Communication Based Mapping of Graph Algorithms to Meshes

Subodh Kumar Jan Prins

January 31, 1994

1 Introduction

The simplified structure of PRAMs make writing parallel programs and reasoning about their correctness very easy. The caveat is that most real life parallel machines are far from PRAMs. The costs of operations assumed in PRAMs are unrealistic. In particular, the ability to read any memory location translates to arbitrary routing, and possibly collisions, on most machines. These generally have high cost and lead to disappointing performance. On the other hand, for specific communication topology certain patterns of communications can be much less costly. For example, on a mesh connected machine, like the MasPar MP-1/MP-2, communication within a row or a column can be done efficiently.

We are investigating the implementation of graph algorithms on meshes. While there exist a variety of PRAM algorithms for these problems, truth is that many graph algorithms do not really require arbitrary routing. We are experimenting with ways to decompose a problem such that any information that a processors needs can be found either in its row or in its column. In general this could result in a duplication of information. The idea is to minimize this duplication while keeping the communication restricted within a row or a column.

Thus, rather than reinventing the wheel, our strategy is to start with an efficient PRAM algorithm and map it to the mesh to minimize the communication time and thus speed up the total execution.

2 Proposed Implementation Project

We propose to demonstrate new implementations of algorithms to find the *Connected Components* of sparse and dense graphs. There are three distinct parts of the problem:

1. Map the problem onto the machine so as to restrict all communications to be within a row or a column.

2. When the number of nodes (or edges) in the graph is larger than the number of processors available, distribute the work to processors in a load balanced manner.

3. Keep the solution work efficient for sparse instances of the problem.

3 Evaluation Strategy

We hope to demonstrate our algorithms by implementing them (in MPL) on an 8K MasPar MP-1 with 64 Kbytes of memory per processor. The *Connected Components* problem requires a large number of what looks like arbitrary routings. We plan to show the effectiveness of row/column communication by doing the following:

- Generate various sizes and categories of graphs (with different edge distributions). We hope that the committee will make some benchmarks available.

- Find the connected components using both cyclic and hierarchical decomposition for each class of graphs.

- Tabulate the running times. Analyze the tradeoff between wasted memory (this means we would be able to solve only smaller problems than what the memory allows) and speedup gained.

- Analyze the load balancing characteristics of the communication based decomposition.