Advanced Programming Techniques

Connected components

Recommended programming language: C++

Disclaimer: efficient implementations of graph data structures and algorithms can be already found in various dedicated libraries. The goal is to write your own code!

1) Implement two serial algorithms for computing all connected components of a graph, one using DFS and the other BFS.

2) Use your parallelized versions of DFS/BFS in order to speed up the computation. Compare the performances between serial and parallel implementations on large datasets.

3) Our algorithm insofar consists of a main loop where we iterate over the vertices and we start a search from every vertex not in one of the connected components that we already computed. Discuss about the feasibility of parallelizing the main loop (you may implement this feature and observe what is going on!).

4) Implement the four parallel algorithms of Cliff Liu and Tarjan, discussed in class in order to compute the connected components. Compare the results with the algorithms from question 2).

5) Use any implement from question 4) in order to implement a parallel algorithm for computing the biconnected components. Apply this algorithm to some large datasets.