openCypher Queries over Combined RDF and LPG Data in Amazon Neptune

Willem Broekema[1], Mohamed Elzarei, Ora Lassila, Carlos Manuel Lopez Enriquez, Marcin Neyman, Florian Schmedding, Michael Schmidt, Andreas Steigmiller, Geo Varkey, Gregory Todd Williams and Amanda Xiang

Amazon Neptune Team, Amazon Web Services, Seattle, WA

1. Introduction

The Semantic Web stack, with the Resource Description Framework (RDF) as foundation, envisioned enhancing the Web with a globally connected network of knowledge [1]. RDF has proven itself for data serialization, interlinking at global scale, and reasoning. In contrast, Labeled Property Graphs (LPGs) emerged organically from companies and organizations, resulting in different flavors of the LPG data model without built-in semantics and with looser constraints.

Users building graph applications today have to weigh the respective characteristics of both graph technologies, and then commit to one, with considerable switching costs. A stack determines the supported data formats and poses restrictions on the possible query expressivity. At Amazon Neptune we have seen that ontologists and data scientists typically choose RDF, whereas graph projects initiated by software engineers gravitate towards LPG.

Our proposed *OneGraph* [3] metamodel is meant to overcome the need of having to choose, and the feature now highlighted is developed in that context. With support for openCypher over RDF data in Neptune Analytics, users now have the possibility to combine RDF and LPG data in a single, integrated database, and run e.g. path queries and algorithms which they could not do before with just SPARQL, while keeping the benefit of RDF features like IRIs [2].

2. Presentation

Imagine an engineering company that tracks how parts, grouped into components, are integrated in products. RDF is suitable for expressing such hierarchies, but SPARQL does not offer paths as first class values, as shown below, so users would have to resort to LPG and openCypher path queries to see how items are connected. The same applies to modelling other sequential processes like supply chains. Another example is a virtual assistant created by developers using openCypher, where the assistant's knowledge should be enriched with internal RDF datasets maintained by ontologists, or external open RDF datasets, in order to better answer questions.

RDF and LPG have different data models that define what a graph is. In RDF a graph is built from triples, using IRIs for resources. In LPGs a graph consists of vertices, edges between

Posters, Demos, and Industry Tracks at ISWC 2024, November 13–15, 2024, Baltimore, USA
*Corresponding author: metawilm@amazon.com.

vertices, and labels and properties for vertices and edges. To overcome the structural difference, we created an interpretation so that any triple corresponds to the declaration of either a vertex (with an id and label), a vertex property, or an edge (with a label).

Another aspect was introducing support for IRIs in the openCypher query language. We reuse the existing `String` type for this, by having the query engine recognize specially formatted strings that denote IRIs. Also we support SPARQL's `PREFIX` in openCypher, to allow abbreviated IRI references in queries using `prefix::suffix` notation. And if an RDF dataset contains blank nodes, they are replaced by unique IRIs at load time (scholemization), which does not take expressive power away from users, and can be built upon the IRI functionality.

For the existing openCypher functions we had to carefully design how they interact with RDF values; and similarly equality and ordering had to be defined properly. Finally, the serialization of all RDF values as openCypher query results had to be defined, where we choose to not introduce incompabilities by e.g. not serializing custom literal datatypes.

To come back to the earlier engineering company example, this SPARQL query returns all `Car` products that include a specific part. The query won't give insight in how exactly the part is embedded in increasingly bigger components:

```
SELECT ?product {
  ?part    ex:partId    "YSH301" .
  ?part    ex:isPartOf* ?product .
  ?product a            ex:Car .
}
```

In contrast, this openCypher query returns the complete trace from part to product as a sequence of nodes and edges, easily inspected or processed further:

```
MATCH p = ( ({ex::partId : "YSH301"})-[: ex::isPartOf*0..]->(: ex::Car) )
RETURN p
```

Our presentation will also include a discussion of challenges and design choices, and the pragmatic approach taken to get a core feature set that naturally maps RDF to openCypher and LPG. These topics include: RDF's notions of *blank nodes* and *named graphs* and its rich type system (e.g. numerics, literals), LPG and RDF composite types [4], edge ids and properties, and the IRI syntax in queries.

To conclude, in this presentation we highlight standardization activities towards bridging RDF and LPG, like composite types for SPARQL and RDF, and the RDF-star Working Group [5], and our effort to align the OneGraph model and implementation with their outcomes. We believe that interoperability between data models (with LPG and RDF as a start) and query languages (SPARQL, openCypher, Gremlin, GQL) will enable customers to use graph database technology in more flexible and powerful ways.

References

[1] O. Lassila, J. Hendler, T. Berners-Lee, The semantic web, Scientific American 284 (2001) 34–43.

[2] M. Duerst, M. Suignard, Rfc 3987: Internationalized resource identifiers (iris), 2005.

[3] O. Lassila, M. Schmidt, O. Hartig, B. Bebee, D. Bechberger, W. Broekema, A. Khandelwal, K. Lawrence, C. M. Lopez Enriquez, R. Sharda, et al., The OneGraph vision: Challenges of breaking the graph model lock-in, Semantic Web 14 (2023) 125–134.

[4] O. Hartig, G. Williams, M. Schmidt, O. Lassila, C. M. L. Enriquez, B. Thompson, Datatypes for lists and maps in RDF literals, in: European Semantic Web Conference, Springer, 2024.

[5] World Wide Web Consortium, RDF-star working group (2024). URL: https://www.w3.org/groups/wg/rdf-star/, last accessed: July 2, 2024.