

SYNPLEX, a task-parallel scheme for the revised simplex method

Julian Hall, University of Edinburgh

The simplex method is frequently the most efficient method of solving general large sparse linear programming (LP) problems, particularly when a family of related problems is to be solved. Despite this, there has been no parallelisation of the simplex method that offers significantly improved performance over a good serial implementation of the revised simplex method.

Each iteration of the revised simplex method consists of several computational components. Of the four major components, one is a matrix-vector product in which data parallelism is readily exploited. The other three are the solution of two large sparse systems of linear equations and, periodically, the factorisation of a large asymmetric sparse matrix.

When the parallelisation of the simplex method was first and most widely considered, parallel factorisation and solution methods for sparse unsymmetric linear systems were in their infancy. As a consequence, the three genuine attempts to exploit parallelism anything like fully in the revised simplex method for general large sparse LP problems have exploited task parallelism. The scope for exploiting task parallelism was achieved by implementing variants of the revised simplex method that require the solution of sets of independent linear systems, with each particular system being solved on a single processor. Although the matrix factorisation was also performed serially, parallelism was exploited by overlapping it with simplex iterations. The success of these attempts was limited, the best performance (relative to an efficient sequential revised simplex solver) being speed-ups of between 2 and 5 when using up to 12 processors. There were several reasons for this limited success. Firstly, the particular variants that were implemented typically require a greater number of iterations to solve a given LP problem than those used by efficient sequential solvers. A further consequence of the variants that were implemented is that, for some of the linear systems, the solutions obtained are never used. Other factors were numerical instability (when the matrix factorisation is overlapped with certain computational components) and the communication overheads of the distributed memory multiprocessors upon which the schemes were implemented.

This presentation will describe SYNPLEX, a task-parallel scheme for the revised simplex method. SYNPLEX eliminates the numerical instability of the schemes above and addresses the issues of both the increase in the number of iterations required to solve the problem and the wasted work. With the target architecture being a shared memory (Sun Fire E15k) multiprocessor, data communication overheads are expected to be reduced. An initial implementation indicated the need for careful data distribution and these issues will be discussed in this presentation. A revised implementation incorporating more sophisticated data management offers greater scope for success on this significant challenge in computational optimization.