Project Proposal

Xi He, Jai Dayal and Kevin Pinto

October 3, 2010

1 Problem Description

We would like to explore the graph coloring problem in our team project. The graph coloring problem deals with assigning colors to the vertices of a graph so that adjacent vertices do not get the same color. The primary objective is to minimize the number of colors used. However, coloring a general graph with the minimum of colors is known to be an NP-hard problem[1], thus we can only rely upon heuristics to obtain a usable solution.

2 Related Work

The following are the related papers we are going to look into.

A. H. Gebremedhin and F. Manne, "Scalable parallel graph coloring algorithms" in Concurrency Practice and Experience, 2000, pages 1131–1146.
URL - http://www.cs.purdue.edu/homes/agebreme/publications/cpe-color.pdf

M. Kubale and L. Kuszner, "A better practical algorithm for distributed graph coloring" in Parallel Computing in Electrical Engineering, 2002, pages 22–25.
URL - http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1115204&isnumber=24596

E. Salari and K. Eshghi, "An ACO algorithm for graph coloring problem" in Int. J. Contemp. Math. Sciences, 2008, pages 293–304.
URL - http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.8226&rep=rep1&type=pdf

3 Algorithms

3.1 Sequential Algorithm

Our sequential algorithm will be based on a general greedy framework: a vertex is selected according to some predefined criterion and then colored with the smallest valid color. The selection and coloring continues until all the vertices in the graph are colored.

3.2 Parallel Algorithm

Our parallel algorithm will divide the vertex set of the graph into p successive blocks of equal size, and then colors every block in parallel. Here p is the number of processors.

4 Performance Metrics

Performance metrics we are going to measure include

- $Speedup(N, K) = \frac{T_{seq}(N,1)}{T_{par}(N,K)}$
- $Efficiency(N, K) = \frac{Speedup(N,K)}{K}$
- $Sizeup(T, K) = \frac{N_{par}(T,K)}{N_{seq}(T,1)}$
- $Speedup\ Efficiency(N, K) = \frac{Sizeup(T,K)}{K}$
- $EDSF(N, K) = \frac{K \cdot T_{par}(N,K) - T_{par}(N,1)}{K \cdot T_{par}(N,1) - T_{par}(N,1)}$

Here N is the number of vertices. K is the number of the processors. $T_{seq}(N, 1)$ and $T_{par}(N, K)$ represent the sequential and parallel running time for the problem with N vertices on K processors, and $N_{seq}(T, K)$ and $N_{par}(T, K)$ indicate the number of vertices in the problem which can be sequentially or in parallel solved within a running time of T on K processors.

References

[1] M. Garey and D. Johnson. *Computers and intractability.* Freeman San Francisco, 1979.