



Efficient parallel algorithms using multi-core shared-memory architectures for graph measures

Sangeetha, M.

Faculty of Applied Science

University of Vavuniya

sangeethamahendran279@gmail.com

ABSTRACT

Exploring the structural and functional properties of large real-world graphs, such as detecting community structure in large social networks and assessing connectivity of different brain regions in large brain graphs, are emerging research areas nowadays. Quantitative graph theory has been developed to deal with the quantification of structural and functional aspects of graphs. Generally, nodal and global graph measures have been employed for measuring the information of a graph. There has been significant recent interest in parallel graph processing due to the need to analyse today's large graphs quickly. Parallel graph algorithms, such as shortest path, graph colouring, and spanning tree, have received significant attention since the start of parallel computing. However, there is no parallelisation of graph measures using any parallel architectures. Modern desktop and laptop computers are equipped with multi-core processors with shared-memory architecture. The use of OpenMP presents many advantages for shared-memory systems. The OpenMP was initially concentrated on a thread-centric model to exploit data-parallelism and loop intensive applications. A task-centric model is also available, and that offers an efficient way to specify parallel tasks naturally. This work has designed and implemented efficient parallel algorithms for five graph measures (characteristic path length, nodal clustering coefficient, global efficiency, local efficiency of a node, and transitivity) on multi-core shared-memory systems using various parallel techniques as loop-centric parallelisation and matrix-matrix multiplication parallelisation. We compared the performances using a different number of cores and for different sizes of random graphs and serial algorithms in the same hardware environment. The experimental results of running time and speedup show a significant improvement in the parallel algorithms compared with serial algorithms.

Keywords: Graph measures, Parallel algorithm, Shared memory, Multicore, OpenMP.