Referee Report

Paper Number:
Reviewer's Name: Jesse Bannon
Name of Paper: To Push or To Pull: On Reducing Communications and Synchronization in Graph Computations
Author(s): Maciej Besta, Michal Podstawaski, Linus Groner, Edgar Solomonik, Torsten Hoefler

Section I. Overview

A. Reader Interest

1. Which category describes this manuscript?

(x) Practice / Application / Case Study / Experience Report
( ) Research / Technology
( ) Survey / Tutorial / How-To

2. How relevant is this manuscript to the readers of this periodical? Please explain your rating under IIIA.

( ) Very Relevant
(x) Relevant
( ) Interesting - but not very relevant
( ) Irrelevant

B. Content

1. Please explain how this manuscript advances this field of research and / or contributes something new to the literature. Please explain your answer under IIIA. Public Comments.

2. Is the manuscript technically sound? Please explain your answer under IIIA. Public Comments.

( ) Yes
(x) Appears to be - but didn't check completely
( ) Partially
( ) No

C. Presentation

1. Are the title, abstract, and keywords appropriate? Please explain your answer under IIIA. Public Comments.

(x) Yes
( ) No

2. Does the manuscript contain sufficient and appropriate references? Please explain your answer under IIIA.

(x) References are sufficient and appropriate
( ) Important references are missing; more references are needed
( ) Number of references are excessive

3. Does the introduction state the objectives of the manuscript in terms that encourage the reader to read on? Please explain your answer under IIIA. Public Comments.

( ) Yes
(x) Could be improved
( ) No

4. How would you rate the organization of the manuscript? Is it focused? Is the length appropriate for the topic? Please explain your answer under IIIA. Public Comments.

( ) Satisfactory
(x) Could be improved
( ) Poor

5. Please rate and comment on the readability of this manuscript. Please explain your answer under IIIA.

(x) Easy to read
( ) Readable - but requires some effort to understand
( ) Difficult to read and understand
( ) Unreadable

Section II. Summary and Recommendation

A. Evaluation

Please rate the manuscript. Please explain your answer under IIIA. Public Comments.

( ) Award Quality
( ) Excellent
( ) Good
(x) Fair
( ) Poor

B. Recommendation

Please make your recommendation. Please explain your answer under IIIA. Public Comments.

( ) Accept with no changes
( ) Author should prepare a minor revision
(x) Author should prepare a major revision for a second review
( ) Reject

Section III. Detailed Comments

A. Public Comments (these will be made available to the author)

Explanation for the Recommendation

The implications of pushing or pulling provides informative insight on performance for various algorithms. The contribution allows readers to take these findings into account when implementing their own graph algorithms, and determine which variation would better suite their needs. However, the structure of the paper and the performance evaluation require a major revision before acceptance.

In the introduction, the graphic of the greedy switch variant implies the paper will focus on using the results of pushing or pulling to develop generic strategies that enable higher performance. Section V is dedicated to these findings, but very few are actually portrayed within the performance analysis. Greedy switch is only given a single table in section VI, and performs better only on two out of five graphs. If there is more data showing performance gains for these acceleration techniques, they should be shown in the performance evaluation with a brief note on why they perform better for certain graphs or algorithms. Otherwise, less focus should be given to these strategies and their mention should be removed from the contribution list.

There are too many contributions listed in the introduction. The last three bullet points are redundant. They could either be mixed in with the first three bullet points or removed entirely. The last bullet point regarding algebraic formulation, while it is mentioned in section 7.1, provides almost no additional contribution. A sparse matrix is stored the same way a graph is represented in the reading: an array of contiguous arrays representing a vertex's neighbors. This section could be reduced substantially, and removed from the contribution list.

The performance evaluation requires the most revision. The algorithms presented in previous sections deserved a better presentation in their performance. A few examples include metrics with Hyper-Threading disabled, runtime variance, and more runs on algorithms other than PageRank. Adding more information to the performance evaluation can simultaneously focus the reading more-so on the first three (main) contributions while removing unneeded sections such as 7.1 and 7.4.

Summary of the Paper and Assessment

To Push or To Pull gives new insights to a suite of graph algorithms, whether or not they perform better using the simple push-pull dichotomy. Every parallel graph algorithm relies on sending neighboring vertices information to one another. The communication and synchronization implications of a vertex sending its own information to others or requesting neighboring vertices' information can severely impact performance. This reading uses the Parallel Random Access Machine (PRAM) model to characterize runtime and synchronization complexities for various concurrent (or exclusive) systems which is applicable for both local machines and distributed clusters.

The abstract and introduction also imply that these insights have lead to new strategies which further increase performance. This is true to an extent: only a single strategy (partition-awareness) enhances performance consistently, while the other three provide no meaningful speedup.

Each algorithm is given a brief explanation, later followed by its theoretical analysis for pushing and pulling on all variants of the PRAM model. Source code is included for each algorithm with meaningful notation describing synchronization, which significantly aids the analysis. The ending subsection of the theoretical analysis summarizes the reoccurring findings. This should have been mentioned at the beginning of the section, to inform the reader what to expect before proceeding.

A brief section is dedicated for accelerating pushing and pulling. The only strategy which provided meaningful results is partition awareness.

Performance evaluation was conducted on multiple Cray nodes, each with 64 GiB or 32 GiB of RAM, as well as a Trivium server with specs similar to a desktop. Hardware counters were used to measure four of the eleven graph algorithms for a single Cray XC30 node, to compare the differences between push, push + partition awareness, and pull. The majority of the remaining graphs show strong/weak-scaling runtimes for algorithms, showcasing the differences in pushing or pulling. Different algorithms benefit from either implementation.

The main contribution of this reading provided insights to the implications of pushing or pulling for a variety of graph algorithms. Their findings within their theoretical analysis showed how pushing or pulling impacts synchronization and cache locality, which consequently impacts performance.