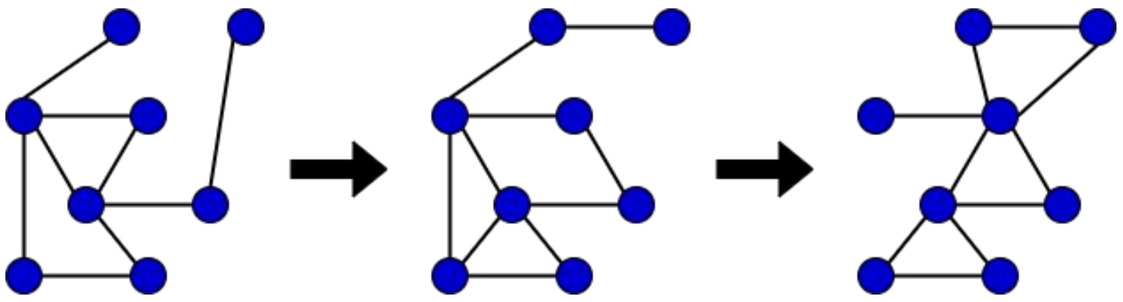
Parallel Local Motif Counting for Large-Scale Dynamic Datasets

Ali Khan[1]
Supervisors: Sanjukta Bhowmick[1], Michela Taufer[2]
[1]University of North Texas, [2]University of Tennessee Knoxville

Introduction

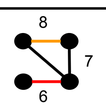
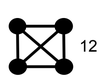
- Motif counting seeks to count small patterns in large networks
- Useful for finding similarity across networks, determining function of proteins, and clustering [1]
- Current methods assume a static network [2]. However, real-world networks change over time



We develop a tool to quickly count the local k-sized motifs in dynamic networks

Background

- Given an original graph G = (V,E) , by applying a batch of updates, B, of either inserted or deleted edges, we obtain the updated graph G'=(V,E') where E'=E ∪ B.
- A motif, also called a graphlet, are commonly occurring small subgraphs of a graph. We examine motifs with k<5 vertices.
- A GLOBAL motif count is the frequency of a motif, H, in a graph
- A LOCAL motif count is the frequency of a motif, H, incident to an edge in a graph. We count the ORBITS, or roles, of the edge in the motif by accounting for the automorphisms of the motif.
- All k<5 motifs and labeled orbits are listed below:

Wedge	Triangle	Star	Path	Tailed Triangle	Square	Diamond	Clique
							

Methodology

For each updated edge e, in parallel
1) Enumerate all distance-1 neighbors
2) Enumerate distance-2 neighbors
3) Construct all k<4 motif local counts from listed neighbors in constant time

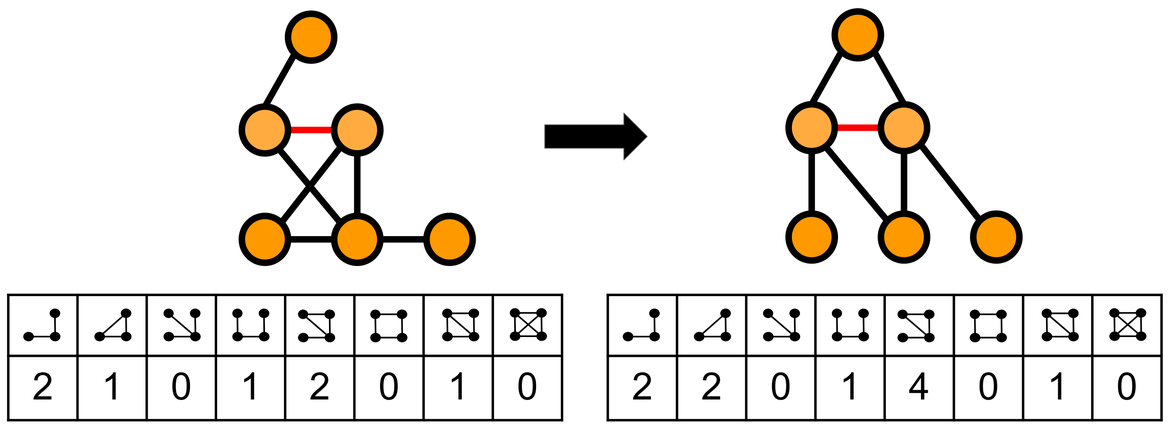
Finally update the local frequencies of motifs in G to obtain counts for G'

Parallelize with OpenMP (outer loop) and Cuda (nested parallelization) using Kokkos C++ Performance Portability Library

Acknowledgments

This work is supported by NSF grants 1900765 and 1900888, as well as 2341588

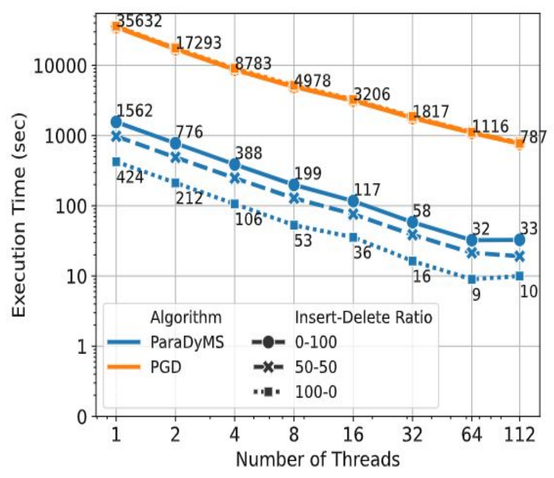
Working Example



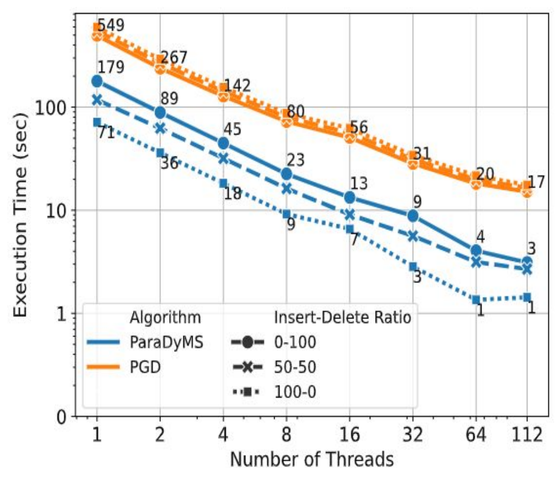
We show all k<5 motif frequencies for an edge (in red) for a static graph as well as after applying set of updates. For the original graph (left) the edge has 2 wedges, 1 triangle, 1 4-path, 2 tailed-triangles, and 1 diamond. For the updated graph (right), the motifs incident to the same edge are 2 wedges, 2 triangles, 1 4-path, 4 tailed-triangles, and 1 diamond.

Preliminary Results

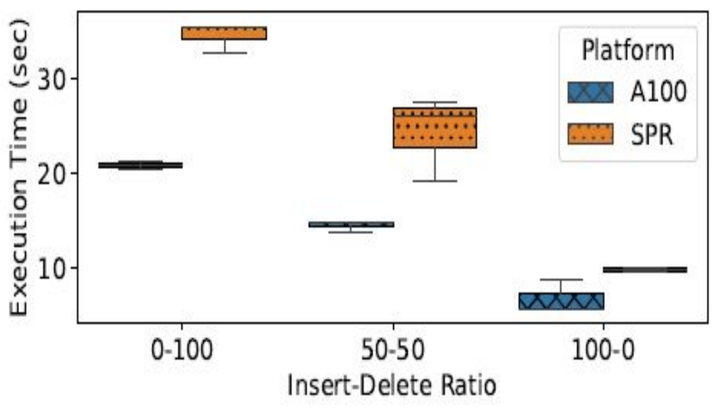
Orkut



Pokec



GPU vs CPU



Top: Results using OpenMP on two datasets: Orkut and Pokec with 1M updated edges and update ratios of 0% insertions and 100% deletions, 50% insertions and 50% deletions, and 100% insertions and 0% deletions. Our algorithm, ParadyMS, beats the baseline[2] while still demonstrating scalability across different number of threads.
Bottom: Execution time of our algorithm running on a single A100 GPU with on Orkut dataset with 1M edges changed compared to running on an Intel Sapphire Rapids CPU using all 112 cores.

References

- [1] N. Pržulj, "Biological Network Comparison Using Graphlet Degree Distribution," Bioinformatics, vol. 23, no. 2, pp. e177–e183, 2007.
- [2] N. K. Ahmed, J. Neville, R. A. Rossi, and N. Duffield, "Efficient Graphlet Counting for Large Networks," in 2015 IEEE international conference on data mining. IEEE, 2015, pp. 1–10.
- [3] A. Khan, N. Tan, J. Marquez, M. Taufer, and S. Bhowmick, "Paradyms: Parallel dynamic motif counting at scale," in CCGrid25: The IEEE International Symposium on Cluster, Cloud, and Internet Computing. IEEE, 2025.