Polynomial-time Algorithms for Enforcing Sequential Consistency in SPMD Programs with Arrays

Wei-Yu Chen Arvind Krishnamurthy Katherine Yelick

To appear at the *16th Workshop on Languages and Compilers for Parallel Computing (LCPC03)*, College Station, TX, 2-4 October 2003

Abstract

The simplest semantics for parallel shared memory programs is sequential consistency, in which memory operations appear to take place in the order specified by the program. But many compiler optimizations and some hardware features explicitly reorder memory operations or make use of overlapping memory operations which may admit the possibility of runtime reordering. To ensure sequential consistency while allowing for these optimizations, traditional data dependence analysis is augmented with a parallel analysis called cycle detection. In this paper, we present new algorithms to enforce sequential consistency for the special case of the Single Program Multiple Data (SPMD) model of parallelism. First, we present a new algorithm for the basic cycle detection problem, which lowers the running time from $O(n^3)$ to $O(n^2)$. Next, we present three polynomial-time methods that more accurately support programs with array accesses. These results are a step toward making sequentially consistent shared memory programming a practical model across a wide range of languages and hardware platforms.