Graph Database Applications
Mastering the Heterogeneity Challenges

Michael Schmidt
DBOnto Workshop, May 27, 2016 (London)
Company Facts

- Founded in 2014
- Headquartered in Walldorf, Germany
- Software & projects around knowledge graph applications
- Solutions for industry, life sciences, cultural heritage, and other domains
Challenges in Knowledge Graph Application Building

- Schema heterogeneity & alignment problems
- Different data modalities (geospatial, temporal, …)
- Data residing in specialized & legacy systems
- Structured Queries vs. (Graph) Analytics

Raw data

Knowledge Graph Application

W3C SPARQL

Open for Innovation

KNIME Spark
The metaphacts Approach

Platform for Knowledge Graph Application Development

- RDF, RDFS & OWL for knowledge representation
- Graph-based -> easing integration
- Built-in semantics
- Low-level and higher-level APIs: SPARQL, LDP, REST, ...
- Choice depends on use case and requirements
- Declarative application development approach
  - HTML5 based, reusable (and mostly domain independent) semantic Web components
  - Generic, composable & standards-based

Ex.: declarative spec. of keyword search field driven by SPARQL

Results computed based on SPARQL query instantiation with user input

| Oxford (city in Oxfordshire, England) |
| Oxford (city in Calhoun and Talladega counties, Alabama, United States of America) |
| Oxford (town in Oxford County, Maine, United States) |
**Hybrid Query Scenario Challenge**

- **Challenge: supporting hybrid search**
  - Combine free-text search with structured data extraction in SPARQL endpoint
  - Reuse existing systems
    - Non-invasive approach
    - Specialized tools (e.g. for text search) often benefit from years of development & experience
  - **Goal**
    - No proprietary, coded solution
    - Still have it declarative
Custom SPARQL SERVICE Extensions

- Key idea: custom SPARQL SERVICE extensions
  - Standards-compliant syntax & clear semantics
  - Elegant & easy to understand
  - Extensible

Example: returned entities including author & type containing the search terms “London” or “Queen”, ordered by Solr score
Graph Analytics vs. Querying

- **Approach:** **unified, GPU based runtime**
  - Data graph loaded into the GPU at startup
  - Runtime provides highly efficient algebraic core operators
    1. Used to accelerate SPARQL query evaluation
    2. Used to execute graph algorithms (e.g. BFS, SSSP, PageRank, ...)
  - Own algorithms can be specified using a domain-specific functional language
    - Translated into programs over the GPU
  - Algorithms exposed as custom SPARQL SERVICE extensions

Blazegraph GPU bridges the gap between declarative SPARQL queries and functional graph analytics programs.

Example: invoking breadth-first search via custom SPARQL SERVICE extension
**metaphacts Reference Architecture**

**Data-driven services**

- Knowledge Graph management UIs (for experts)
- Knowledge graph applications (for end users)
- Semantic data connectors for external tools

**Unified API Stack**

- REST APIs (domain specific)
- Generic higher-level APIs (e.g. LDP)
- Base API: SPARQL Endpoint (implicitly incl. SERVICE extensions)

**Graph Database**

- RDF triple store
- Unified management of schema and instance-level data
- GPU-based, unified runtime for querying and graph analytics
- Extensions for geospatial & temporal data
- Open Source & extensible

**Graph Data Processing (Querying & Analytics)**

- Access via custom SPARQL SERVICE extensions

**Access via SPARQL SERVICE (federation) or one-time import**

- OBDA Endpoint
- JDBC
- MySQL, Oracle, ...

**Relational Data**

- Text Index
- Specialized & legacy systems
Key Characteristics

- **Non-invasive**
  -> no need for large-scale data migration upfront

- **Unified API stack**
  -> data access is data type & data source independent

- **Flexible & extensible**
  -> incrementally bring semantics into the enterprise

- **Standards compliant**
  -> no vendor lock-in (for core architecture)
Research @ metaphacts

What:
Design a scalable, federated semantic enterprise search system over distributed, heterogeneous data sources.

Key challenges:
• Integration of specialized and legacy systems
• Efficient federated query evaluation
• Design & implement generic APIs for search

What:
Build an open, service-based platform for management and efficient processing of sensor based geo data.

Key challenges:
• Scalable backend services for the storage, retrieval, and processing of semantic geo data
• Flexible, micro-service based architecture

Data level
• Efficiency and query optimization
• GPU acceleration & analytics

Architectural level
• Integration with Big Data frameworks (SPARK, …)

Application level
• Supporting management of semantic assets (queries, ontologies, mappings, …)
• Abstraction layers & APIs

Looking for partners to address the challenges ahead!