RICE UNIVERSITY

Department of Computer Science

Mail Stop 132
6100 S. Main St.
Houston, Texas 77005-1892
Phone: 713-285-5977
Fax: 713-285-5930
E-mail: vardi@cs.rice.edu

Moshe Y. Vardi
George Professor
and Department Chair

Oct. 16, 2000

Professor Anita Jones
Department of Computer Science
Thornton Hall
University of Virginia
Charlottesville, VA 22903, USA

Dear Anita:

It is my pleasure to nominate Professor Kenneth W. Kennedy for the ACM Turing Award for contributions to the theory and practice of compiler optimization and for leadership in the development of software for parallel computation.

Theory: Kennedy is a leader in the area of compile-time analysis and improvement of programs, often termed "code optimization". His contributions to this area span a period of twenty-five years; equally important, they cover a diverse set of problems. The complete list of his contributions is too long to be included here, but a brief selection illustrates the importance of his work.

In the arena of data-flow analysis, Kennedy and his students have worked on both intraprocedural analysis and interprocedural analysis. He showed how to solve backward-flow problems with the interval algorithm, introduced techniques to use both graph grammars and attribute grammars to solve data-flow problems, and published the first realistic comparison of two different solution techniques. His work on interprocedural analysis techniques led to faster algorithms for the classical problems and to adoption of this technology in commercial compilers.

In dependence analysis, a key technology that underpins automatic detection of parallelism, Kennedy and his students have published many discoveries. These papers range from explanations of the work of others through simple and effective new dependence tests to studies of the coverage of various dependence tests. Among the many companies that have used their work in commercial systems are IBM, Convex, Cray Research, Digital Equipment, Masspar, Thinking Machines, the Portland Group, Pacific-Sierra Research, and Applied Parallel Research.

Kennedy's work on automatic transformations to expose parallelism is well known. It includes algorithms that target vector machines ("vectorization"), algorithms that target parallel machines ("parallelization"), approaches to the problem of distributing both data and computation, and work on optimizing communications and I/O. Many of these techniques are employed by commercial compilers.

In attribute grammars, Kennedy and his student Scott Warren published a seminal paper describing the first implementation technique to produce efficient translators for a reasonably general class of grammars. With other students Kennedy explored alternative implementation techniques and the application of attribute grammars to program optimization. One of his students (Farrow) has become a leading authority on the practical application of attribute grammars and has built a commercial AG system now in use by many companies.

Perhaps, less well known is Kennedy's work on improving code for uniprocessors. He played a central role in the early work on dead-code elimination and operator-strength reduction; his algorithms are in widespread use today. In his quest for compiler-based techniques to manage the memory hierarchy, he has worked on register allocation for both scalar and vector registers, on techniques for exposing reuse of values to the compiler, on prefetching for data caches, and on loop blocking to improve both cache and register locality.

Implementation: The implementation and design work done in Kennedy's group on programming environments has helped to change the relationship between tools, compilers, and code written by programmers. The Rn environment pioneered the idea that the information-rich setting of a programming environment was an ideal setting for implementing aggressive compilation techniques. The ParaScope environment carried these ideas over into the realm of automatic detection of parallelism, a tradition carried on by their current project, the D System. These systems have influenced a generation of work in both academia and industry.

1. Kennedy and his collaborators constructed the PFC Automatic Vectorization and Parallelization System, one of the two most important research vectorizers in the early 1980's. It was used as the prototype for the IBM and Convex vectorizing compilers and influenced the design of the many commercial products, including those from Alliant, Stardent, Applied Parallel Research, Cray, Digital Equipment Corporation, Fujitsu, Hitachi, NEC, Maspar, Pacific-Sierra Research, Thinking Machines, and Univac.

2. Kennedy led the development of the Rn/Parascope Programming Environment for scientific computing that pioneered the use of the interprocedural analysis and optimization in compiling for high performance on both uni- and multi-processor computer systems. ParaScope served as the model for the Convex applications compiler and several other systems under development by major computer manufacturers.

3. Kennedy led the design and implementation team for Fortran D, an extended version of Fortran that supports machine-independent data parallel programming. Fortran D was one of the early data-parallel languages, and its syntax, semantics, and compilation strategy have been extremely influential. The syntax and semantics were a major contribution to the High Performance Fortran Forum, which standardized data-parallel Fortran. The compilation techniques pioneered in Fortran D include computation partitioning by generalizations of the owner-computes rule, message vectorization and aggregation, data-flow placement of communication and synchronization primitives, and automatic blocking of messages for pipelined computations. These techniques form the basis of compilers and tools now available from the Portland Group, Applied Parallel Research, IBM, Digital, and other vendors.

Leadership:

1. Kennedy founded the Center for Research on parallel Computation (CRPC), one of the first 11 National Science Foundation Science and Technology Centers, and the only one devoted to high performance computing. The goal of the CRPC was to take a leadership role in developing the science and technology infrastructure needed to make parallel computational truly usable. The Center was a consortium of seven institutions: Rice University, Caltech, Argonne National Laboratory, Los Alamos National Laboratory, Syracuse University, the University of Tennessee, and the University of Texas. Among the accomplishments of this center were the development of Fortran D, a direct predecessor to High Performance Fortran (the result of a standardization effort chaired by Kennedy), the development of interprocedural compilation technology for Fortran on shared memory machines, the development of new methods of compiler optimization of memory hierarchy usage, Fortran M and CC++ (new languages for task parallelism), development of PVM (a portable message-passing system supporting heterogeneous machines), development of the automatic differentiation system ADIFOR, new parallel algorithms for numerical optimization problems, and initial prototypes of SCALAPACK (a scalable linear-algebra package). CRPC helped to develop a national agenda for parallel-computation research through a series of workshops on special topics, and it pioneered new educational programs including degree programs in computational science and engineering, undergraduate research programs, and programs to increase the participation of minorities in science, engineering, and mathematics. Finally, through many outreach programs, CRPC served as a clearinghouse for information and ideas on parallel computation.

   To accomplish the goals, Kennedy assembled a geographically-distributed team of internationally-recognized researchers on parallel computation and focused them on the goal of overcoming the barriers to machine-independent parallel programming. The Center involved nearly 100 Ph.D researchers and at least that many graduate students. Kennedy was able to accomplish this due in no small measure to his outstanding personal qualities of strong leadership ability and deep technical expertise. The results produced by the CRPC are used by industries all over the nation, including all the major parallel computer manufacturers.

2. Kennedy founded and chaired the High Performance Fortran Forum, which produced the specification for for High Performance Fortran (HPF). The Forum consisted of representatives from academia, industry and government working to produce a machine-independent language for parallel programming. Within a year and a half, they had created the first widely-accepted, machine-independent, data-parallel variant of Fortran. Much of HPF was taken directly from Kennedy's Fortran D research, which had proved that such a language was possible. Commercial compilers for HPF are now available from PGI, IBM, Digital, and other vendors. Although HPF itself is an informal standard, many of its innovative features have been adopted by ANSI and ISO into the official Fortran 95 and Fortran 2000 standards. Perhaps equally important, the HPF Forum process has been widely copied in other informal standards efforts, resulting in timely standards for message-passing and shared-memory programming.

3. Kennedy has been an active member of the technical community. He serves on the editorial boards of International Journal of Computer Mathematics, Journal of Parallel and Distributed Computing, Concurrency - Practice and Experience, and ACM Transactions on Software Engineering and Methodology. He was the program chair of Supercomputing '91, he has been a member of National Academy of Engineering since 1990, he serves on the President's Council of Advisors on Science and Technology since 1991, he was a member of the NSF Advisory Committee for Computer Research during 1984-8 and chaired that committee during 1985-7, he was a member of the Board of the Computing Research Association during 1985-91, and he is a member of the National Research Council's Computer Science and Telecommunications Board since 1991.

4. Together with Bill Joy, Kennedy chaired the President's Information Technology Advisory Committee (PITAC). The charge of the Committee, established in February 1997,

   > "to provide guidance and advice on all areas of high performance computing, communications and information technologies. The Committee members bring a broad range of expertise and interests from business and universities. They will provide valuable guidance to the administration's efforts to accelerate development and adoption of information technologies that will be vital for American prosperity in the 21st century."

   To carry out its charge, Joy and Kennedy lead PITAC through a series of public meetings in which they studied numerous issues related to their charge, including: high-end computing, scalable infrastructure, software, funding modes, research management, and socio-economic and workforce issues. In August 1998, PITAC issued an interim report, which was followed by a final report on Feb. 24, 1999.

   The report concludes that information technology will be one of the key factors driving progress in the 21st century, and a vigorous information technology research and development effort is essential for achieving America's 21st century aspirations. At the same time, federal support for research in information technology is seriously inadequate. To address this problem, the Committee recommended that the federal government increase its support for information technology research by $1.37 billion by FY 2004.

   The Administration has already responded to the report by proposing a dramatic $366 million increase in next year's computing research budget. As an example, the Administration's proposal calls for the NSF CISE research budget to increase next year by roughly 50%, from roughly $200 million to roughly $300 million.

   Our community owes an enormous debt to Bill Joy, Ken Kennedy, and the rest of the PITAC, for making a compelling case regarding the crucial importance of information technology research to the future of this country.

For his contributions in theory, implementation, and leadership, I cannot think of a better candidate for the ACM Turing Award then Kenneth W. Kennedy.

Yours sincerely,

Moshe Y. Vardi