Md Kamruzzaman

(http://cseweb.ucsd.edu/~mkamruzz/)

Department of Computer Science and Engineering
University of California, San Diego
9500 Gilman Drive, La Jolla, CA 92093-0404
Phone: 1-858-205-9350
mkamruzz@cs.ucsd.edu

Objective and Interests

A research and development position that focuses on the emerging challenges for the performance improvements and energy efficiency in current and next generation computing platforms.

Education

University of California, San Diego — Sep 2006 – present
Ph.D. in Computer Science, Advisers: Dean Tullsen and Steven Swanson
Dissertation: "Exploiting Non-Traditional Parallelization Techniques."
Expected Graduation Date: November, 2012

University of California, San Diego — Sep 2006 – Sep 2009
M.Sc. in Computer Science. GPA: 3.88 in 4.0 scale
Selected courses: Principles of Computer Architecture, Parallel Architecture, Parallel Computation, Advanced Compilers, Internet Algorithms, Algorithm Design, Complexity, and Operating Systems

Bangladesh University of Engineering and Technology — Dec 1998 – Feb 2004
B.Sc. in Computer Science and Engineering. GPA: 3.88(overall), 3.95(CSE courses) in 4.0 scale
Selected courses: Microprocessors, Digital System Design, VLSI Design, Computer Networks, Database, Software Engineering, Computer Graphics, Artificial Intelligence, and Simulation and Modeling

Strengths

- Strong background in computer architecture, compiler optimization and parallel programming
- Strong problem solving skill and in-depth knowledge in various algorithms
- Self motivated, quick-learner, innovative and capable of thinking critically

Research Projects

Software Data Spreading (PLDI 2010)

I introduced a novel compiler-directed thread migration scheme that aggregates cache capacity across cores and chips to increase locality for performance improvements and energy efficiency. The technique is software only and applies to any shared memory parallel architecture.

Contributions:

- Demonstrated the potential of thread migration as a first order primitive for compiler optimization.
- Showed that intelligent thread migration reduces as much as 80% of the last level cache misses.
- Designed a compiler algorithm that analyzes the data sharing patterns and automatically transforms a code to exploit data spreading.
- Achieved 17% speedup on average, and as high as 3.3× for some applications.

Inter-core Prefetching (ASPLOS 2011)

I developed the first helper thread prefetching technique that enables local access of the data prefetched remotely in separate cores. The technique uses thread migration and a chunk based synchronization to achieve remote prefetching. Unlike prior techniques, it also works on systems that do not have any shared caches.

Contributions:

- Developed the software infrastructure that supports fast thread migration and allows the effective implementation of inter-core prefetching.
- Investigated the interactions between chunking granularity, number of helper threads, compute density, data access pattern, and data sharing in the context of inter-core prefetching.
- Showed that it is possible to extract more than 2× speedup using just two cores in real systems.
- Demonstrated performance gain of 31 to 63% on average depending on the architecture, and reduction of energy consumption by up to 50%.

Underclocked Software Prefetching (IEEE Micro 2012)

I demonstrated the effectiveness of dynamic frequency scaling coupled with inter-core prefetching to achieve energy efficiency. Inter-core prefetching naturally creates memory intensive prefetching threads that are less sensitive to frequency and provides opportunities for frequency scaling.

Contributions:

- Developed the necessary software infrastructure to implement underclocked software prefetching.
- Demonstrated a theoretical model that shows how dynamic frequency scaling can exploit the decoupling and rate mismatch of prefetching and execution created by inter-core prefetching.
- Analyzed the interaction between number of helper threads, chunk size, different power states, and the latency to change frequencies.
- Showed energy savings of 47% on average for different kernels.

Coalition Threading (PACT 2012)

I introduced coalition threading which means the intelligent combination of traditional and non-traditional parallelism. I did the first investigation about the effectiveness of coalition threading on pre-parallelized code, and found inter-core prefetching as a powerful non-traditional parallelization technique that blends nicely with traditional techniques like data parallelism for better scalability.

Contributions:

- Designed the compiler system based on Rose for the effective implementation of coalition threading on both openmp and pthread code.
- Analyzed the impact of applying different parallelization techniques on the loop level.
- Devised the important loop characteristics that a compiler heuristic can use to intelligently combine traditional parallelization and inter-core prefetching.
- Constructed the linear classifier based heuristic that provides 17% speedup on average on top of the best speedup by traditional parallelization, which is only 0.7% less well than an oracle.

Load Balanced Pipeline Parallelism (In submission)

I proposed and evaluated load balanced pipeline parallelism that naturally extracts locality and load balancing for pipeline parallelism in the loop level to earn better performance and utilization. The technique extracts the maximum parallelism that is possible by any non-speculative technique for all thread counts, and outperforms several well-established techniques.

Contributions:

- Designed the necessary software infrastructure for load balanced pipeline parallelism and for the well-established parallel stage decoupled software pipelining.
- Demonstrated the theoretical analysis that explains the parallelism extracted by load balanced pipeline parallelism and decoupled software pipelining.
- Investigated a thorough evaluation on two state of the art architectures and justified the results predicted by the theoretical model.
- Showed a performance gain of 51% and energy savings of 36% on average over parallel stage decoupled software pipelining for applications that are hard to parallelize.

Publications

- "Software Data Spreading: Leveraging Distributed Caches in Multicore Systems to Improve Single Thread Performance".
  Md Kamruzzaman, Steven Swanson, and Dean Tullsen. In **PLDI 2010**.
- "Inter-core Prefetching for Multicore Processors Using Migrating Helper Threads".
  Md Kamruzzaman, Steven Swanson, and Dean Tullsen. In **ASPLOS 2011**.
- "Underclocked Software Prefetching: More Cores, Less Energy".
  Md Kamruzzaman, Steven Swanson, and Dean Tullsen. In **IEEE Micro 2012.**
- "Coalition Threading: Combining Traditional and Non-Traditional Parallelism to Maximize Scalability".
  Md Kamruzzaman, Steven Swanson, and Dean Tullsen. In **PACT 2012.**
- "Load-Balanced Pipeline Parallelism".
  Md Kamruzzaman, Steven Swanson, and Dean Tullsen. In submission.

Technical Skills

- Languages: Assembly, C, C++, Java, OCaml, Prolog, Python, R
- Programming: Lex, Yacc, MFC, OpenGL, POSIX Threads, OpenMP
- Tools: ROSE Compiler Framework, ATOM, PIN
- Advanced algorithms: Graph algorithms, Linear programming, Combinatorial optimizations, String algorithms, Advanced data structures, AI search techniques, Random number generating algorithms
- Special architectures: Programming Cell Broadband Engine
- Hardware performance counters: Using performance counters for Intel and AMD machines
- Simulators: SMTSIM, NS2

Professional Experience

Teaching Assistant, Algorithms — Winter, 2011
University of California, San Diego
Assisted Professor Andrew Kahng in designing questions, grading and conducting discussion sections.

Intern
Qualcomm Research, San Diego — Jun 2007 – Sep 2007
Supervisor: Gary McGrath
Developed a Google Earth like 3D interface for a Network Planning Tool using OpenGL. The tool performs real time simulation of the network coverage for a set of base stations, and allows 3D exploration of the map. I designed the algorithms and data structures necessary for a smooth real time visualization experience.

Lecturer — Feb 2006 – Sep 2006
CSE Department, Bangladesh University of Engineering and Technology
Conducted lab classes of Digital System Design, Microprocessors, and Computer programming

Lecturer and Programming Contest Coach — Feb 2004 – Feb 2006
American International University-Bangladesh
Instructed Algorithms, Artificial intelligence, Compilers, Concrete mathematics, and Theory of computation. Prepared the teams for ACM ICPC Dhaka and Kanpur site regional contest, and other national contests.

Programming Contest Trainer — 2003 – 2005
Dhaka University, Jahangirnagar University, Independent University, North South University, and International Islamic University Chittagong

Programming Achievements

- Participated in **ACM ICPC World Finals** in 2002 (Hawaii), 2003 (Beverly Hills)
- **Champion** in **ACM ICPC Asia Regional 2001**, Dhaka Site; **3rd** in 2002, Dhaka Site; **4th** in 2003, Kanpur Site, and **5th** in 2003, Dhaka Site
- **Champion** in Tri University Programming Contest, 2002; **Champion** in Regional Warm up Contest, 2002; **Champion** in AIPC-2002 and AIPC-2003; **2nd** in National Computer Programming Contest, 2003; **Champion** in Inter University Programming Contest, 2003

Awards and Honors

- **UCSD CSE Department fellowship in 2006-2007**
- Dean's Award in Bangladesh University of Engineering and Technology in 1st, 3rd, and 4th year
- University Talent Scholarship, Bangladesh University of Engineering and Technology
- University Grants Commission Scholarship, Bangladesh (The top 10 students out of 3500)

Professional Activities

- External reviewer of HPCA 2012 and HPCA 2013; Reviewer of IEEE Micro, 2012
- Judge and Problem setter of ACM ICPC Asia Regional 2005, Dhaka Site; National Computer Programming Contest-2004, Bangladesh
- Member of Elite Problem setters' Panel, ACM Valladolid Online Judge, Spain
- Member of ACM

Reference

Available upon request