The More the Merrier: Efficient Multi-Source Graph Traversal

Manuel Then* then@in.tum.de
Moritz Kaufmann* kaufmannm@in.tum.de
Kien Pham† kien.pham@nyu.edu
Huy T. Vo† huy.vo@nyu.edu
Fernando Chirigati† fchirigati@nyu.edu
Tuan-Anh Hoang-Vu† tuananh@nyu.edu
Alfons Kemper* kemper@in.tum.de
Thomas Neumann* neumann@in.tum.de

* Technische Universität München † New York University

Motivation for Multi-Source Traversal

Many graph algorithms run multiple breadth-first searches + Real-world graphs are often small-world networks
- Shortest paths
- Closeness centrality
- K-hop neighborhoods
- Social networks
- Web graphs
- Communication networks

Goals

Leverage multi-source traversal in graph algorithms.
Optimize data access patterns for this kind of traversal and avoid redundant computation to improve graph analytics performance.

Multi-Source BFS (MS-BFS)

- Concurrently run many independent BFS traversals on the same graph
- Share traversals whenever possible
- Store BFS state per vertex as a bitset (3 bits per vertex and traversal)
- Represent BFS as SIMD bit operations
- Fully utilize cache line-sized memory accesses of modern CPUs

Evaluation

MS-BFS-based closeness centrality \( \frac{(C_i - 1)^2}{(N - 1) \cdot \sum_{v \in V} w(v, i)} \) on 4x Intel Xeon E7-4870v2, 1TB memory

Further improvements: Aggregated neighbor processing, Direction-optimizing, Prefetching, Batching for maximum sharing

Challenges

- Traversals require random data accesses with bad cache behavior and often cause CPU stalls
- Single bit accesses waste memory bandwidth
- Independent BFS runs redundantly visit vertices multiple times

Source available at https://github.com/mtodat/ms-bfs