Automatic permutation for arbitrary static access Patterns

Joar Sohl, Jian Wang, Andréas Karlsson, Dake Liu

ABSTRACT:

A significant portion of the execution time on current SIMD and VLIW processors is spent on data access rather than instructions that perform actual computations. The ePUMA1 architecture provides features that allow arbitrary data elements to be accessed in parallel as long as the elements reside in different memory banks. Using permutation to move data elements that are accessed in parallel, the overhead from memory access can be greatly reduced; and, in many cases completely removed. This paper presents a practical method for automatic permutation based on Integer Linear Programming (ILP). No assumptions are made about the structure of the access patterns other than their static nature. Methods for speeding up the solution time for periodic access patterns and reusing existing solutions are also presented. Benchmarks for e.g. FFTs show speedups of up to 3.4 when using permutation compared to regular implementations.