In Situ Graph Querying and Analytics with GraphGen

Extended Abstract

Amol Deshpande

University of Maryland, College Park, MD

amol@cs.umd.edu

ABSTRACT

After several decades of research but limited adoption in practice, graph querying and analytics are finally starting to gain a foothold in the data management landscape. This is driven to a large degree by the increasing desire to model and query the key *entities* and the *interconnections* between them explicitly; and the observation that the use of network science, graph algorithms, and graph mining can lead to crucial new insights that are not accessible without reasoning about those interconnections in a holistic and collective manner. In addition to the traditional application domains like social media, Web, biological networks, and RDF knowledge bases, graph data models are a natural fit for new and important application domains like personal data management, provenance, metadata management, machine learning, and others. Even data that is not naturally graph-structured is increasingly viewed from the graph lense, examples being shopping transaction data, healthcare data, source code repositories, parcel shipment data, etc.

This increased attention has led to much recent work on specialized graph databases (e.g., Neo4j, Titan, OrientDB, Dgraph, Blaze-graph, Amazon Neptune), and graph analysis frameworks (e.g., Giraph, GraphLab, Ligma, GraphX, and numerous others [?]). Several established relational databases have added support for limited forms of graph querying and analytics including SAP HANA, Oracle, Aster Data, and SQL Server. However, there is no clear unifying or overarching theme that connects these systems and other research on graph querying and analytics, resulting in a highly fragmented landscape. We contend that the key reasons for this are two-fold. First, graph querying/analytics is typically a small part of the overall data management process, and many of the solutions (especially graph databases) require complete buy-in so that the data can be stored in an appropriate graph-aware format and appropriate indexes can be built. Second, the querying or analysis workloads typically considered to be within the scope of graph data management are highly varied, and include point queries (e.g., pattern matching, reachability, shortest paths), network science (e.g., community detection, centrality), graph mining (e.g., influence propagation, similarity-based ranking), graph algorithms (e.g., bipartite matching, min-cut), temporal analytics (e.g., network evolution), “what-if” analytics (e.g., vulnerability analysis), among others. As a result, many users prefer crafting custom solutions to solve specific

problems in the context of their environments, that use custom data structures that are not reusable or standardized.

In this talk, I will present our vision for an *in situ graph querying and analytics framework*, called **GraphGen**, that uses the familiar abstraction of “views” or “data virtualization” to: (a) construct graphs by combining data from a number of heterogeneous data sources, and (b) query and analyze them using powerful, flexible APIs. I will discuss how this framework can serve as a unifying abstraction that covers the spectrum of use cases and approaches above. Our focus in the work done so far has been on the common scenario where the data originally resides in an RDBMS [? ? ?]; I will describe our prototype that enables users to declaratively specify graph extraction tasks over such data, visually explore the extracted graphs, and write and execute graph algorithms over them, either directly or using existing graph libraries like the NetworkX Python library. GraphGen has a fundamentally different goal from recent work on using RDBMSs to store graph data through “shredding”. Instead, GraphGen is intended to analyze “hidden” graphs that are present in existing databases (relational or not). GraphGen attempts to utilize the underlying systems to the full extent possible by pushing down computation, uses a novel condensed representation to handle graphs that may be too large to extract in their entirety, allows writing programs using a general subgraph-centric API, and features several optimizations for efficient extraction and querying of large graphs. I will conclude with a discussion of new optimization opportunities and open challenges.

CCS CONCEPTS

• **Information systems** → **Graph-based database models**; *Data-base query processing*; *Online analytical processing engines*;

KEYWORDS

Graph databases; Graph analytics

ACM Reference Format:

Amol Deshpande. 2018. In Situ Graph Querying and Analytics with GraphGen. In *Proceedings of 1st Joint International Workshop on Graph Data Management Experiences & Systems (GRADES) and Network Data Analytics (NDA)*, Houston, TX, USA, June 10–15, 2018 (GRADES-NDA’18), ?? pages. <https://doi.org/10.1145/3210259.3210261>

ABOUT THE SPEAKER

Amol Deshpande is a Professor in the Department of Computer Science at the University of Maryland at College Park, with a joint appointment in the University of Maryland Institute for Advanced Computer Studies (UMIACS). He received his MS and PhD Degrees in Computer Science from the University of California at Berkeley in 2001 and 2004 respectively, and his B.Tech., Degree from the Indian Institute of Technology, Bombay in 1998. His research projects

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

GRADES-NDA’18, June 10–15, 2018, Houston, TX, USA

© 2018 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-5695-4/18/06.

<https://doi.org/10.1145/3210259.3210261>

have spanned a spectrum of data management topics including graph databases, big data analytics, collaborative data science platforms, data management for IoT, and query processing. He has authored more than 80 research publications, and has received best paper awards at the VLDB 2004, EWSN 2008, and VLDB 2009 conferences. He received a National Science Foundation (NSF) CAREER award in 2006, which is given to outstanding junior faculty who exemplify the role of teacher-scholars so they can further their research in their areas of expertise. He has served as a program committee member for many conferences in databases and sensor networks, including as PC Co-Chair for SIGMOD 2019 and as a Vice Chair/Group Leader for SIGMOD 2016, PVLDB 2017 and 2019, and

ICDE 2011 and 2013. He is on the editorial boards for the Journal of Distributed and Parallel Databases, ACM TODS Journal, VLDB Journal, and IEEE Transactions on Cloud Computing.

REFERENCES

- [1] K. Xirogiannopoulos and A. Deshpande. 2017. Extracting and Analyzing Hidden Graphs from Relational Databases. In *SIGMOD*.
- [2] K. Xirogiannopoulos, U. Khurana, and A. Deshpande. 2015. GraphGen: exploring interesting graphs in relational data (Demo proposal). *PVLDB* 8, 12 (2015).
- [3] K. Xirogiannopoulos, V. Srinivas, and A. Deshpande. 2017. GraphGen: Adaptive Graph Processing using Relational Databases. In *Proceedings of the GRADES@SIGMOD/PODS Workshop*.
- [4] Da Yan, Yingyi Bu, Yuanyuan Tian, and Amol Deshpande. 2017. Big Graph Analytics Platforms. *Foundations and Trends in Databases* 7, 1–195.