

# Graph Data Management Systems for New Application Domains

Philippe Cudré-Mauroux  
University of Fribourg,  
Switzerland  
pcm@unifr.ch

Sameh Elnikety  
Microsoft Research,  
Redmond, WA, USA  
samehe@microsoft.com

## ABSTRACT

Graph data management has long been a topic of interest for database researchers. The topic gained renewed interest recently, motivated by the rapid emergence of new application domains including social networks and the Web of data. This tutorial characterizes graph data management techniques and categorizes recent graph data management systems. In this context, we focus on the management of very large graphs such as social networks or the Web of data, rather than on the management of many smaller graphs (which frequently appear in bioinformatics and cheminformatics). The first part of this tutorial describes the requirements imposed by new application domains, and provides a classification of recent systems according to their data and computation models. Our classification also highlights the main representations used to store the graph (dense/sparse native graphs, triple storage or relational layouts), and the access patterns and typical queries considered (reachability or neighborhood queries, updates versus reads, transactional requirements and graph consistency models). In the second part of this tutorial, we map the data and computation models to concrete graph management systems, highlighting target application domains, implementation techniques, scalability and workload requirements. We pay special attentions to declarative models that allow query optimization (as performed in Horton [1] and Neo4j [2]), and contrast them to procedural models (such as Pregel [3]), which are more general but severely limit optimizations.

## 1. INTRODUCTION

Both academia and industry show a renewed interest in graph data management, following the adoption of graph models in several new applications domains. Two of those domains are from our perspective of particular interest to the database community: social networks and the Web of data, which both generate extremely large and dynamic graphs that cannot be handled by legacy graph management applications. This recently led to an explosion of new systems [4] and approaches to manage large graph databases. Though tackling the same fundamental problems (e.g., the management of very large graphs), both domains focus on distinct graph models and characteristics.

Permission to make digital or hard copies of all or part 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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Articles from this volume were invited to present their results at The 37th International Conference on Very Large Data Bases, August 29th - September 3rd 2011, Seattle, Washington.  
*Proceedings of the VLDB Endowment*, Vol. 4, No. 12  
Copyright 2011 VLDB Endowment 2150-8097/11/08... \$ 10.00.

## 2. SOCIAL NETWORKS

Social networks are among the most popular sites on the Web and continue to grow rapidly. They provide mechanisms to establish identities, create relationships, and share information. The resulting social graph provides a basis for communicating, distributing and locating content. The social graph, also called social network, has several properties. Nodes have several types representing entities such as people, photos or videos, while edges represent relationships among nodes, such as friendships or tagging information. Social networks are large, scale-free graphs: the number of nodes and edges are in the billions. They are scale-free due to the popularity distribution of their nodes; some nodes have more connections than others, leading to a power-law distribution of node degrees.

Social applications issue queries and transactions against the social network. Their workloads include both read and update queries. Updates include primary-key updates mostly (e.g., adding a new person or adding a friendship edge between two people, tagging a friend in a photo, or updating a status). Reads dominate the workload. They range from simple look-ups and one-hop network reads to complex reachability queries such as finding the shortest path between two nodes or searching for nodes that satisfy certain predicates along a path starting from seed nodes. Some of these queries can be expressed as SQL queries, while other (such as reachability queries) cannot.

To manage social network data, systems either use a relational engine, which comes with convenient features such as indexes and transactional support, or use a native graph engine. In the former case, reachability queries are rewritten, for example by limiting the path length expressed in SQL. In the latter case, new systems storing and managing graph natively using nodes and edges are built from scratch, which can then support more expressive queries (such as reachability and betweenness centrality queries), but which also leads to a complete reengineering of the query execution and query optimization engines.

Representative social queries include the following:

- Updates on individual nodes (e.g., to change the status of a given user).
- Edge inserts (e.g., to connect users to new photos or webpages).
- Scope queries (e.g., to retrieve all nodes directly related to a user).
- Shortest-path queries (e.g., to find how two users are related).

- Reachability queries (e.g., to find all employees reporting directly or indirectly to the CTO).

### 3. WEB OF DATA

Structured data is increasingly created, transmitted, and archived across the Web. One of the most important movements in this context, the Linked Data movement [5] [6], promulgates the rapid online publication of large-scale structured data following three basic principles: i) the creation of a Unique Resource Identifier (URI) for each resource published, ii) the publication of structured data (typically expressed as RDF triples [7]) at the URI corresponding to the resource, and iii) the inclusion of links relating each resource to similar online resources. Hence, a real Web of data is emerging for the first time on the Internet, creating a gigantic graph of related data items.

Publication of Linked Data is currently booming on the Web, fostered by important governmental agencies, companies, and scientists from various domains. Unrelated communities of users are increasingly interested in bulk-exporting, querying, and integrating heterogeneous data across the Web using this formalism. Database administrators either use legacy relational systems or native triple stores to manage Linked Data. They typically provide a SPARQL [8] interface to allow third parties to pose queries against their data online.

Typical operations on Web data graphs include:

- Large bulk inserts (e.g., from relational databases [9]).
- Triple pattern look-ups (e.g., to find a particular node based on constraints on its direct neighbors and on its edges).
- Distributed entity retrieval queries (e.g., to retrieve information from interlinked entities stored in separate databases).
- Molecule queries (e.g., to retrieve all the nodes and edges directly or indirectly related to a given node).
- Path queries (e.g., to identify all pairs of nodes connected by a given series of edges).
- Inference queries (e.g., to retrieve nodes based on the logical properties defined by their edges).

While some queries are similar for both applications (e.g., scope queries and molecule queries), most operations are in the end fairly different between the two domains. Social networks are often concerned with punctual updates, transitive closures and betweenness queries, while Web of data queries often revolve around edge look-ups, joins, and logical inference based on edges.

### 4. TUTORIAL OUTLINE

The tutorial we outline below discusses the use of graph data management systems for both social networks and the Web of data. It suggests a taxonomy of recent graph data management systems based on the fundamental differences exposed above, and discusses some of their implication in terms of graph models and query execution.

The first part of the tutorial starts with a series of motivating examples for both application domains. We then cover the most relevant graph queries and manipulation techniques for both domains, covering inserts and updates, distributed and pattern look-ups, path and reachability queries, network analysis queries, and sub-graph selects (e.g., scope or molecule queries). We

conclude the first part of the tutorial by discussing statistical and inference queries.

The second part of the tutorial gives an overview of recent advances in graph data management. We start this second part by proposing a taxonomy of graph data management systems taking into account graph models (covering nodes as entities, edges as entities, and semi-structured models), storage and indexing techniques, and query interfaces. We then introduce a series of recent graph data management systems, focusing on one key feature for each system. Finally, we conclude the second part by introducing current research directions such as distributed graph management techniques, graph consistency models, graph recommender systems, and multi-graph computations.

### 5. PRESENTERS

Philippe Cudré-Mauroux is an Associate NSF Professor at the University of Fribourg in Switzerland. Previously, he was a postdoctoral associate working in the database systems group at MIT. He received his Ph.D. from the Swiss Federal Institute of Technology in Lausanne (EPFL), where he won both the Doctorate Award and the EPFL Press Mention in 2007. Before joining the University of Fribourg, he worked on distributed information management systems for HP, IBM T.J. Watson Research, and Microsoft Research Asia. His research interests are in exascale data management and infrastructures for non-relational data. He was the main investigator of the GridVine decentralized RDF storage system and is currently building dipLODocus[RDF], a scalable and efficient back-end to store and analyze very large graphs of Web data in the cloud. He will be PC Chair of the International Semantic Web conference in 2012 in Boston.

Sameh Elnikety is a researcher at Microsoft Research in Redmond, Washington. He received his Ph.D. from the Swiss Federal Institute of Technology (EPFL) in Lausanne, Switzerland, and a M.Sc. from Rice University in Houston, Texas. His research interests include social network systems, graph databases, and large-scale software systems. Sameh is currently building Horton, a distributed system that manages large graphs on commodity servers while providing a declarative query language with multi-version transactional support. Sameh Elnikety is the PC Chair of Social Network Systems (SNS 2011) workshop, and the keynote speaker at Graph Data Management (GDM 2011) workshop. Sameh's work on database replication received the best paper award at Eurosys 2007.

### 6. REFERENCES

- [1] <http://research.microsoft.com/en-us/projects/ldg/>
- [2] <http://neo4j.org/>
- [3] Grzegorz Malewicz et al. Pregel: a system for large-scale graph processing. In *Proceedings of SIGMOD*, 2010, 135-146.
- [4] <http://nosql-database.org/>
- [5] <http://www.w3.org/DesignIssues/LinkedData>
- [6] <http://linkeddata.org/>
- [7] <http://www.w3.org/RDF/>
- [8] <http://www.w3.org/TR/rdf-sparql-query/>
- [9] <http://www.w3.org/2001/sw/rdb2>