Graphalg: Efficient Execution of User-Provided Graph Algorithms in a Graph Database

Daan de Graaf [1], Bram van de Wall [1], Soham Chakraborty [2], and Nikolay Yakovets [1]

[1] Database Group, Eindhoven University of Technology, Netherlands
{d.j.a.d.graaf,a.a.g.v.d.wall,hush}@tue.nl
[2] Programming Languages Group, Delft University of Technology, Netherlands
s.s.chakraborty@tudelft.nl

As graph databases grow in popularity, they are used to answer increasingly diverse and complex queries. Graph query languages typically focus on subgraph matching. This makes them excellent tools for some classes of queries, but they are not expressive enough to express arbitrary algorithms. As a result, many users treat the database as a storage layer to export data from and develop algorithms in external tools, wasting computation power and storage space.

We present *Graphalg*, a domain-specific language for writing graph algorithms. Graphalg is fully integrated into the AvantGraph [2] database: Graphalg programs can be embedded into regular Cypher queries and are executed directly by the database.

Our language is based on linear algebra, a proven approach for graph analytics [1], but a novel strategy in graph databases. Using familiar linear algebra operations like matrix multiplication, a wide variety of graph algorithms can be concisely expressed in Graphalg.

We aim for Graphalg to not only be a high-level and powerful language but also one that can be efficiently executed by a database. Avantgraph transforms queries and Graphalg programs into a unified representation that is holistically optimized. This enables *cross-optimization*: optimizations that cross the boundary between query and algorithm, and that would not be possible with separate processing pipelines.

Our work represents a significant jump in the programmability of graph databases, blurring the line between graph databases and graph analytics frameworks. Graphalg offers functionality similar to a graph analytics framework but with a more convenient interface and more opportunities for optimization. We show that graph databases need not be limited tools for answering simple queries. Instead, they can be powerful and highly programmable platforms for efficient large-scale data analysis.

References

[1] Timothy A. Davis. Algorithm 1000: SuiteSparse:GraphBLAS: Graph Algorithms in the Language of Sparse Linear Algebra. *ACM Transactions on Mathematical Software*, 45(4):44:1–44:25, December 2019.

[2] Wilco V. 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.