Project Proposal

Dissecting The Complex: Benchmarking vertex-centric node importance algorithms in iPregel and comparing results with standard node importance scores

Theresa Hradilak
theresa.hradilak@student.hpi.de

Robin Wersich
robin.wersich@student.hpi.de

Hasso-Plattner-Institute, University of Potsdam
Working Group Data Analytics and Computational Statistics

1 Objective

Traditionally, easy-to-calculate measures of importance are used to estimate the relevance of persons in real-world networks, e.g., the h-index for scientists or the follower count for influencers. With the continuing increase in computational power, other more computationally intensive measures based on network algorithms like PageRank, Betweenness Centrality, and Closeness Centrality become feasible. We aim to investigate and benchmark the efficiency and programmability of vertex-centric node importance algorithms in iPregel, especially compared to corresponding standard sequential algorithms. As a second step, we aim to compare the calculated network-analysis-based node importance scores with commonly used node importance scores for the examined networks.

2 Approach

We will benchmark the performance of SSSP, PageRank, Betweenness Centrality, and Closeness Centrality on four differently sized real-world networks for a vertex-centric iPregel implementation and a standard sequential implementation. As Betweenness Centrality and Closeness Centrality are not yet implemented in iPregel, we will add an implementation and evaluate the programmability for these two examples.

We will retrieve standard, easy-to-calculate node importance scores for each of our networks. Then we will compare them to the results of PageRank, Betweenness Centrality, and Closeness Centrality on our networks.

3 Resources

3.1 Datasets

We plan to use networks from the SNAP [Leskovec and Krevl, 2014] and ICON [Aaron Clauset and Sainz, 2016] network databases. Table 1 shows the exact datasets we plan to use.

3.2 Literature

We plan to refer to the following literature:

- In regards to implementation of vertex-centric frameworks and vertex-centric programming in general: The original iPregel paper [Capelli et al., 2019] and the original Pregel whitepaper [Malewicz et al., 2010]
- For a general analysis of vertex-centric algorithms against their sequential counterparts: A systematic study about this topic [Khan, 2016]
- For vertex-centric algorithm implementations to compare and refer to: A paper about a vertex-centric Betweenness Centrality implementation [Li et al., 2015] and a paper about vertex-centric graph diameter computation [Pennycuff and Weninger, 2015]

Network Dataset	Nodes	Edges	Characteristics	Source	Standard node importance measure
AS-20, Autonomous System Network, as example test dataset	6474	13233	undirected	SNAP	-
cit-HepTh, Arxiv High Energy Physics paper citation network	27,770	352,807	directed, labeled	SNAP	h-index
higgs-twitter, Twitter interaction network	456,631	14,855,875	directed	SNAP	follower count
sx-stackoverflow, Stack Overflow interaction network	2,601,977	36,233,450	directed	SNAP	reputation count
IMDb actors (2011), Actor collaboration network	1,660,332	171,438,479	undirected	ICON	prizes, wage

Table 1: Planned to use networks

- For the analysis of using graph-analysis-based node importance scores instead of standard node importance scores on the example of h-index vs. PageRank: A paper about a PageRank-Index implementation [Senanayake et al., 2015] and a paper about combining h-index and PageRank [Gao et al., 2016]

3.3 Tooling

For the implementation of our project we intend to use:

- iPregel as the vertex-centric framework to benchmark upon and for our implementations of Betweeness Centrality and Closeness Centrality
- Ligra for the conversion of SNAP networks to the iPregel required Ligra format
- SNAP and/or Neo4j for standard, sequential implementations of node importance algorithms
- ggplot2 and iGraph R packages for visualization of our results
- Python for neccessary preprocessing of networks
- Overleaf and Zotero for documentation and citation organisation

4 Timeline

4.1 Timeplan

Table 2 shows our envisioned time plan.

4.2 Deliverables

1. Open-Source implementation of Betweenness Centrality and Closeness Centrality using iPregel framework. Programmability evaluation text in regards to vertex-centric programming for these two algorithms.
2. Table comparing runtime and storage usage of vertex-centric algorithms implemented in iPregel to that of sequential algorithms for SSSP, PageRank, Closeness Centrality and Betweenness Centrality for each of our four different networks.
3. Table with obtained node importance scores for all four networks. Documented observations and visualizations of comparison against suitable obtained standard node importance scores for at least the Twitter interaction network (*higgs-twitter*) and the physics paper citation network (*cit-HepTh*).

#	Task	Milestone	Date
1	Find standard, sequential implementations for PageRank, Betweenness Centrality, Closeness Centrality, and SSSP (probably from SNAP). Get them to run on the small example network Autonomous Systems *AS-20*.		27.05.21
2	Implement vertex-centric Betweenness Centrality, and Closeness Centrality algorithms in iPregel. Evaluate programmability. Check implementation correctness against results calculated with standard, sequential implementations on *AS-20* network.	All algorithms ready	03.06.21
3	Preprocess network datasets and convert into fitting format to run analysis with iPregel and sequential implementation framework. Obtain corresponding standard node importance scores for at least *higgs-twitter* and *cit-HepTh* networks.	All input data ready	10.06.21
4	Run analysis with selected algorithms on network datasets with iPregel and sequential framework implementations. Benchmark runtime and needed storage. Store benchmarking and analysis results appropriately.	All result data obtained	27.06.21
5	Compare calculated network-analysis-based node importance scores to obtained standard node importance scores.		11.07.21
6	Visualize and document results in Project Report.	Project (Report) Finished	15.08.21

Table 2: Time Plan

5 Additional Work - Six Degrees of Wikipedia

If we have time left at the end of our project we would like to make some additional analyses on the SNAP *wiki-topcats* network to find heuristics for the Six Degrees of Wikipedia Game. The Six Degrees of Wikipedia Game consists in finding the shortest path from one Wikipedia article to another using only intra-Wikipedia hyperlinks. One can try it out on this webpage.

As Betweenness Centrality calculates the number of shortest paths a node belongs to in a network, we assume that this value might be useful to generate heuristics for winning the game. More concretely, a node with a high Betweenness Centrality score should be a good choice as the next node on the shortest path the player is trying to find. For a player it would be e.g. interesting to know which categories of nodes tend to have a high Betweenness Centrality.

The *wiki-topcats* data set provides us with a hyperlink connection network of articles in the top categories of Wikipedia. Additionally, each node is labeled with the category it belongs to. We want to analyse which categories tend to have nodes with high Betweenness Centrality scores.

References

[Aaron Clauset and Sainz, 2016] Aaron Clauset, E. T. and Sainz, M. (2016). The colorado index of complex networks. `https://icon.colorado.edu/`.

[Capelli et al., 2019] Capelli, L. A., Hu, Z., Zakian, T. A., Brown, N., and Bull, J. M. (2019). iPregel: Vertex-centric programmability vs memory efficiency and performance, why choose? *Parallel Computing*, 86:45–56.

[Gao et al., 2016] Gao, C., Wang, Z., Li, X., Zhang, Z., and Zeng, W. (2016). PR-Index: Using the h-Index and PageRank for Determining True Impact. *PLOS ONE*, 11(9):e0161755.

[Khan, 2016] Khan, A. (2016). Vertex-Centric Graph Processing: Good, Bad, and the Ugly. page 4.

[Leskovec and Krevl, 2014] Leskovec, J. and Krevl, A. (2014). SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`.

[Li et al., 2015] Li, B., Gao, Z., Niu, J., Lv, Y., and Zhang, H. (2015). Vertex-centric Parallel Algorithms for Identifying Key Vertices in Large-Scale Graphs. In *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*, pages 225–231, New York, NY. IEEE.

[Malewicz et al., 2010] Malewicz, G., Austern, M. H., Bik, A. J. C., Dehnert, J. C., Horn, I., Leiser, N., and Czajkowski, G. (2010). Pregel: a system for large-scale graph processing. page 11.

[Pennycuff and Weninger, 2015] Pennycuff, C. and Weninger, T. (2015). Fast, exact graph diameter computation with vertex programming. In *1st High Performance Graph Mining workshop*. Barcelona Supercomputing Center.

[Senanayake et al., 2015] Senanayake, U., Piraveenan, M., and Zomaya, A. (2015). The Pagerank-Index: Going beyond Citation Counts in Quantifying Scientific Impact of Researchers. *PLOS ONE*, 10(8):e0134794.