Interim Project Report 2 - Analysis of A Faster Algorithm for Betweenness Centrality by Brandes

Sequential Code Method:

The fastest existing method for the exact BC computation, Brandes Algorithm, requires Θ(n*m) operations for unweighted graphs and Θ(n*m+n^2 log n) for graphs with positive edge weights . The algorithm computes for every node s ∈ V a slightly-modified version of a single-source shortest-path tree (SSSP tree), producing for each s the directed acyclic graph (DAG) of all shortest paths starting at s. Exploiting the information contained in the DAGs, the algorithm computes the dependency δs(v) for each node v, that is the sum over all nodes t of the fraction of shortest paths between s and t that v is internal to. The betweenness of each node v is simply the sum over all sources s ∈ V of the dependencies δs(v). Therefore, we can see the dependency δs(v) as a contribution that gives to the computation of cB(v).

For weighted graph SSSP is calculated using Dijkstra Algorithm and for unweighted using BFS in the implementation.

Results:g++ Brandes.cpp , ./a.out , enter the filename.
UnWeighted Graph:Calculation of Single source shortest path by BFS

==========
INPUT GRAPH STATS
>Weighted: false
>#ofNodes: 6
>#ofEdges: 6
==========

Time Taken : 0.000281096

Approx GFlops : 0.00012807

> Closeness Centrality
Node 0: 0.555556
Node 1: 0.833333
Node 2: 0.555556

Node 3: 0.5
Node 4: 0.625
Node 5: 0.416667

> Node Betweenness Centrality
Node 0: 0
Node 1: 0.8
Node 2: 0
Node 3: 0
Node 4: 0.4
Node 5: 0

> Edge Betweenness Centrality
Edge 0-1: 0.266667
Edge 0-2: 0.0666667
Edge 1-2: 0.266667
Edge 1-3: 0.333333
Edge 1-4: 0.533333
Edge 4-5: 0.333333

Weighted Graph:Calculation of Single Source Shortest Path by Dijkstra
==========
INPUT GRAPH STATS
>Weighted: true
>#ofNodes: 6
>#ofEdges: 7
==========

Time Taken : 0.000356913

Approx GFlops : 0.000117676

> Closeness Centrality
Node 0: 0.217391
Node 1: 0.384615
Node 2: 0.333333
Node 3: 0.172414
Node 4: 0.294118

Node 5: 0.238095

> Node Betweenness Centrality
Node 0: 0
Node 1: 0.766667
Node 2: 0.183333
Node 3: 0
Node 4: 0.366667
Node 5: 0

> Edge Betweenness Centrality
Edge 0-1: 0.122222
Edge 0-2: 0.188889
Edge 0-5: 0.0222222
Edge 1-2: 0.388889
Edge 1-3: 0.333333
Edge 1-4: 0.511111
Edge 4-5: 0.311111

The SSSP on each node can be parallelized using various approaches discussed in class.The for loop in the Brandes method is completely parallel.