



LAXMAN DHULIPALA

Provably Efficient and Scalable Shared-Memory Graph Algorithms

Friday, August 7, 2020 – 12:30 p.m. – REMOTE

Analyzing large graphs quickly and cost-effectively is a major challenge, and existing systems capable of processing large graphs have high computational cost, only solve a limited set of problems, and have poor theoretical guarantees. Similarly, existing systems for dynamic or streaming graphs implement ad-hoc algorithms with unknown theoretical costs and suboptimal performance. This thesis argues that a shared-memory approach enables algorithms and systems for static, dynamic, and streaming graphs with strong theoretical guarantees, low computational cost, and state-of-the-art practical performance.

The first part of this thesis studies analytics for static graphs. We design a rich interface with provably-efficient primitives for parallel graph processing that extends Ligma. Using the interface, we describe 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 using a commodity multicore machine. We adapt these algorithms for graphs stored in non-volatile memory, thereby extending our results to graphs larger than main memory. Compared to existing results, our results apply to a much broader set of problems, use orders of magnitude fewer resources, and in many cases run an order of magnitude faster.

The second part of this thesis studies the batch-dynamic model, which generalizes the classic dynamic algorithms model by allowing algorithms to ingest batches of updates. In this model, we design work-efficient parallel algorithms with poly-logarithmic depth for the dynamic trees and dynamic connectivity problems. We show that shared-memory parallel batch-dynamic algorithms can achieve strong speedups, and outperform the fastest sequential dynamic algorithm baselines.

The final part of this thesis studies streaming graphs. We design a compressed purely-functional tree, called a C-tree, which admits efficient batch updates and enables a dynamic graph representation based on nested trees. Using this representation, we design a serializable graph-streaming system called Aspen that can concurrently apply updates and queries. Compared to existing work, Aspen achieves orders of magnitude higher update rates, while using less memory. We show that Aspen can concurrently update and analyze the WebDataCommons hyperlink graph on a single commodity multicore machine.

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