Recent advances in Graph Data Management

Domagoj Vrgoč

Pontificia Universidad Católica de Chile and IMFD Chile

Abstract. In this half-day tutorial we will introduce the audience to graph databases and review a series of theoretical results that have recently been adapted by graph processing systems to handle large scale Knowledge Graphs. Throughout the tutorial we will be aiming to answer one key question: what is needed to effectively manage large Knowledge Graphs? The tutorial is divided into four largely independent sections. In the first section we will introduce attendees to popular graph data models and query languages and try to bridge the gap between RDF and property graphs. The second section will overview worst-case optimal algorithms, a recent advancement in join processing, and show how this theoretical concept can be adapted in practice to evaluate basic graph patterns. The third section deals with path queries and shows how to adapt an old idea from database theory to efficiently compute a representation of a potentially exponential set of paths in linear time. In the final section we introduce MillenniumDB, a system build on these ideas with the goal of managing Wikidata and discuss what it takes to make theory work in practice at this scale.

1 Overview and relevance

Graph databases have received a lot of attention in recent years, mostly due to their role as the underlying storage and query mechanism for knowledge graphs. This led to different graph solution being developed throughout the years, most notably the RDF data format together with the SPARQL query standard, which are being widely used in large open knowledge graphs such as Wikidata or DBpedia, while commercial systems generally deploy the property graph data model, with the recent GQL ISO standard formalizing query languages in this setting.

The objective of this tutorial is to provide a detailed overview of this landscape, and give an in-depth explanation of recent developments in graph query processing. As such, the tutorial will be divided into four equal length parts, the first one serving as an introduction to the topic, surveying prominent data models and query languages. In Parts 2 & 3 we then take a deep-dive into contemporary techniques for evaluating the most common classes of graph queries: those involving finding a pattern in the graph, and those involving finding paths. Finally, in Part 4 we present MillenniumDB, a system build on these principles, supporting both RDF/SPARQL and property graphs/GQL. Here we illustrate the challenges required to bring theoretical concepts into practice, and showcase the ability of this system to host Wikidata, as well as several open knowledge graphs that have been developed at IMFD Chile.

We believe the topic to be highly relevant to the Semantic Web community, as it deals with its core technologies for storing and accessing data. The topic is timely, given that knowledge graphs are being actively used to manage diverse datasets, and many new techniques for processing graph queries are currently emerging, with the potential to solve long standing performance issues that many systems face. Equally important, given that SPARQL 1.2 is currently in the process of being standardized, and the GQL standard for property graphs has just been published, we would like to highlight novel features that can be shared between the two and point to efficient methods for evaluating them.

2 Content and schedule

The tutorial is divided into four 50 minute sections. Each section will contain a dedicated discussion/hands-on portion as detailed below.

Part 1: Graph Data Models and Query Languages. [50 min] In this introductory segment we explain what graph databases are and how they provide the backbone for knowledge graphs on the Web. Following this, we will explore the main paradigms for modelling graph databases: RDF and property graphs, and explain why, at both the modelling level and the implementation level, they can be viewed as the same data model. We will then open a discussion on the suitability of these approaches for modelling complex datasets that require higher arity relations, and discuss some alternatives. Following this, we continue by an overview of graph query languages, aiming to explain what separates them from relational databases and SQL. For this, we first give an abstract view of main graph query features, which are commonly grouped into three categories: (i) graph patterns, which look for occurrences of a small subgraph in the graph database; (ii) path queries, which navigate between graph nodes using paths; and (iii) complex graph queries, which combine and extend the two with other features. To ground the discussion, we then turn to concrete languages, focusing on SPARQL for the RDF data format, and presenting the recent GQL ISO standard for property graph querying to cover the broad spectrum of languages available over this data model. Throughout this segment we will put a special focus on knowledge graphs that the audience can try out for themselves through their Web browser. Concretely, we will use the Wikidata public SPARQL endpoint, and several other public query endpoints which are hosted at IMFD Chile.

- *Interaction with the audience.* There will be a 15 minute discussion on the suitability of RDF and property graphs to model complex knowledge graphs such as Wikidata. Additionally, we will have a hands-on session for trying out queries at different endpoints (with introductory examples provided).

Part 2: Graph Patterns and WCO algorithms. [50 min] In the second part of the tutorial we explain how to evaluate graph patterns, which are a common feature of all graph query languages. In essence, graph patterns can be viewed as relational joins, but typically involve many more joins than a relational query. To tackle this issue, we introduce the notion of *worst-case-optimal (wco)* join

algorithms, a recent breakthrough in join query processing, and argue why they are perfectly suited for graph databases. To make the presentation concrete, we will illustrate the *Leapfrog* wco-algorithm, which is implemented by several relational and graph systems. We will then analyse the performance of wco algorithms compared to standard join processing techniques. We will conclude by explaining the main drawbacks of wco algorithms: the large memory footprint they require to store multiple permutations of base relations, and what this means in terms of data maintenance and transactions/updates.

- *Interaction with the audience.* Following a series of examples illustrating where the wco-behavior kicks-in, the audience will be asked to provide sample queries that work well in a wco-based system versus a traditional implementation. For this, a Wikidata endpoint hosting MillenniumDB, a wco-based graph engine, will be used (see https://wikidata.imfd.cl/).

Part 3: Path Queries. [50 min] One of the key query features separating graph databases from their relational counterpart are path queries. At their core, path queries explore how graph nodes are connected by paths conforming to some constraint specified by the query. While present in most modern graph query languages and standards, path queries still pose many challenges in terms of efficiency due to the fact that the number of paths matching a query can easily become exponential of even infinite. To overcome this challenge, we will present how an idea from early theoretical work in the area can be used to handle both SPARQL property paths, and the requirement to return different number of paths of a specific type supported in the property graph query standard GQL. We will start by explaining the theoretical framework which is based around the notion of automata guided graph-search and enumeration algorithms. We will then follow up by explaining implementation challenges for these techniques, particularly as they relate to the underlying storage model of the database, and finish with a detailed performance comparison with existing approaches. We will also review main open problems in the area, particularly relating to novel features supported by the GQL standard.

- *Interaction with the audience.* During this segment the audience will be asked to construct specific queries running over several public endpoints we will host using both SPARQL and GQL. There will also be a discussion on the challenges of extending SPARQL with the ability to return paths.

Part 4: The MillenniumDB graph engine. [50 min] In the final section we present our experience building a graph engine based on the ideas presented in this tutorial. In particular, we will highlight the challenges that need to be overcome to transfer theoretical ideas into practice and will explain what it took for the engine to be able to process Wikidata or datasets of similar size. During the process we will overview the architecture of the system, called MillenniumDB, show how it supports different data models from Part 1, show how it deploys wco-algorithms from Part 2 and how we deal with the cost they incur in terms of storage and indexing, and finally explain the structure of the engine's path processing module from Part 3. Given that MillenniumDB supports

both RDF/SPARQL and property graphs/GQL, we will discuss the interaction between the two and demonstrate how the engine operates in the context of Wikidata and several other datasets that we are already hosting. We will finish by explaining more advanced features such as similarity indices and integration with Web interfaces and programming languages.

– *Interaction with the audience.* During this segment the audience will be able to interact with knowledge graphs running on MillenniumDB via several public endpoints. Additionally, we will hold an open discussion on features that the audience sees as lacking in our solution (or other modern engines), and what they expect from a knowledge graph system in general.

3 Formal details

Format: The tutorial is intended to be presented in a half-day format.

Level: The tutorial is presented at a beginner level and attendees with general knowledge of a standard Computer Science curriculum should be able to follow.

Prerequisite knowledge: No prerequisite knowledge is needed for part 1. General understanding of algorithms, automata, and complexity will be useful to fully understand the concepts presented in parts 2 and 3 of the tutorial.

Target audience: Part 1 is aimed at attendees that wish to learn about the technology behind knowledge graphs. Parts 2, 3 & 4 are intended for people wishing to see recent advancements in graph query processing and practitioners and engineers implementing data processing systems.

Learning outcomes: Following the tutorial attendees should have a clear idea of different graph data formats and query languages currently in use. Attending parts 2, 3 & 4 should also allow to understand new paradigms in graph query evaluation, and give developers enough knowledge to start implementing them.

Presenter information. Domagoj Vrgoč is an Assistant Professor at Pontificia Universidad Católica de Chile, and an Associate Researcher at the Millennium Institute for Foundational Research on Data (IMFD). He has over a decade of research experience in the area of graph query languages and algorithms. He was invited to deliver several tutorial presentations, including SIGMOD'24, AMW'23, SPIRE'22, and he received several awards for his work, including ICDT Test-of-time award in 2023, and SIGMOD best industry paper award in 2023. He is the project lead of the open source graph engine MillenniumDB.

Materials. Slides will be provided via open access. All the interactive material will be handled via public endpoints available through a Web browser.

Previous editions. Parts of this tutorial were previously presented at SPIRE'22, AMW'23 Summer School and will be presented at SIGMOD'24. While there is overlap with previous editions, the proposed tutorial has a strong focus on Semantic Web applications, querying Wikidata, and building scalable systems, making it more applied in nature.