Graph Algorithms for Everyone, Everywhere

Daan de Graaf [1], Robert Brijder [1], Nikolay Yakovets [1]

[1] Database Group, Eindhoven University of Technology, Netherlands
{d.j.a.d.graaf,r.brijder,hush}@tue.nl

We recently introduced GraphAlg [1], a domain-specific language for graph algorithms that compiles to relational algebra. We demonstrated that GraphAlg is a viable solution to algorithm support in databases by integrating it into the AvantGraph [6] database, enabling custom algorithm execution without moving data to external tools. Our ongoing work brings GraphAlg to users whose data is stuck in other database systems without algorithm support.

To meet users where they are, we propose to integrate GraphAlg into other database management systems including DuckDB [5], LingoDB [3] and Umbra [4]. Because GraphAlg compiles to (an extension of) relational algebra, relational query engines only require minimal changes to execute GraphAlg programs. Furthermore, we provide the GraphAlg compiler that handles parsing, type checking and high-level optimization as a standalone library for integrations to leverage.

Another issue in driving adoption is the availability of accessible learning resources for GraphAlg. While GraphAlg is based on familiar linear algebra operations such as matrix multiplication, the task of learning any new programming language can be daunting for users. To address this, we provide the *GraphAlg Playground*[1]: An interactive tutorial on programming in GraphAlg. The GraphAlg Playground includes an online development environment that can be used to experiment with algorithms before deploying them in database systems:

1. a code editor with syntax highlighting and integrated error diagnostics;
2. a fully-featured GraphAlg compiler, running locally in the browser; and
3. a WebAssembly [2] backend to run GraphAlg programs in the browser.

Our work makes GraphAlg more easily *available* as well as more *accessible*, thus enabling user-defined algorithm execution for many more users. We show that just like AvantGraph, other database systems can be turned into highly programmable platforms for convenient and efficient graph analytics.

References

[1] Daan de Graaf, Robert Brijder, Soham Chakraborty, George Fletcher, Bram van de Wall, and Nikolay Yakovets. Algorithm support for graph databases, done right. (under review).

[2] Andreas Haas, Andreas Rossberg, Derek L. Schuff, Ben L. Titzer, Michael Holman, Dan Gohman, Luke Wagner, Alon Zakai, and JF Bastien. Bringing the web up to speed with WebAssembly. In *Proceedings of the 38th ACM SIGPLAN Conference on Programming*

[1] https://wildarch.dev/graphalg

Language Design and Implementation, PLDI 2017, pages 185–200, New York, NY, USA, June 2017. Association for Computing Machinery.

[3] Michael Jungmair, André Kohn, and Jana Giceva. Designing an Open Framework for Query Optimization and Compilation. *Proceedings of the VLDB Endowment*, 15(11):2389–2401, July 2022.

[4] Thomas Neumann and Michael J. Freitag. Umbra: A disk-based system with in-memory performance. In *10th Conference on Innovative Data Systems Research, CIDR 2020, Amsterdam, The Netherlands, January 12-15, 2020, Online Proceedings*. www.cidrdb.org, 2020.

[5] Mark Raasveldt and Hannes Mühleisen. DuckDB: an Embeddable Analytical Database. In *Proceedings of the 2019 International Conference on Management of Data*, SIGMOD '19, pages 1981–1984, New York, NY, USA, June 2019. Association for Computing Machinery.

[6] Wilco van Leeuwen, Thomas Mulder, Bram van de Wall, George Fletcher, and Nikolay Yakovets. AvantGraph query processing engine. *Proceedings of the VLDB Endowment*, 15(12):3698–3701, August 2022.