

Brian Richard Murphy
brm@cs.stanford.edu

Mailing Address

809 Shary Ave
Mountain View, CA 94041
mobile: +1-650-714-8003

Work Address

Intel Labs
2200 Mission College Blvd. SC12-303
Santa Clara, CA 95054
fax: +1-650-745-0982

Research Interests

Program analysis and optimization, interprocedural analysis, parallelizing compilers, programming tools, web site programming architecture.

Education

PH.D. IN COMPUTER SCIENCE, STANFORD UNIVERSITY	Jan, 2002
S.M. IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE, M.I.T.	Sep, 1990
S.B. IN COMPUTER SCIENCE AND ENGINEERING, M.I.T.	Sep, 1990

Dissertation

FRAMEWORKS FOR PRECISE PROGRAM ANALYSIS

Program analysis has become more frequently used for applications such as automatic parallelization, debugging, and program comprehension. These new applications require new methods that provide greater precision at lower computational cost than traditional techniques developed for program analysis in compilers. However, these new methods incur new costs in complexity of understanding and use. To facilitate their use we need sound frameworks which support the construction of analyses based upon them, allowing reuse without constant reinvention for each application.

In the exploration of three difficult program analysis problems, we resolved a number of issues involved in their solution, and developed three frameworks which can help in the future to ease the task of constructing analyses to solve related problems. Some of our frameworks have already been generalized and applied to other analyses beyond those for which they were originally designed.

Automatic parallelizers are an important tool for shared-memory multiprocessors, as they enable sequential code to realize improved performance without extra programmer effort. The SUIF automatic parallelizer proved highly effective on scientific applications. Essential to its success was a new framework for interprocedural analysis which facilitated construction of the many analyses involved in parallelization. The technique of region-based analysis with selective cloning it incorporated allows precise interprocedural analysis results at low cost.

Parallelizer results can be improved by a more systematic treatment of predicates in the program. A general framework for a predicated data-flow analysis provides this treatment, by associating predicates with data-flow values in any analysis. A predicated data-flow analysis can approximate a feasible path analysis, or derive guards for conditionally parallel loops. Several issues prove critical: sound treatment of approximation, avoiding an explosion of predicates with scoping and normalization mechanisms, and reducing analysis costs through a sparse formulation.

The technique of analysis with partial transfer functions makes it possible to build precise and efficient interprocedural analyses for languages with recursion and higher-order

functions, but it is underutilized due to its complexity and inscrutability. We develop a foundational framework for exploiting this promising technique to facilitate more efficient interprocedural analyses. This development should serve as a foundation for continuing work.

Adviser: Professor Monica Lam

Awards and Honors

AT&T Bell Labs Doctoral Scholar	1991–1995
Tau Beta Pi, Eta Kappa Nu honor societies	1989

Research and Academic Experience

Stanford University Stanford, CA
Research Assistant (1990-2001). Performed research in parallelizing compilers and program analysis.

Collaborated on programming, documentation, and release engineering for the SUIF National Compiler Infrastructure project, and its predecessor, the SUIF 1 compiler system.

Administered various computer systems and software for research group. Extensive C++ design and coding, Perl scripting.

Teaching Assistant (WINTER 1993): CS243 Topics in Compilers.

Teaching Assistant (SPRING 1995, 1996): CS343 Advanced Compilation Techniques.

I.B.M. Almaden Research Center San Jose, CA
Functional Languages Group (1989–1990, 1991): Designed and implemented a type inference system for the FL language based upon abstract interpretation. Developed FL-to-C compiler prototype implemented in Common Lisp.

(SUMMER 1988): Contributed to the design of the computer language FL. Completed and maintained a compiler for FL in Lisp through many language changes. Explored issues related to the optimization of FL programs by algebraic rewriting.

Graphics Datastreams Group (SUMMER 1987)

Massachusetts Institute of Technology Cambridge, MA
Teaching Assistant (SPRING 1989): 6.001 Structure and Interpretation of Computer Programs.

Teaching Assistant (SPRING 1989): 6.035 Compiler Design.

Professional Experience

Intel Corporation Santa Clara, CA
Just-in-time Researcher (Feb 2000-): Developed high-level analyses and optimizations for a multi-language, multi-target just-in-time compiler. Personally originated at least one novel and patentable technique. Concurrently assisted several short-term special-purpose projects, e.g.: to improve existing JIT performance, involving both analysis and low-level instrumentation and performance optimization.

Eliren Intl., Inc.

Menlo Park, CA and Beijing, China

Chief Technical Officer (2000-2001), Board Director (2001): Supervised technical development and operations for internet activities (content/community web site, online games) of this multimedia startup company, which produces online, print, and TV content for contemporary Chinese women. Supervised negotiations with providers. Directed technical staff and programmers, as well as personally handling much computer system, web site, and database configuration, administration, integration, and tuning. In the early days, experimented with candidate systems for web site architecture and content update. Advised CEO and founder about technical issues.

M.I.T. Project Athena – SYSTEMS PROGRAMMER (1986–1987) Cambridge, MA

Datamap – PROGRAMMING CONSULTANT (1984–1985) Upper Darby, PA

Publications

JOURNAL ARTICLES

M. W. Hall, S. P. Amarasinghe, B. R. Murphy, S. Liao, and M. S. Lam, Interprocedural Parallelization Analysis in SUIF. To appear in *ACM Transactions on Programming Languages and Systems*.

J. Anderson, M. Hall, S. Amarasinghe, B. Murphy, S. Liao, E. Bugnion, and M. Lam, Achieving High Performance on Digital AlphaServers with the SUIF Compiler. *Digital Technical Journal*, Vol. 10 No. 1, 1998, pages 71-80.

M. W. Hall, J. M. Anderson, S. P. Amarasinghe, B. R. Murphy, S. Liao, E. Bugnion, M. S. Lam. Maximizing Multiprocessor Performance with the SUIF Compiler. *IEEE Computer*, December, 1996.

S. P. Amarasinghe, J. M. Anderson, C. S. Wilson, S.-W. Liao, B. R. Murphy, R. S. French, M. W. Hall, and M. S. Lam, Multiprocessors from a Software Perspective. *IEEE Micro*, 16(3), June 1996.

CONFERENCE PAPERS

S. Moon, M. W. Hall, B. R. Murphy. Predicated Array Data-Flow Analysis for Run-Time Parallelization. *International Conference on Supercomputing*, July, 1998.

M. W. Hall, S. P. Amarasinghe, B. R. Murphy, S. Liao, and M. S. Lam, Detecting Coarse-Grain Parallelism Using an Interprocedural Parallelizing Compiler. *Proceedings of Supercomputing '95*, December, 1995.

M. W. Hall, B. R. Murphy, and S. Amarasinghe. Interprocedural Parallelization Analysis: A Case Study. In *Proceedings of the Seventh SIAM Conference on Parallel Processing for Scientific Computing*, February, 1995.

A. Aiken and B. R. Murphy. Implementing Regular Tree Expressions. In *5th ACM Symposium on Functional Programming Languages and Computer Architectures*, August, 1991.

A. Aiken and B. R. Murphy. Static Type Inference in a Dynamically Typed Language. In *18th ACM Symposium on Principles of Programming Languages*, Orlando, FL, 1991.

SELECTED WORKSHOP PAPERS

B. R. Murphy, M. S. Lam. Program analysis using partial transfer functions. *Workshop on Partial Evaluation and Program Manipulation*, Jan, 2000.

M. W. Hall, B. R. Murphy, S. P. Amarasinghe, S. Liao, and M. S. Lam. Interprocedural Analysis for Parallelization. *8th International Workshop on Languages and Compilers for Parallel Computing (LCPC95)*. Springer-Verlag, August, 1995.

MISC

B. R. Murphy, *A Type Inference System for FL*. Master's thesis, M.I.T., 1990.