The Journal of Supercomputing, 2, 55–79 (1988)

Maze Routing on a Hypercube Multicomputer

YOUNGJU WON and SARTAJ SAHNI
Computer Science Department, 136 Lind Hall, University of Minnesota, Minneapolis, MN 55455

(Received May 1987; final version accepted March 1988)

Abstract. The implementation of Lee's maze routing algorithm on an MIMD hypercube multiprocessor computer can follow several plausible mappings and synchronization strategies. These are evaluated experimentally on an NCUBE/7 hypercube computer with 64 processors. Different grid partitioning and mapping strategies result in a different balance between computation and communication time. The total routing time is significantly impacted by the synchronization and termination detection scheme used. Further, by rearranging the computation, it is possible to overlap much of the interprocessor communication with the computation and realize a significant reduction in the overall run time. By choosing the right partitioning and synchronization scheme and by overlapping computation and communication, a good speedup is obtained on large routing grids.

1. Introduction

Lee's maze router is a popular wire routing algorithm. In the single layer case, the wiring surface is represented by a grid, as shown in Figure 1. Some of the cells are blocked (shown as shaded cells in Figure 1a) while others are available for routing. There are two specially designated cells: s, the source cell, and t, the target. These cells are the end points of a wire that is to be routed using only those cells that are not blocked. The wire begins at s, passes through available cells, and finally reaches t. A wire can be routed from one cell to the next by crossing a cell boundary (but not through a cell corner). One may assume that cells s and t are not blocked. The objective is to find a shortest route from s to t. In this paper, we consider the single layer case only.

Lee's algorithm for maze routing is a three-phase algorithm. These phases are *front wave expansion, path recovery,* and *sweeping*. During *front wave expansion,* a breadth-first search beginning at s is performed. Cells that are one unit from s are labeled, then those two units from s are labeled, then those three units from s are labeled, and so on. This labeling continues until the target cell t is reached. Blocked cells are not labeled during *front wave expansion.* Figure 1b shows the effect of the *front wave expansion* phase on the initial configuration. We use the four labels $\rightarrow$, $\leftarrow$, $\downarrow$, and $\uparrow$ to point to the cell from which we reached the current cell. Thus, all four cells adjacent to s have a label that points to s.

If the front wave expansion reaches the target cell t, the *path recovery* phase begins. This involves backtracking from t to s by simply following the arrow labels from t to s (see Figure 1b). Now the wire path has been identified. Before the next wire can be