Using Complex-Network Properties For Efficient Graph Analysis

Thomas MESSI NGUÉLÉ [a,b], Maurice TCHUENTE [b] and Jean-François MÉHAUT [b]
[a] *IRD, UMI 209 UMMISCO, Université de Yaoundé I, BP 337 Yaoundé Cameroun*
[b] *Université de Grenoble Alpes, INRIA-LIG, Corse, BP: 38400 Grenoble, France*

Abstract. A complex network is a set of entities in a relationship, modeled by a graph where nodes represent entities and edges between nodes represent relationships. Graph algorithms have inherent characteristics, including data-driven computations and poor locality. These characteristics expose graph algorithms to several challenges, because most well studied (parallel) abstractions and implementation are not suitable for them. This work shows how to use some complex-network properties, including community structure and heterogeneity of node degree, to tackle one of the main challenges in graph analysis applications: improving performance, by a proper memory management and an appropriate thread scheduling. In this paper, we first proposed Cn-order, a heuristic that combines advantages of the most recent algorithms (Gorder, Rabbit and NumBaCo) to reduce cache misses in graph algorithms. Second, we proposed deg-scheduling, a degree-aware scheduling to ensure load balancing in parallel graph applications. Then we proposed comm-deg-scheduling, an improved version of deg-scheduling that uses Cn-order to take into account graph order in scheduling. Experimental results on a 32 cores NUMA machine (NUMA4) (with Pagerank and livejournal for example) showed that Cn-order used with deg-scheduling (comm-deg-scheduling) outperforms the recent orders: with 32 threads, we reduce time by 35.92% compared to Gorder, 21.59% compared to Numbaco and 15.12% compared to Rabbit.

Keywords. graph analysis, performance, cache miss, scheduling, load balancing

1. Introduction

A network is a set of entities such as individuals or organizations, with connections between them. Such a system is modeled as a graph G = (V, E) where entities are represented by vertices in V, and connections are represented by edges in E. As outlined in [11], the advent of computers and communication networks that allow to gather and analyze data on a large scale, has lead to a shift from the study of individual properties of nodes in small specific graphs with tens or hundreds of nodes, to the analysis of macroscopic and statistical properties of large graphs also called complex networks, consisting of millions and even billions of nodes.

This huge size combined with the complexity of the questions raised, make the design of efficient algorithms for parallel complex network analysis a real challenge [9]. The first difficulty is that graph computations are data driven, with high data access to computation ratio. As a consequence, memory fetches must be managed very carefully, otherwise as observed in other domains such as on-line transactions processing, the processors can be stalled up to 50% of the time due to cache misses [15].

On the other hand, the irregular structure of complex networks combined with the spatial locality of graph exploration algorithms, make difficult the scheduling task. In-