Ritwik Takkar
11/2/22

Paper Report:
Pregel: A System for Large-Scale Graph
Processing by Grzegorz Malewicz et al.

ritwiktakkar.com

1 Summary

Myriad computing problems have to do with graphs like, for example, the Web graph and social networks that lend their graphs to be large in size. As such, they struggle to be processed efficiently by traditional models. Pregel is a novel model designed by Google for efficient processing of large graphs through an expressive and easy to program framework. The paper describes a production quality, scalable, fault-tolerant implementation. Users report that switching to “think like a vertex” mode supports leveraging the model by thinking in terms of the API for their needs.

2 Strengths of the paper

The authors discussed the four existing options to implement an algorithm to process large graphs and why none of them met the authors’ pruposes. Some papers I read do leave me wondering whether the authors exhausted their search for existing implementations or tools to accomplish what their new, proposed method provides. I was not left wondering that with this paper.

Given MapReduce’s popularity within the development, especially systems community, the authors did well to analogize their “vertex-centric approach” in that “users focus on a local action, processing each item independently, and the system composes these actions to lift computation to a large dataset.”

I always enjoy reading the experiments conducted by massive companies like Google because it is sort of unrelatable to folks in academia given the difference in resource availability. For example, Figures 7-9 depict results based on scaling Pregel across 300 multi-core machines.

3 Weakness of the paper

It seems Pregel prioritizes correctness of computation over speed. Obviously, this is a tradeoff for the authors to choose. But I was left a little curious as to why the authors chose to make that decision exactly and what, if any, routes for future work exist to improve the speed of this framework.

4 Future work opportunities

Pregel’s authors note a practical concern, i.e., they are no longer free to change the API without considering compatibility given the program becoming a piece of production infrastructure for the user base. However, they also mention that the interface has been designed in a sufficeintly abstract manner providign flexibility to be “resilient to the further evolution of the underlying system.” I’d be curious to see what aspects of Pregel were modified and how, and whether overtime, which community is Pregel most popular with and why.