Bellman Ford on GPU

using Compute Unified Design Architecture (CUDA)

1. GRAPH REPRESENTATION ON CUDA

A graph G (V, E) is commonly represented as an adjacency matrix. For sparse graphs such are presentation wastes a lot of space. Adjacency list is a more compact representation for graphs.

We represent graphs in compact adjacency list form, with adjacency lists packed into a single large array. Each vertex points to the starting position of its own adjacency list in this large array of edges. Vertices of graph G (V, E) are represented as an array Va. Another array Ea of adjacency lists stores the edges with edges of vertex i+1 immediately following the edges of vertex i for all i in V. Each entry in the vertex array Va corresponds to the starting index of its adjacency list in the edge array Ea. Each entry of the edge array Ea refers to a vertex in vertex array Va.

1. BELLMAN- FORD ALGORITHM

Bellman Ford algorithm is a SSSP(Single Source Shortest Path) finding algorithm that calculates shortest paths from a single source vertex to all of the other vertices in a weighted directed graph. It uses the relax method in which the approximate distance to each vertex is always greater than or equal to the true distance, and is replaced by the minimum of its newly calculated value and old value. In this algorithm all the edges are relaxed for |V| - 1 times, where |V| is the number of vertices in the graph. In this way each node will get its shortest distance from the source node.

1. Parallel Bellman Ford

In this implementation, number of threads is equal to the number of edges in the graph and these edges will relax in parallel for |V| number of iterations.

In the first kernel “initializeArray”, the node weight of each node except the source node is initialised to „infinity‟, and for the source node it is initialised to „0‟.

In the second kernel “RELAX” the cost of each neighbour is updated if it is greater than the cost of current vertex plus the edge weight to that neighbour.

1. Comparison of Basic Bellman Ford using CPU and Parallel Bellman Ford using GPU.

Experimental Results:

|  |  |  |
| --- | --- | --- |
| No. of edges | Time in seconds | |
| CPU | GPU |
| 1500 | 1.2 | 1.7 |
| 2500 | 3 | 2 |
| 5000 | 13 | 6 |
| 10000 | 55 | 27 |
| 20000 | 228 | 135 |
| 30000 | 570 | 352 |

