International Journal of Parallel Programming, Vol. 29, No. 5, October 2001 (© 2001)

Guest Editor's Introduction

It is my pleasure to introduce this special issue on compiler optimization of nested loops. Loop transformation is an important and well-developed field and it is thus very interesting to see significant new results in this subject area.

The papers appearing in this special issue were chosen from among the papers of the *14th Annual ACM/IEEE International Conference on Supercomputing*. Three outstanding papers from the conference were selected. The authors were invited to revise the papers to meet journal quality standards and the result is a distinguished collection of three papers. Let us now introduce these papers.

"Fast Greedy Weighted Fusion," by Ken Kennedy presents a new algorithm for loop fusion. Loop fusion is important for memory hierarchy optimization to increase data reuse. Unfortunately, the problem of fusion for data reuse is known to be NP-hard. The new algorithm uses weighted greedy fusion to produce high-quality solutions fast, where the "weight" is determined by the amount of reuse. It is shown that the complexity of fusion remains at $O(V(E+V))$ if the re-weighting operation takes no more than $O(V)$ time.

In "Synthesizing Transformation for Locality Enhancement of Imperfectly-Nested Loops," Nawaaz Ahmed, Nikolay Mateev, and Keshav Pingali address an important problem of loop transformations for imperfectly nested loops. Loop transformations and tiling are very effective for enhancing locality, but are only directly applicable to perfectly-nested loops. The authors propose to embed the iteration space of each statement into a "product space." The product space is a special iteration space which can be viewed as a perfectly nested loop nest. The product space can itself be transformed to increase locality further, after which the fully permutable loops can be tiled. Experimental results confirm the effectiveness of this approach.

The third paper, Vivek Sarkar's "Optimized Unrolling of Nested Loops," introduces a new approach to automatically selecting unroll factors and generating compact code for the selected factors. Loop

0855-7485/01/1000-0461$19.50/0 © 2001 Plenum Publishing Corporation

unrolling is widely used and the paper makes several important contributions to unrolling. They include a more detailed cost model, a new code generation algorithm producing more compact code than past work, and a new algorithm for efficient enumeration of feasible unroll vectors. The experimental results show an average speedup of $1.08\times$ on seven SPEC95fp programs and a speedup of $2.2\times$ on matrix multiply. Even larger performance improvement can be expected on processors with a larger register file and a higher degree of ILP.

I would like to thank all the authors for their outstanding contributions to this special issue. Their work has made this issue an important contribution to the field of compiler optimization and loop transformation.

Alex Veidenbaum, U.C. Irvine
Guest Editor