Scaling K-Path Centrality using Optimized Distributed Data Structure

Lance Fletcher[1,2], Trevor Steil[1], Roger Pearce[1,2]
[1]Center for Applied Scientific Computing (CASC), Lawrence Livermore National Laboratory (LLNL)
Livermore, CA, USA
[2]Department of Computer Science and Engineering, Texas A&M University
College Station, TX, USA
{fletcher28,steil1,rpearce}@llnl.gov

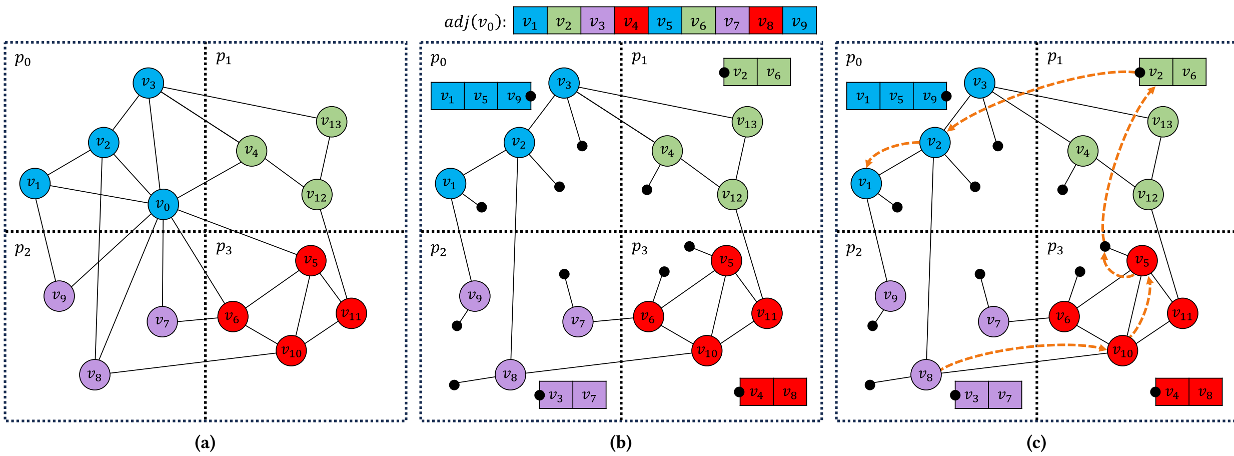

Figure 1: A 1D partitioning (a) of a graph with a hub vertex v_0 stored on processor p_0. A vertex delegation partitioning (b) of the same graph which shows the adjacency list $adj(v_0)$ delegated amongst all processors. The smaller adjacency list contained in each processor's partition represents the portion of $adj(v_0)$ owned by each processor. In (c), the dashed orange arrows are steps of a path taken in a vertex delegation partitioned graph. The order of the vertices visited is $v_8 \rightarrow v_{10} \rightarrow v_5 \rightarrow v_0 \rightarrow v_2 \rightarrow v_1$, the order of the processors visited is $p_2 \rightarrow p_3 \rightarrow p_3 \rightarrow p_1 \rightarrow p_0 \rightarrow p_0$.

ABSTRACT

K-Path centrality is based on the flow of information in a graph along simple paths of length at most K. This work addresses the computational cost of estimating K-path centrality in large-scale graphs by introducing the random neighbor traversal graph (RaNT-Graph). The distributed graph data structure employs a combination of vertex delegation partitioning and rejection sampling, enabling it to sample massive amounts of random paths on large scale-free graphs. We evaluate our approach by running experiments which demonstrate strong scaling on large real-world graphs. The RaNT-Graph approach achieved a 56,544x speedup over the baseline 1D partition implementation when estimating K-path centrality on a graph with 89 million vertices and 1.9 billion edges.

KEYWORDS

distributed computing, centrality, random paths, random walks

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-ABS-855042). Funding from LLNL LDRD project 21-ERD-020 was used in this work.

1 INTRODUCTION

κ-Path centrality (KPC) is a centrality metric based on the concept of information flowing through a graph along simple paths of length at most κ. A simple path is one which contains no repeating vertices. KPC assigns each vertex v a value based on the sum of the probabilities a simple path of length at most κ originating from all other vertices passes through v [1]. Estimating KPC is done by sampling many simple paths of length at most κ and assigning a vertex a value based on the number of paths which traverse through it. Estimating KPC has shown to identify vertices in graphs with high betweenness centrality [1] and has been utilized in many other graph problems [2–5]. To estimate KPC on large graphs, many paths must be sampled which quickly becomes computationally expensive. In the present work, we introduce the random neighbor traversal graph (RaNT-Graph), a distributed graph data structure capable of sampling massive numbers of random paths and walks.

2 APPROACH

The imbalances of storage, compute, and communication are problems often associated with graph algorithms due to the non-uniform

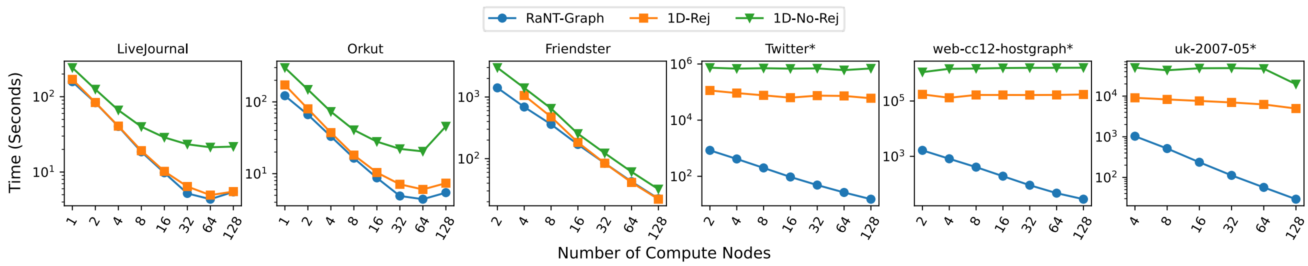

Figure 2: Strong scaling of RaNT-Graph, 1D-Rej, and 1D-No-Rej on various real world graphs where T paths were sampled (see Table 1). *Due to large compute time required, 1D-Rej and 1D-No-Rej values were estimated on these graphs. Times were estimated by timing the sampling of 1M paths and extrapolating this value based on the desired amount of paths to be sampled T as shown in Table 1.

topology present in many graphs. RaNT-Graph utilizes vertex delegation partitioning [6] to mitigate these issues. Delegating a vertex distributes the adjacency lists of high-degree vertices or *hubs* amongst all processors in a round-robin fashion as shown in Figure 1b. This partitioning technique has been employed in a variety of graph algorithms and has proven to help scaling capabilities [7–12].

In a 1D partitioning, each vertex is owned by a single processor. When sampling a large amount of paths, a hub is likely to have more paths pass through it than a lower degree vertex. Therefore, in a 1D partitioning, processors which own hubs often must do more work and communication than other processors. The vertex delegation partitioning of the graph reduces these imbalances by distributing the adjacency list of the high degree vertices.

Sampling a simple path involves recursively stepping to unvisited vertices until a termination condition is met. A step involves either stepping to an undelegated vertex or a delegated vertex. When stepping to an undelegated vertex u, the processor which owns u randomly chooses the successive node. When stepping to a delegated vertex v, a random edge e connected to v must be chosen first, then the processor which owns e continues the recursive process. Since paths are independent of each other, it is obvious that many paths can be sampled in parallel on a distributed graph. Figure 1c depicts both types of steps occuring in a single path.

RaNT-Graph also employs rejection sampling to quickly select the next vertex to traverse to in a simple path. Given the current vertex in a path v, a vertex from v's neighborhood $\mathcal{N}(v)$ must be chosen that has not previously been visited in the path $\mathcal{S}$. Constructing the set of unvisited vertices $\mathcal{U} = \mathcal{N}(v) \setminus \mathcal{S}$ takes $O(|\mathcal{N}(v)|)$ time and can be costly for large degree vertices. Therefore, the next vertex in a path is determined by selecting a random neighbor of v and accepting it if it is not already present in the path. Conversely, if the selected vertex is already in the path then a new neighbor is chosen until an unvisited vertex is found.

Lastly, RaNT-Graph utilizes the asynchronous communication library YGM [13] and is built upon many of the its distributed containers. YGM's ability to increase throughput via message buffering and it's asynchronous communication make it ideal for algorithms requiring irregular communication.

3 EXPERIMENTS

All experiments were conducted on LLNL's Catalyst cluster where each compute node is equipped with dual Intel Xeon E5-2695v2 processors totaling 24 cores and 128GB of DRAM. The network uses an Infiniband QDR interconnect. All implementations tested were written in C++ and utilized YGM.

To examine the strong scaling capability of the RaNT-Graph approach, we estimate KPC on multiple large scale graphs. Table 1 shows the total vertices n, total edges m, maximum degree d_{max}, total paths to sample T, and the maximum path length κ. T and κ are derived from equations proposed in the original κ-path paper [1] where $T = \lfloor 2\kappa^2 n^{1-2\alpha} \ln n \rfloor$ and $\kappa = \lfloor \ln(n+m) \rfloor$ with $\alpha = 0.2$. We compare our approach with two 1D partitioned implementations, one which uses rejection sampling (1D-Rej) and one which does not (1D-No-Rej). As seen in Figure 2, when d_{max} is large, RaNT-Graph provides a substantial speedup over the 1D partitioned implementations.

Table 1: Graphs used in strong scaling experiments.

Graph	n	m	d_{max}	T	κ
Orkut [14]	3M	117M	33K	74M	18
LiveJournal [15]	4.85M	43M	20K	102M	18
Twitter [16]	42M	1.2B	3M	580M	21
Friendster [14]	66M	1.8B	5.2K	857M	22
web-cc12-hostgraph [17]	89M	1.9B	3M	1B	22
uk-2007-05 [18]	106M	3.3B	975K	1.2B	22

4 CONCLUSION

Estimating κ-path centrality can require sampling large amounts of paths when applied to large-scale graphs. We introduce RaNT-Graph, a novel graph data structure optimized for sampling massive amounts of random simple paths. It combines vertex delegation partitioning with rejection sampling to reduce compute, storage, and communication imbalances caused by high-degree vertices. We demonstrate the strong scalability of RaNT-Graph on multiple large-scale real-world graphs. When compared to the baseline 1D partitioned implementations, our approach yields up to a $56,544\times$ speedup.

REFERENCES

[1] T. Alahakoon, R. Tripathi, N. Kourtellis, R. Simha, and A. Iamnitchi, "K-path centrality: A new centrality measure in social networks," in *Proceedings of the 4th Workshop on Social Network Systems*, ser. SNS '11. New York, NY, USA: Association for Computing Machinery, 2011.

[2] J. Blackburn, R. Simha, N. Kourtellis, X. Zuo, M. Ripeanu, J. Skvoretz, and A. Iamnitchi, "Branded with a scarlet "C": cheaters in a gaming social network," in *Proceedings of the 21st international conference on World Wide Web*, ser. WWW '12. New York, NY, USA: Association for Computing Machinery, Apr. 2012, pp. 81–90.

[3] P. De Meo, E. Ferrara, G. Fiumara, and A. Provetti, "Mixing local and global information for community detection in large networks," *Journal of Computer and System Sciences*, vol. 80, no. 1, pp. 72–87, Feb. 2014.

[4] ——, "Enhancing community detection using a network weighting strategy," *Information Sciences*, vol. 222, pp. 648–668, Feb. 2013.

[5] A. Biswas and B. Biswas, "Community-based link prediction," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18 619–18 639, Sep. 2017.

[6] R. Pearce, M. Gokhale, and N. M. Amato, "Faster Parallel Traversal of Scale Free Graphs at Extreme Scale with Vertex Delegates," in *SC '14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, Nov. 2014, pp. 549–559, iSSN: 2167-4337.

[7] J. Zeng and H. Yu, "A Distributed Infomap Algorithm for Scalable and High-Quality Community Detection," in *Proceedings of the 47th International Conference on Parallel Processing*, ser. ICPP '18. New York, NY, USA: Association for Computing Machinery, Aug. 2018, pp. 1–11.

[8] Y. Pan, R. Pearce, and J. D. Owens, "Scalable Breadth-First Search on a GPU Cluster," in *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, May 2018, pp. 1090–1101, iSSN: 1530-2075.

[9] J. Zeng and H. Yu, "A Scalable Distributed Louvain Algorithm for Large-Scale Graph Community Detection," in *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2018, pp. 268–278, iSSN: 2168-9253.

[10] H. Cao, Y. Wang, H. Wang, H. Lin, Z. Ma, W. Yin, and W. Chen, "Scaling graph traversal to 281 trillion edges with 40 million cores," in *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, ser. PPoPP '22. New York, NY, USA: Association for Computing Machinery, Mar. 2022, pp. 234–245.

[11] T. Reza, C. Klymko, M. Ripeanu, G. Sanders, and R. Pearce, "Towards Practical and Robust Labeled Pattern Matching in Trillion-Edge Graphs," in *2017 IEEE International Conference on Cluster Computing (CLUSTER)*, Sep. 2017, pp. 1–12, iSSN: 2168-9253.

[12] B. A. Page and P. M. Kogge, "Scalability of Hybrid SpMV with Hypergraph Partitioning and Vertex Delegation for Communication Avoidance," *International Conference on High Performance Computing & Simulation (HPCS 2020)*, Mar. 2021.

[13] B. Priest, T. Steil, G. Sanders, and R. Pearce, "You've got mail (ygm): Building missing asynchronous communication primitives," in *2019 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, 2019, pp. 221–230.

[14] J. Yang and J. Leskovec, "Defining and evaluating network communities based on ground-truth. corr abs/1205.6233 (2012)," *arXiv preprint arXiv:1205.6233*, 2012.

[15] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan, "Group formation in large social networks: membership, growth, and evolution," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 44–54.

[16] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*, 2010, pp. 591–600.

[17] R. Meusel, S. Vigna, O. Lehmberg, and C. Bizer, "Graph structure in the web—revisited: a trick of the heavy tail," in *Proceedings of the 23rd international conference on World Wide Web*, 2014, pp. 427–432.

[18] P. Boldi, M. Rosa, M. Santini, and S. Vigna, "Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks," in *Proceedings of the 20th international conference on World Wide Web*, 2011, pp. 587–596.