

PegasusN: A Scalable and Versatile Graph Mining System

Ha-Myung Park,¹ Chiwan Park,² U Kang²

¹KAIST, ²Seoul National University

hamyung.park@kaist.ac.kr, chiwanpark@snu.ac.kr, ukang@snu.ac.kr

Abstract

How can we find patterns and anomalies in peta-scale graphs? Even recently proposed graph mining systems fail in processing peta-scale graphs. In this work, we propose PegasusN, a scalable and versatile graph mining system that runs on Hadoop and Spark. To handle enormous graphs, PegasusN provides and seamlessly integrates efficient algorithms for various graph mining operations: graph structure analyses, subgraph enumeration, graph generation, and graph visualization. PegasusN quickly processes extra-large graphs that other systems cannot handle.

Introduction

Graphs are everywhere: friendship networks, the World Wide Web, knowledge bases, biological networks, etc. Many machine learning tasks are directly or indirectly related to graphs; some tasks such as clustering, partitioning, and classification operate directly on graphs, while others use graph properties such as degrees, graph patterns, connected components, and centralities (Faloutsos, Faloutsos, and Faloutsos 1999; Kang et al. 2010; 2011; Jung et al. 2017). Meanwhile, with the rapid development of technologies, graphs that do not fit into the memory are now commonplace: e.g., more than one billion Facebook users form a huge friendship network, and trillions of web pages are linked to each other on the Web.

Several systems have been developed to handle such enormous graphs but they all fail in processing web-scale graphs. Single machine systems like GraphChi (Kyrola, Blelloch, and Guestrin 2012) and distributed memory systems like GraphX (Gonzalez et al. 2014) cannot process graphs exceeding the external memory space and the distributed memory space, respectively. Even Pegasus (Kang, Tsourakakis, and Faloutsos 2009), a MapReduce based system resolving the space shortage problem, cannot process extra large graphs because of heavy network and disk I/Os by reading and shuffling the entire data many times.

In this work, we introduce PegasusN, a scalable and versatile graph mining package that runs on Hadoop and Spark. To handle enormous graphs, PegasusN provides and seamlessly integrates distributed algorithms for the following four types of operations with various applications (see Figure 1).

Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

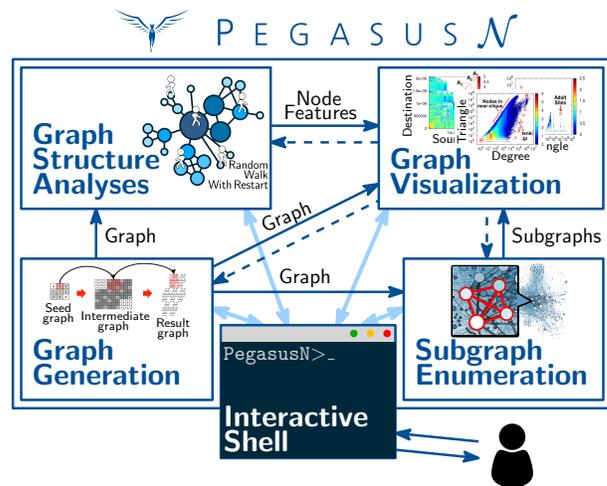


Figure 1: System overview.

Graph structure analyses on a peta-scale graph with PageRank, radii/diameter, shortest-paths, random walk with restart, connected components, and label propagation (Park et al. 2016; 2017). Applications: pattern and anomaly detection, spectral analysis, personalized recommendation, graph partitioning, and node importance measurement.

Subgraph enumeration to find all subgraphs of a large graph that are isomorphic to a pattern graph (Park, Myaeng, and Kang 2016). Applications: social network analysis, significance measurement of motifs, web spam detection, and community detection.

Peta-scale graph generation of R-MAT, Kronecker, and Erdős-Rényi random graph (Jeon, Jeon, and Kang 2015). Applications: simulation for new algorithm, graph modeling, and graph sampling/extrapolation.

Graph visualization providing a quick and useful view of peta-scale graphs (Jeon, Jeon, and Kang 2015).

The codes of PegasusN are available at <http://datalab.snu.ac.kr/pegasusn>.

System Overview

PegasusN provides an integrated interactive shell managing all the functionalities. Graph structure analyses and subgraph enumeration enable us to compute various graph fea-

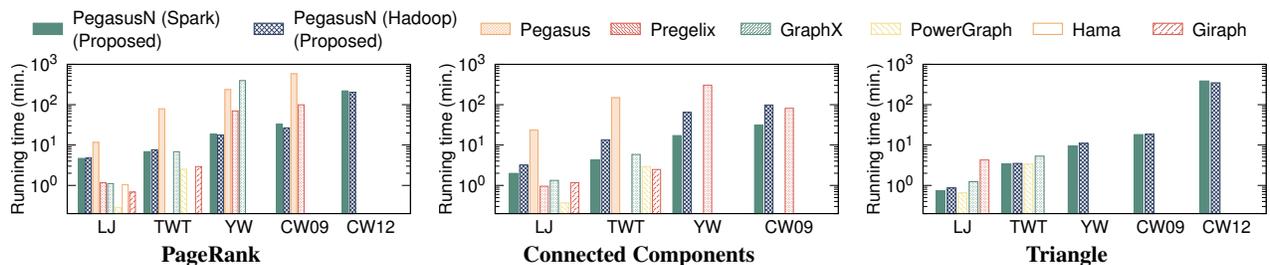


Figure 2: Running times of various systems on three tasks. Missing methods for some datasets mean they failed on the datasets.

tures. Visualization of the distribution and correlation of the features helps us find patterns and anomalies from the graph, and model the graph along with the graph generator of PegasusN; we verify how well the model represents the original graph by visualizing the model’s synthetic graphs generated by the graph generator. Then, the synthetic graphs are regarded as samples of the original graph and they are used for various purposes including graph algorithm testing.

Each module in PegasusN is designed to reduce intermediate data and balance workloads for superior performance in distributed systems. **Pre-partitioning** – many iterative graph algorithms require dividing the entire graph into overlapping or non-overlapping subgraphs multiple times. Instead of generating subgraphs from scratch in every iteration, PegasusN partitions the graph into blocks that do not share any edges with each other and makes each subgraph by combining blocks. The blocks are created in consideration of the sparsity of the graph so that subgraphs are similar in size, and thus workloads are evenly distributed. By caching the blocks in memory or on disk once at the beginning, PegasusN avoids the generation of massive intermediate data in every iteration. **Pruning** – in iterative algorithms, data to process change over time and dwindle rapidly in size. PegasusN significantly reduces communication cost as well as computational cost by excluding from the next iteration the data that will not change in the future.

Performance

Figure 2 shows the running times of PegasusN and existing graph mining systems on three tasks: PageRank, connected components, and triangle enumeration. We use a cluster of 20 machines equipped with an Intel Xeon E5-2640v3 and 32GB RAM. We use five real-world graphs summarized in Table 1. PegasusN is the only one that succeeds in processing CW12, the largest graph used in this experiment.

Conclusion

PegasusN is a scalable graph mining system for Hadoop and Spark. The system supports four types of graph algorithms (graph structure analyses, subgraph enumeration, graph generation, and graph visualization) in a distributed manner. PegasusN shows the best performance on real world graphs, processing CW12, the largest graph used in this paper with 6.2 billion nodes and 72 billion edges. PegasusN provides extremely useful toolsets that are needed by data mining and machine learning researchers and practitioners.

Table 1: Summary of datasets.

Dataset	Nodes	Edges	Description
CW12	6231126594	71746553402	Webgraph by Lemur Proj. (2009)
CW09	1684876525	7939647897	Webgraph by Lemur Proj. (2012)
YW	720242173	6636600779	Webgraph by Yahoo (2002)
TWT	41652230	1468365182	Follow network in Twitter
LJ	4847571	68993773	Friendship network in LiveJournal

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIT) (No.R0190-15-2012, High Performance Big Data Analytics Platform Performance Acceleration Technologies Development). U Kang is the corresponding author.

References

Faloutsos, M.; Faloutsos, P.; and Faloutsos, C. 1999. On power-law relationships of the internet topology. In *SIGCOMM*, 251–262.

Gonzalez, J. E.; Xin, R. S.; Dave, A.; Crankshaw, D.; Franklin, M. J.; and Stoica, I. 2014. Graphx: Graph processing in a distributed dataflow framework. In *OSDI*, 599–613.

Jeon, B.; Jeon, I.; and Kang, U. 2015. Tegviz: Distributed tera-scale graph generation and visualization. In *ICDMW*, 1620–1623.

Jung, J.; Park, N.; Sael, L.; and Kang, U. 2017. Bepi: Fast and memory-efficient method for billion-scale random walk with restart. In *SIGMOD*, 789–804.

Kang, U.; McGlohon, M.; Akoglu, L.; and Faloutsos, C. 2010. Patterns on the connected components of terabyte-scale graphs. In *ICDM*, 875–880.

Kang, U.; Papadimitriou, S.; Sun, J.; and Tong, H. 2011. Centralities in large networks: Algorithms and observations. In *SDM*, 119–130.

Kang, U.; Tsourakakis, C. E.; and Faloutsos, C. 2009. PEGASUS: A peta-scale graph mining system. In *ICDM*, 229–238.

Kyrola, A.; Blelloch, G. E.; and Guestrin, C. 2012. Graphchi: Large-scale graph computation on just a PC. In *OSDI*, 31–46.

Park, H.; Park, N.; Myaeng, S.; and Kang, U. 2016. Partition aware connected component computation in distributed systems. In *ICDM*, 420–429.

Park, C.; Park, H.; Yoon, M.; and Kang, U. 2017. PMV: pre-partitioned generalized matrix-vector multiplication for scalable graph mining. *CoRR* abs/1709.09099.

Park, H.; Myaeng, S.; and Kang, U. 2016. PTE: enumerating trillion triangles on distributed systems. In *KDD*, 1115–1124.