

Design of the GraphBLAS API for C

Aydin Buluç, Tim Mattson, Scott McMillan, José Moreira, Carl Yang

Abstract—The GraphBLAS effort aims to standardize linear-algebraic building blocks for graph computations. A time consuming part of this standardization effort is to translate the mathematical specification to an actual Application Programming Interface (API) that (i) is faithful to the mathematics as much as possible and (ii) enables efficient implementations on modern hardware. This paper documents the efforts taken by the C language specification subcommittee and presents the main concepts, constructs, and objects within the GraphBLAS API.

I. INTRODUCTION

Graphs and matrices are fundamental abstractions in computer science and applied mathematics, respectively. Graphs have been used to represent matrices, and especially sparse matrices, where they are often useful to conceptualize dependencies between rows or columns. Consequently graphs became a popular abstraction in sparse matrix research [1]. Conversely and more recently, matrices have started to pay back their dues and helped increase the performance of graph algorithms due to matrices being a better match for computer architectures. Many graph algorithms have been mapped to the language of linear algebra over the years [2].

High-performance systems and libraries that allow efficient implementation of graph algorithms have been built in recent years [3], [4], [5], [6]. This had led to a concern among the community that a fragmentation of concepts and abstractions might occur. The mapping of graph algorithms into the language of matrices and vectors were sufficiently well understood that the community decided to establish an effort to standardize the fundamental operations [7]. This has led to the formation of the GraphBLAS Forum [8], a loosely coupled group of researchers and practitioners from academia, industry and federally-funded research organizations. The mathematical foundations of the GraphBLAS released first, which is summarized in a recent paper [9].

A subcommittee from the general GraphBLAS forum took the task to map the mathematics to an actual programming language. The authors of this paper form that subcommittee. We had to define the concepts, the objects, and function signatures. We had to balance multiple and often conflicting objectives: (i) simplicity and ease of use, (ii) enabling high-performance implementations, and (iii) adherence to the underlying mathematics.

This paper provides a high-level summary of the GraphBLAS application programming interface (API) specification for the C language. It also explains the rationale behind many design choices and provides an accessible introduction to the specification contents.

II. BASIC CONCEPTS

The GraphBLAS standard defines a set of matrix and vector operations over semiring algebraic structures. These operations can be used to express a wide range of graph algorithms.

The GraphBLAS C API is built on a collection of objects exposed to the C programmer as opaque data types. Functions that manipulate these objects are referred to as *methods*. These methods fully define the interface to GraphBLAS objects to create or destroy them, modify their contents, and copy the contents of opaque objects into non-opaque objects the contents of which are under direct control of the programmer.

ACKNOWLEDGMENTS

This work is supported by the Applied Mathematics Program of the DOE Office of Advanced Scientific Computing Research under contract number DE-AC02-05CH11231.

REFERENCES

- [1] A. George, J. R. Gilbert, and J. W. Liu, *Graph theory and sparse matrix computation*. Springer Science & Business Media, 2012, vol. 56.
- [2] J. Kepner and J. Gilbert, *Graph algorithms in the language of linear algebra*. SIAM, 2011, vol. 22.
- [3] A. Buluç and J. R. Gilbert, “The Combinatorial BLAS: Design, implementation, and applications,” *The International Journal of High Performance Computing Applications*, vol. 25, no. 4, pp. 496 – 509, 2011.
- [4] V. Gadepally, J. Bolewski, D. Hook, D. Hutchison, B. Miller, and J. Kepner, “Graphulo: Linear algebra graph kernels for nosql databases,” in *International Parallel and Distributed Processing Symposium Workshop (IPDPSW)*. IEEE, 2015, pp. 822–830.
- [5] K. Ekanadham, W. P. Horn, M. Kumar, J. Jann, J. Moreira, P. Pattanaik, M. Serrano, G. Tanase, and H. Yu, “Graph Programming Interface (GPI): A linear algebra programming model for large scale graph computations,” in *Proceedings of the ACM International Conference on Computing Frontiers*, ser. CF ’16. New York, NY, USA: ACM, 2016, pp. 72–81.
- [6] N. Sundaram, N. Satish, M. M. A. Patwary, S. R. Dulloor, M. J. Anderson, S. G. Vadlamudi, D. Das, and P. Dubey, “Graphmat: High performance graph analytics made productive,” *Proceedings of the VLDB Endowment*, vol. 8, no. 11, pp. 1214–1225, 2015.
- [7] T. Mattson, D. Bader, J. Berry, A. Buluç, J. Dongarra, C. Faloutsos, J. Feo, J. Gilbert, J. Gonzalez, B. Hendrickson, J. Kepner, C. Leiserson, A. Lumsdaine, D. Padua, S. Poole, S. Reinhardt, M. Stonebraker, S. Wallach, and A. Yoo, “Standards for graph algorithm primitives,” in *High Performance Extreme Computing Conference (HPEC ’13)*. IEEE, 2013, (position paper).
- [8] “The graphblas forum,” <http://graphblas.org/>.
- [9] J. Kepner, P. Aaltonen, D. Bader, A. Buluç, F. Franchetti, J. Gilbert, D. Hutchison, M. Kumar, A. Lumsdaine, H. Meyerhenke, S. McMillan, J. Moreira, J. Owens, C. Yang, M. Zalewski, and T. Mattson, “Mathematical foundations of the GraphBLAS,” in *IEEE High Performance Extreme Computing (HPEC)*, 2016.