Nicholas Edmonds

296 Tyrella Ave.
Mountain View, CA 94043

Tel: (812) 202-0698
nicholas.g.edmonds@gmail.com

Objective

To work on interesting HPC and/or scientific problems that span parallel programming, computer architecture, and language development. I prefer an innovative environment where attention is paid to current research as well as modern languages and software design.

Education

Indiana University 2003-2013
M.S. in Computer Science GPA: 3.97/4.00
Ph.D. in Computer Science GPA: 3.99/4.00

Vanderbilt University 2000-2003
B.S. in Computer Science and Mathematics GPA: 3.60/4.00
Awards: *magna cum laude*, Dean's List, Clayton Kincaid Memorial Scholar, Sprint Foundation Scholar, National Merit Scholar

Experience

Google Mountain View, CA September 2013 - present
Software Engineer III
Worked on the Counter Abuse Team which detects and classifies many types of abuse on Google's publicly-facing products. Abuse detection systems utilizing a combination of blacklist content filters, graph analytics, NLP classifiers, and machine learning classifiers are coordinated using an on-line, event-driven framework providing both synchronous and asynchronous interfaces. This framework enables product-specific policies and flexible configuration to react quickly to threats. I have focused on two primary areas: 1) redesigning and maintaining a system to handle content takedown post-detection, as well as perservation and reporting to appropriate authorities where required by law 2) Designing and maintaining a fuzzy image matching system to store human annotations and use them to derive abuse confidence models.

Indiana University Bloomington, IN August 2004 - August 2013
Research Assistant in the Open Systems Lab (later renamed CREST)
Worked on three primary (interrelated) projects: **1)** the original Parallel Boost Graph Library (BGL), which offers distributed graphs and graph algorithms to exploit BSP parallelism on distributed memory machines while retaining the same interfaces as the (sequential) BGL. The Parallel BGL was incorporated into the Boost collection of software libraries in Q1 2009. **2)** Active Pebbles, a programming and execution model for message driven computation. **3)** The Parallel Boost Graph Library 2.0 which uses Active Pebbles principles and the AM++ active message library to transparently exploit coarse and fine-grained parallelism. Parallel BGL 2.0 uses message-driven computation and declarative parallelism to improve scalability and increase programmer productivity. Work is underway to extend to dynamic graph problems and implement a dynamically tuned runtime.

Google Mountain View, CA January 2008 - August 2008
Software Engineering Intern
Worked in the Google Application Performance Analysis portion of the Platforms team developing application benchmarks for BigTable. Setup a testing environment consisting of dedicated GFS and BigTable cells running on dedicated hardware. Developed a load generator capable of generating synthetic traffic based on profiles of production applications. Tested effects of Linux kernel resource containers on BigTable tablet servers and GFS chunkservers. Transitioned load generator code to BigTable SREs for use in performance testing as well as modeling the effects of adding new applications to existing BigTable cells by synthetically generating traffic prior to application rollout.

Intel Champaign, IL May 2006 - August 2006
Graduate Intern
Engineered developer tools for multi-threaded software development on next-generation multi-core processors. Worked on detecting high-level data-races in multi-threaded code using access interleaving invariants as part of the Thread Checker project. Also looked at methods to better exercise program scheduling in the testing and debugging stages of development through guided scheduling and other methods.

Sandia National Laboratory Livermore, CA May 2004 - March 2006
Graduate Intern
Developed MASS, a Modular Architecture for Sensor Systems in cooperation with two Sandia employees. MASS is both hardware and software extensible and allows developers to customize both the resources available on, and the power consumption of, their sensor nodes. MASS consists of a networking stack and set of user APIs for performing operations with and discovering information about modules in the system, which allows developers to have a fully functional sensor network application by writing only the application specific code. MASS is written in ANSI C and runs on a variety of microprocessors including the Cygnal C8051 and ARM-7. My contribution to the project included everything from device drivers to software architecture design and software development.

Indiana University Bloomington, IN January 2004 - May 2004
Assistant Instructor
Developed curriculum for graduate Introduction to Computer Security course laboratories. Wrote and presented lab activities on topics ranging from different types of attacks to network sniffing and spoofing and intrusion detection, among other topics. Taught four labs per week as well as performing typical associate instructor duties such as grading, etc.

HCA Healthcare Nashville, TN February 2002 - May 2003
IT&S Project Management Intern
Oversaw and coordinated the activities of a 2000 member IT/IS workforce. Coordinated operations of multiple Regional Data Centers spread across three countries. Interacted with projects at every level from project management to implementation. Served in an advisory role on $160m replacement of accounts receivable system. Member of team developing Project Management Center to train Project Managers in order to allow more efficient corporate operations through better resource planning and allocation.

Special Skills

Programming Languages : C++, C, Python, (No)SQL, Ruby, Scheme, LISP

Tools and Abilities : Hybrid parallel programming (e.g., threads + MPI), OpenMP and Cray loop-directives based parallelism, Generic C++ (contributor to Boost), OpenGL Development, Unix Networking, POSIX Threads, MPI, Distributed Systems Development, Embedded Systems Development including JTAG debugging and Device Driver development

Platforms : Cray XMT and XT5, IBM BG/P, various flavors of x86 clusters connected via Myrinet and Infiniband, Sun Niagara, Embedded systems (C8051, TI MSP430), MacOS clusters, and a variety of single-user workstations

Selected Publications

Active Messages as a Spanning Model For Paralle Graph Computation. Nicholas Edmonds. Ph.D. Dissertation. Indiana University, 2013.

Expressing Graph Algorithms Using Generalized Active Messages. Nick Edmonds, Jeremiah Willcock, and Andrew Lumsdaine. International Conference on Supercomputing. Eugene, OR. June 2013.

Expressing Graph Algorithms Using Generalized Active Messages. Nick Edmonds, Jeremiah Willcock, and Andrew Lumsdaine. Principles and Practice of Parallel Programming. Shenzhen, China. February 2013. Poster.

Active Pebbles: Parallel Programming for Data-Driven Applications. Jeremiah Willcock, Torsten Hoefler, Nick Edmonds, and Andrew Lumsdaine. International Conference on Supercomputing. Tuscon, AZ, June 2011.

Active Pebbles: A Programming Model For Highly Parallel Fine-Grained Data-Driven Computations. Jeremiah Willcock, Torsten Hoefler, Nick Edmonds, and Andrew Lumsdaine. Principles and Practice of Parallel Programming. San Antonio, TX, February 2011. Poster.

Scalable Parallel Solution Techniques for Data-Intensive Problems in Distributed Memory. Nicholas Edmonds and Andrew Lumsdaine. SIAM Workshop on Combinatorial Scientific Computing. Darmstadt, Germany. May 2011.

Design of a Large-Scale Hybrid-Parallel Graph Library. Nicholas Edmonds, Jeremiah Willcock, Torsten Hoefler, and Andrew Lumsdaine. International Conference on High Performance Computing, Student Research Symposium. Goa, India. December 2010.

Extensible PGAS Semantics for C++. Nicholas Edmonds, Douglas Gregor, and Andrew Lumsdaine. Conference on Partitioned Global Address Space Programming Model. New York, New York. October 2010.

AM++: *A Generalized Active Message Framework.* Jeremiah Willcock, Torsten Hoefler, Nicholas Edmonds, and Andrew Lumsdaine. Parallel Architectures and Compilation Techniques. Vienna, Austria. September, 2010.

A Space-Efficient Parallel Algorithm for Computing Betweenness Centrality in Distributed Memory. Nick Edmonds, Torsten Hoefler, and Andrew Lumsdaine. International Conference on High Performance Computing. Goa, India. December, 2010.

Single-Source Shortest Paths with the Parallel Boost Graph Library. Nick Edmonds, Alex Breuer, Douglas Gregor, Andrew Lumsdaine. The Ninth DIMACS Implementation Challenge Workshop. Piscataway, NJ. November, 2006.

Method for Module Interaction in a Modular Architecture for Sensor Systems (MASS). Jesse Davis, Doug Stark, and Nicholas Edmonds. In Proceedings International Embedded and Hybrid Systems Conference (IEHSC). Singapore. May 2005.

MASS: Modular Architecture for Sensor Systems. Nicholas Edmonds, Doug Stark, and Jesse Davis. In Proceedings 4th International Workshop on Information Processing in Sensor Networks (IPSN). Los Angeles, California. April, 2005.

Software Application for Modular Sensor Network Node. Jesse Davis, Doug Stark, and Nicholas Edmonds. U.S. Patent Application No. 10/970,684. October 20, 2004.

Grants and Awards

IEEE Technical Committee on Parallel Processing *Best Presentation Award* for *Design of a Large-Scale Hybrid-Parallel Graph Library* in the International Conference on High Performance Computing (HiPC) Student Research Symposium. Goa, India. Dec. 2010.

Best Student Poster Award for *Active Pebbles: A Programming Model For Highly Parallel Fine-Grained Data-Driven Computations* in the 16^{th} ACM SIGPLAN Annual Symposium on Principles and Practices of Parallel Programming. San Antonio, TX. February 2011.

Invited Talks

Version 2.0 of the Parallel Boost Graph Library: Message-driven solutions to data-driven problems. Nicholas Edmonds. SIAM Annual Meeting. Minneapolis, MN. July 2012.

The Parallel Boost Graph Library `spawn`*(Active Pebbles).* Nicholas Edmonds. KDT Mind Meld. Santa Barbara, CA. March 2012.