Project Proposal
GPU-Based Fast Minimum Spanning Tree Using Data Parallel Primitives

Dibyadarshan Hota
Department of Computer Science and Engineering
NITK Surathkal
Email: dibyadarshan.hota@gmail.com

Omkar Prabhu
Department of Computer Science and Engineering
NITK Surathkal
Email: omkarp.nitk@gmail.com

Shashank P
Department of Computer Science and Engineering
NITK Surathkal
Email: shashankp5424@gmail.com

***Abstract*—A minimum spanning tree is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. Minimum spanning tree plays a key role in a broad domain of applications, including network organization, touring problems and VLSI layout.**

I. Problem statement

Our aim is to implement a minimum spanning tree algorithm using Prim's approach on Nvidia GPU under CUDA architecture using Min-Reduction[1]. Prim's approach is a greedy algorithm used to find a spanning tree whose sum of edge weights is as small as possible. It is a classical problem in graph theory that plays a key role in a broad domain of applications. By using new developed GPU-based Min-Reduction data parallel primitive in the key step of the algorithm, higher efficiency can be achieved.

II. Introduction

The minimum (weight) spanning tree (MST) problem is given an connected undirected weighted graph $G = (V, E, w)$ with non-negative weights, find a spanning tree of minimum weight, where the weight of a tree T is defined as:

$$w(T) = \sum_{e \in E(T)} w(e)$$

Prim's approach in solving MST involves the following steps:

- Choose any starting vertex. Look at all edges connecting to the vertex and choose the one with the lowest weight and add this to the tree.
- Look at all edges connected to the tree that do not have both vertices in the tree. Choose the one with the lowest weight and add it to the tree.
- Repeat step 2 until all vertices are in the tree.

III. Project execution plan

A. Implementation framework

Framework used is CUDA. CUDA is a parallel computing platform and application programming interface (API) model created by Nvidia.

B. Inputs

We choose the random graph generator from Georgia Tech graph generator suite[2].

C. Expected outputs

Experimental results show that we obtain about 2 times speedup on Nvidia GTX260 GPU over the CPU implementation and 3 times speedup over non-primitives GPU implementation.

D. Metrics to measure

Analysis of runtimes of serial Prim's algorithm and parallel implementation of Prim's algorithm.

E. Testing Platform

The specifications of the system used by the authors of the paper[1] are Intel Pentium4 3GHz CPU,2G host memory, NVIDIA GeForce GTX260 GPU, 896M device memory, Linux RedHat 5 OS.

IV. Project timeline

The project will be split in multiple phases:

- Phase 1: Conceptual understanding of MST, Prim's, related parallel primitives.
- Phase 2: Serial implementation of Prim's approach for minimum spanning tree.
- Phase 3: Implementing Prim's algorithm using CUDA.
- Phase 4: Run-time analysis.

V. Work Distribution

We intend to split each phase into different tasks and assign members of the team equal parts. Collaboration and individual tasks will be distributed on GitHub[3].

VI. Up-to-date progress

Progress so far includes:

- Thorough understanding of greedy approach used by Prim's algorithm.
- Understanding of CUDA architecture.
- Basic idea of Min-Reduction data parallel primitive required to optimize the algorithm (presented in the chosen paper).

References

[1] W. Wang, S. Guo, F. Yang, and J. Chen, "Gpu-based fast minimum spanning tree using data parallel primitives," *2nd International Conference on Information Engineering and Computer Science*, 2010.

[2] D. Bader and K. Madduri, "Gtgraph: A synthetic graph generator suite," *Tech. Rep*, 2006.

[3] P. Omkar, H. Dibyadarshan, and P. Shashank, "Gpu-based-fast-minimum-spanning-tree github repository." https://github.com/Dibyadarshan/GPU-Based-Fast-Minimum-Spanning-Tree, Oct 2018.