007.

[22] 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.