Analysis of Parallel Incremental/Decremental Graph Colouring on GPU

A Project Report

submitted by

MOHAMMED SHAMIL

in partial fulfilment of the requirements

for the award of the degree of

MASTER OF TECHNOLOGY

under the guidance of

Dr. Rupesh Nasre

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2016

THESIS CERTIFICATE

This is to certify that the thesis titled **Analysis of Parallel Incremental/Decremental Graph Colouring on GPU**, submitted by **Mohammed Shamil**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Rupesh Nasre
Research Guide
Assistant Professor
Dept. of Computer Science and Engineering
IIT Madras, 600 036

Place: Chennai

Date: 11 May, 2016

ACKNOWLEDGEMENTS

Thanks to all those who made TEX and LATEX what it is today.

ABSTRACT

KEYWORDS: Colour Quality; Compressed Sparse Row Representation; Decremental Graph Colouring; GPGPU; Graph Colouring; Incremental Graph Colouring; NP-hard; nVIDIA Cuda; Parallel Computing; Parallel Graph Algorithms; Vertex Colouring.

Graphs are a well studied and widely used data structure in the field of algorithms, programming and computing. There are a lot of interesting applications of graphs and various algorithms are built on top of the graph data structure. One of the most famous and well studied graph problems is that of graph colouring. There are a lot of different versions of graph colouring problem of which the most common ones are that of vertex colouring and edge colouring. The problem is seemingly simple, to allocate a colour to every vertex/edge of a graph so that adjacent vertices/edges don't share the same colour minimizing the number of colours used. Graph colouring is a very important and yet very challenging graph problem with ongoing active research. Graph colouring finds application in a varied range of problems including various scheduling problems like job scheduling on distributed computing systems, register allocation in compilers, pattern matching problems and solving Sudoku boards.

Though the problem is seemingly simple, it is computationally hard. The graph colouring problem we are exploring in this work, that of vertex colouring, is an NP-hard problem. The sequential approaches like greedy colouring are simply not fast enough whereas advanced approximate/randomized solutions either produce colourings of bad colour quality or aren't fast enough. Thus came the parallel approaches to Graph Colouring. Most of the parallel versions of Graph Colouring algorithms were designed with either multi-core CPUs or heavy duty super computers in mind. With the advent of General-Purpose computing on GPUs (GPGPU), we have access to cheap heavy multi-threaded parallel computing power. Our work is based on parallel computing on nVidia GPUs using Cuda programming language.

We explore different parallel graph colouring algorithms on nVIdia GPUs in this work and try to adapt them to support addition of edges, called incremental graph colouring, and deletion of edges, called decremental graph colouring. In the first section, we explore different parallel graph algorithms and adapt a couple of them, one based on *speculation* and *conflict resolution* and the other on *Vertex Independent Sets*, to work on nVidia GPUs. In the following sections, we adapt the GPU parallel colouring algorithm to support additions and deletions of edges. In the incremental part, we explore different methods to maximize parallelization while colouring newly added edges and use propagation to improve overall colour quality. In the decremental part, we explore different options to either process the vertices, on which the deleted edges were incident, on the go or to process them together and use propagation to propagate the information across the graph improving the colour quality.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

IITM	Indian Institute of Technology, Madras
RTFM	Read the Fine Manual
GPU	Graphics Processing Unit
GPGPU	General-Purpose computing on Graphics Processing Units
CSR	Compressed Sparse Row

CHAPTER 1

INTRODUCTION

1.1 Graphs and Graph Algorithms

Graphs are really important mathematical concepts and in the area of computing, their various forms are widely used as data structures to aid various algorithms. Graphs are commonly used to denote relations between different entities and hence is a very important and integral part of many algorithms. On a practical level, we deal with graphs in the order of billions of nodes and edges on a daily basis. Especially with the advent of social networks and big data, a lot of active research is ongoing in the analysis and understanding of large graphs.

Many problems in the area of Computer Science, Biology etc. are solved with the help of algorithms which are based on graphs. Shortest path problem, Traveling Salesman Problem (TSP), network flow problems, vertex cover problem, graph colouring etc. are important graph-based problems with many practical applications in the real world. Our work is on Graph Colouring which is one of the most famous and well studied graph problems.

1.2 Vertex Colouring

Graph Colouring problem entails *colouring/labeling* of the vertices/edges of a graph based on some set of conditions which are to be satisfied. In other words, its a problem in which you allocate a colour/number to every vertex/edge of a graph such that a set of constraints are satified. There are different versions of Graph Colouring and the one which is of interest to us is that of Vertex Colouring.

1.2.1 Classical Vertex Colouring Problem

Vertex Colouring is the most basic version of Graph Colouring and other Graph Colouring problems can be presented as a Vertex Colouring problem. In it's classical form, Vertex Colouring is:

> ***Vertex Colouring:*** *Colouring all the vertices of a graph such that adjacent vertices have different colours. That is, there shouldn't be an edge where the incident vertices share the same colour.*

There are other forms of vertex colouring where additional conditions than the one given above need to be considered while colouring. In our work, we are concerned only with the classical form of vertex colouring which hereinafter interchangeably referred to simply as Graph Colouring.

1.2.2 Chromatic Number $\chi(G)$

A graph *G* is said to *k*-colourable, if *G* can be coloured using *k* colours. For example, from the *Four Colour Theorem*, we have that all planar graphs are 4-colourable. Also, all bipartite graphs are 2-colourable.

The *Chromatic Number* of a graph G, denoted by $\chi(G)$, is the minimum number of colours required to colour a graph. That is, $\chi(G)$, is the minimum value of all *k* for which the graph *G* is *k*-colourable. Therefore, if a graph is *k*-colourable, we have:

$$\chi(G) \leq k$$

1.2.3 Complexity

Graph Colouring is a computationally complex problem. To decide if a Graph can be coloured using *k* colours, is an NP-complete problem. Whereas, finding the Chromatic Number of a graph ($\chi(G)$) is proved to be an NP-hard problem.

There exist many algorithms like Greedy Colouring, approximation algorithms and randomized algorithms. There also exist polynomial time algorithms for some specific family of graphs. For example, it can be decided if a graph can be coloured using 2 colours by checking if it is a bipartite graph. This can be done in polynomial time using Breadth First Search (BFS).

1.2.4 Applications

Graph Colouring problem, which started as a map colouring problem (four colour theorem), finds many important real applications including but not limited to:

- Scheduling problems like job scheduling across multiple nodes in a distributed computing environment
- Register allocation problem during compilation
- Solving Sudoku
- Pattern matching applications

1.3 Parallelization

1.4 GPGPU

CHAPTER 2

PARALLEL GRAPH COLOURING

CHAPTER 3

PARALLEL GRAPH COLOURING: INCREMENTAL

CHAPTER 4

PARALLEL GRAPH COLOURING: DECREMENTAL

CHAPTER 5

EXPERIMENTAL EVALUATION

CHAPTER 6

CONCLUSION AND FUTURE WORK