



Laxman Dhulipala

Provably Efficient and Scalable Shared-Memory Graph Algorithms

Friday, November 8, 2019 – 9:30 a.m. – GHC 6501

Parallel graph algorithms are important to a variety of computational disciplines today due to the widespread availability of large-scale graph-based data. Existing work that processes very large graphs restricts itself to static graphs and uses distributed memory, which often requires large computational and hardware cost, while still being prohibitively slow. This thesis will argue that shared-memory algorithms and techniques can solve a wide range of fundamental problems on very large static, dynamic, and streaming graphs quickly, provably-efficiently, and scalably, using a modest amount of computational resources.

The first part of this thesis will focus on efficient parallel graph processing. We will propose the Graph Based Benchmark Suite, which contains provably-efficient implementations of over 20 fundamental graph problems. Our implementations solve these problems on the largest publicly available graph, the WebDataCommons hyperlink graph, with over 200 billion edges, in a matter of seconds to minutes. We will then introduce techniques to efficiently process graphs that are stored on non-volatile memory (NVRAMs). Compared to existing results in the literature for the WebDataCommons graph, both our shared-memory and non-volatile memory implementations use orders of magnitude fewer resources, and in many cases also run an ^{order} of magnitude faster than existing distributed memory codes.

The second part of this thesis will focus on algorithms and systems for dynamic and streaming graphs. We will work in the Parallel Batch-Dynamic model, which allows algorithms to ingest batches of updates and exploit parallelism. We will describe algorithms for forest-connectivity connectivity, and for maintaining low-outdegree orientations in this model. Next, we will turn to practical data structures and algorithms for representing graphs that change over time. We design a dynamic graph representation based on a new compressed purely-functional tree data structure, called a C-tree, which admits provably-efficient parallel batch updates. Compared to existing work, our dynamic graph representation scales to much higher update rates, while using significantly less memory. Based on these ideas, we propose a new graph-streaming system called Aspen, and show that using Aspen, we can concurrently update and analyze the WebDataCommons hyperlink graph on a single commodity multicore machine with a terabyte of main memory.

Thesis Committee:
Guy E. Blelloch, Chair
Umut Acar
Phillip B. Gibbons
Vahab Mirrokni, Google Research
Julian Shun, MIT