Descartes: Dynamically Emergent Specialization via Contextually Adaptive Re-compilation using Task Execution Spectra for Profile-Driven Polyvariant On-Line Partial Evaluation

by

Michael Ross Blair

Master of Science, Computer Science & Engineering, M.I.T. (1990)
Bachelor of Science, Computer Science & Electrical Engineering, M.I.T. (1986)

Submitted to the
Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements
for the degree of

Doctor of Science
in the field of
Computer Science

at the

MASSACHVSETTS INSTITVTE OF TECHNOLOGY

February 2008

Signature of Author . . . .
Department of Electrical Engineering and Computer Science
November 30, 2007

Certified by . . . .
Thomas F. Knight, Jr.
Senior Research Scientist
Thesis Supervisor

Accepted by . . . .
Terry P. Orlando
Chairman, Departmental Committee on Graduate Students

[This page intentionally left very nearly blank.]

Descartes: Dynamically Emergent Specialization via Contextually Adaptive Re-compilation using Task Execution Spectra for Profile-Driven Polyvariant On-Line Partial Evaluation

by
Michael Ross Blair

Submitted to the
Department of Electrical Engineering and Computer Science
on November 30, 2007
in partial fulfillment of the requirements for the degree of
Doctor of Science

Abstract

Dynamically adaptive, profile-driven, spectral program specialization is an effective and practical technique for building large, sophisticated, efficient software systems from high-level, abstract, modular programs using type-driven polyvariant on-line partial evaluation grounded in statistical inference.

In profiling-based methods of program optimization, one gathers dynamic information from sample program runs in order to focus optimization efforts on those code fragments that stand to benefit most. This ordinarily requires a programmer-crafted suite of program inputs from which dynamic statistical data about a program's behavior can be deduced. Moreover, this profiling is usually done only once and, thereafter, the optimized code is frozen.

Unfortunately, improving a program's *average* performance on some *typical* inputs does not ensure *optimal* performance on any *specific* input. Worse, it is often difficult to ascertain reliably what constitutes a typical input for highly general programs. For example: What is a typical circuit for a circuit simulator? What constitutes a typical program for a compiler or interpreter or operating system? The answers can vary wildly among different sites and program users. They can even vary dramatically from day to day for a user with changing needs and goals.

What is needed is a mechanism for programs to adapt themselves automatically to their diverse and changing application environments.

To that end, the DESCARTES system marries a set of run-time profiling tools to a profile-driven kernel-level source-to-source program specializer and an optimizing compiler for the MIT C SCHEME dialect of LISP. This yields on-the-fly identification and specialization of performance-critical high-level, abstract program modules, selectively reducing them to streamlined equivalent code in a robust, disciplined, judicious and efficient manner.

The principal original contributions of this work (excluding tools), include:

1) *Spectral Specialization* - source-to-source program specialization with respect to *context spectra*, constituting dynamically adaptive, profile-driven, polyvariant on-line partial evaluation grounded in statistical inference; where...

2) *Context Spectra* - robust, compact and efficient representations of the point estimates of dynamic execution contexts to profile each procedure's dynamic data via weighted distributions over generalized structural types; with...

3) *Statistical Inference as Feedback* - the use of statistically inferred data as a rigorous feedback mechanism to focus, drive and throttle ongoing program optimization efforts while simultaneously regulating overhead through self-tuned parameters, thereby avoiding traditional architectural *ad hoc* limits (a.k.a. "magic constants").

Put simply, DESCARTES performs source-level code optimization by using dynamic profile information, all driven by statistical inference. The novelty lies in the nature of the optimization performed, what data is used to do it, and how the overhead of this ongoing process is managed. The original contributions listed above stake specific claims in each of these areas.

Experiments using this prototype have demonstrated that a handful of carefully chosen principled specializations can dramatically improve code speed with only moderate profiling overhead and modest code-space growth. For example, profile-driven automatic optimization of a program to analyze genetic pedigrees (as Bayesian belief nets) has produced a speedup factor of 5 over the original highly optimized code produced by the MIT C SCHEME compiler. A thermal diffusion simulation was sped up by a factor of 8, and the tree- recursive Fibonacci program was sped up by a factor of 2. These were achieved with less than a 2-fold increase in code size and below 5% overhead in execution time for profiling.

This dissertation explores the design space for effective profile-driven adaptive program specialization, under the practical constraint that the overall system must be efficient with respect to the dynamic overhead.

Specifically, it details how the DESCARTES prototype decides what code to optimize, how to optimize it, to what degree and at what cost. It concludes by considering how this prototype system could be used as the foundation for a fully automated, continual, on-the-fly, dynamically adaptive program optimization system.

Thesis Supervisor: Thomas F. Knight, Jr.
Title: Senior Research Scientist

[This page intentionally left very nearly blank.]

[This page intentionally left very nearly blank.]

Part O

Prologue

Part I

Examples

[This page intentionally left very nearly blank.]

Part II

Details

Part III

Conclusion

[This page intentionally left very nearly blank.]

Part IV

Appendices