Scalable Integration and Processing of Linked Data

Andreas Harth, Aidan Hogan, Spyros Kotoulas, Jacopo Urbani
Outline

- Session 1: Introduction to Linked Data
  - Foundations and Architectures
  - Accessing Linked Data
  - Introduction to SPARQL
- Session 2: Integrating Web Data with Reasoning
  - Introduction to RDFS/OWL on the Web
  - Introduction and Motivation for Reasoning
- Session 3: Distributed Reasoning: Because Size Matters
  - Problems and Challenges
  - MapReduce and WebPIE
- Session 4: Putting Things Together (Demo)
  - The LarKC Platform
  - Implementing a LarKC Workflow
INTRODUCTION
Motivation

- With increased use of computers more and more data is being stored
  - Organisations rely on data for business decisions
  - Data drives policy decisions in government
  - Individuals rely on data from the Web for information and communication

- Data volumes explode
  - More and more data available on the Web is represented in Semantic Web standards
  - Linking Open Data (LOD) initiative

- Semantic Web technologies facilitate the integration of data from multiple sources
- Combining data from multiple sources enables insights
Linked Data Now!

http://www.ted.com/talks/tim_berners_lee_on_the_next_web.html
Linked Data on the Web
Linked Data on the Web
Linked Data on the Web

2008-02
Linked Data on the Web

2008-03
Linked Data on the Web

As of September 2008

2008-09
Linked Data on the Web

As of March 2009

2009-03
Linked Data on the Web

As of July 2009

2009-07
Linked Data on the Web
Linked Data on the Web
Types of Data in the Linking Open Data Cloud

http://www4.wiwiss.fu-berlin.de/lodcloud/state/ (Sept 2010)
Scenario Overview

- Semantic Technologies facilitate access to data

1. Query

Q: data about Berlin?
Q: famous people that died in Berlin?
Q: data about Hegel?
Q: Hegel’s publications?
Q: data about Marlene Dietrich?
Q: Dietrich’s songs?

2. Answer
DBpedia

- Linked Data version of Wikipedia
- Scripts that extract data (text, links, infoboxes) from Wikipedia
- Published as Linked Data
- Interlinking hub in the Linked Data web

- Berlin
  - http://dbpedia.org/resource/Berlin

- Hegel
  - http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel

- Marlene Dietrich
  - http://dbpedia.org/resource/Marlene_Dietrich
BBC Music

- Data about BBC (radio) programmes, artists, songs…
- Combination of BBC-internal data (playlists), MusicBrainz (artists, albums), Wikipedia (artists)
- Underpinning the BBC Music website
- Data published according to Linked Data principles

- Marlene Dietrich
  - http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5
Virtual International Authority File (VIAF)

- Joint project of national libraries and related organisations
  - 21 institutions, among them the Deutsche Nationalbibliothek
- Provide access to “authority files”
- Matching and interlinking collections from participating institutions

- Hegel
  - http://viaf.org/viaf/89774942
- Marlene Dietrich
  - http://viaf.org/viaf/97773925
LINKED DATA PRINCIPLES
Semantic Technologies

- Semantic Web technologies, standardised by the W3C, are mature:
  - RDF recommendation in 1999, update in 2004
  - RDFa (RDF in HTML) note in 2008
  - RDFS recommendation in 2004
  - SPARQL recommendation in 2008
  - OWL recommendation in 2004, update in 2009

- Linked Data is a subset of the Semantic Web stack, including web architecture:
  - IRI (IETF RFC 3987, 2005)
  - HTTP (IETF RFC 2616, 1999)
Linked Data Principles

1. Use URIs as names for things
2. Use HTTP URIs so that people can look up those names.
3. When someone looks up a URI, provide useful information, using the standards (RDF*, SPARQL)
4. Include links to other URIs. so that they can discover more things.

http://www.w3.org/DesignIssues/LinkedData
1. Use URIs as Names for Things

- Use a unique identifier to denote things
- URIs are defined in RFC 2396

Hegel, Georg Wilhelm Friedrich
- http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- http://viaf.org/viaf/89774942
- ...

Hegel, Georg Wilhelm Friedrich: Gesammelte Werke / Vorlesungen über die Logik
- urn:isbn:978-3-7873-1964-0
Names for Things

Now! That should clear up a few things around here!
2. Use HTTP URIs

- Enables “lookup” of URIs
- Via Hypertext Transfer Protocol (HTTP)
- Piggy-backs on hierarchical Domain Name System to guarantee uniqueness of identifiers
- Uses established HTTP infrastructure
- Connects logical level (thing) with physical level (source)
- Important: distinction between name/“thing URI” and location/“source URI” („other resource“/„non-information resource“ vs. „information resource“)
Information Resources vs. Other Resources

Marlene Dietrich, the person

Name?
Creator?
Birth date?
Last change date?
License?
Copyright?
...

File containing data about Marlene Dietrich

RDF
Correspondence between thing-URI and source-URI („hash URIs“)

User Agent

HTTP
GET

Web Server

RDF

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5#artist

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5.rdf
Hypertext Transfer Protocol (HTTP)

$ curl -H "Accept: application/rdf+xml" -v http://www.w3.org/People/Berners-Lee/card#i

Request:

GET /People/Berners-Lee/card HTTP/1.1
User-Agent: curl/7.21.0
Host: www.w3.org
Accept: application/rdf+xml

Response:

HTTP/1.1 200 OK
Date: Mon, 28 Mar 2011 17:16:30 GMT
Server: Apache/2
Content-Location: card.rdf
Content-Type: application/rdf+xml; qs=0.9
Connection: close
Correspondence between thing-URI and source-URI („slash URIs“)

User Agent

HTTP GET 303 HTTP GET RDF

Web Server

http://dbpedia.org/resource/Marlene_Dietrich

http://dbpedia.org/data/Marlene_Dietrich

http://dbpedia.org/page/Marlene_Dietrich
3. Provide Useful Information

- When somebody looks up a URI, return data using the standards (RDF*, SPARQL)
- Resource Description Framework, a format for encoding graph-structured data (with URIs to identify nodes/vertices and links/edges)
Resource Description Framework

- Directed, labeled graph
- triple(subject, predicate, object)
  - subject: URI (or blank node)
  - predicate: URI
  - object: URI (or blank node) or RDF literal (string, integer, date...)

- RDF/XML is the most widely deployed serialisation
- Other serialisations possible (N-Triples, Turtle, Notation3...)

- Quadruples (or quads) used as internal representation when integrating data
- quad(subject, predicate, object, context)
  - context: URI (used to store origin of triple)
Merging Data with RDF
4. Link to Other URIs

- Enable people (and machines) to jump from server to server
- External links vs. internal links (for any predicate)

- Using external vocabularies enables linking
- Vocabularies might be interlinked, too

- Special owl:sameAs links to denote equivalence of identifiers (useful for data merging)
Equivalences via owl:sameAs

http://viaf.org/viaf/89774942
- http://dbpedia.org/resource/Georg_Wilhelm_Friedrich_Hegel
- http://www.idref.fr/026917467/id
- http://libris.kb.se/resource/auth/190350
- http://d-nb.info/gnd/118547739

http://www.bbc.co.uk/music/artists/191cba6a-b83f-49ca-883c-02b20c7a9dd5#artist
- http://dbpedia.org/resource/Marlene_Dietrich

http://viaf.org/viaf/97773925
- http://dbpedia.org/resource/Marlene_Dietrich
- http://d-nb.info/gnd/118525565
- http://libris.kb.se/resource/auth/238817
- http://www.idref.fr/027561844/id

http://dbpedia.org/resource/Berlin
- http://mpii.de/yago/resource/Berlin
- http://data.nytimes.com/N50987186835223032381 - Berlin (Germany)
- http://www4.wiwiss.fu-berlin.de/flickrwrappr/photos/Berlin
- http://data.nytimes.com/16057429728088573361 - Gaspe Peninsula (Quebec) (?)
Benefits of Linked Data

- Explicit, simple data representation
  - Common data representation (Resource Description Framework, RDF) hides underlying technologies and systems

- Distributed System
  - Decentralised distributed ownership and control facilitates adoption and scalability

- Cross-referencing
  - Allows for linking and referencing of existing data, via reuse of URIs

- Loose coupling with common language layer
  - Large scale systems require loose coupling, via HTTP as common access protocol

- Ease of publishing and consumption
  - Simple and easy-to-use systems and technologies to facilitate uptake

- Incremental data integration
  - Start with merged RDF graphs and provide mappings as you go
Challenges

- Ramp-up cost for data conversion
  - May be alleviated by semi-automatic mappings and adequate tool support for manual conversion
- Integrated data may be messy at first
  - But can be refined as need arises
- Distributed creation and loose coordination may result in inconsistencies
  - Can be detected, diagnosed, and fixed with appropriate tools
The Pedantic Web Group

- Get the community to contact publishers about errors/issues as they arise
- Get involved: http://pedantic-web.org/
- ~200 members!

Acknowledgements to: Aidan Hogan, Alex Passant, Me, Antoine Zimmermann, Axel Polleres, Michael Hausenblas, Richard Cyganiak, Stéphane Corlosquet
Data Integration System Architecture

Integration

Wrapper 1
Source 1

Wrapper 2
Source 2

Wrapper n
Source n
Semantic Web Components

User Interface & Applications

Query: SPARQL

Data interchange:
- RDF
- XML
- URI/IRI
Linked Data: Minimal Components

User Interface & Applications

1. Query
2. Answer

Query: SPARQL
Data interchange:
RDF
XML
URI/IRI
Architecture Styles

Warehousing/
Crawl-Index-Serve

Virtual Integration/
Distributed Querying

1. Query
2. Answer

0. Crawl-Index
Basic Application: Entity Browsing

**Warehousing/Crawl-Index-Serve**

- SWSE, Falcons, Sindice, Watson, FactForge...

**Virtual Integration/Distributed Querying**

- Tabulator, Disco, Zitgist...
ACCESSING LINKED DATA
RDF Graph vs. Information Resource Graph

RDF Graph

Information Resource Graph
Information Resource Graph Explosion

Directed graph rooted in http://danbri.org/foaf.rdf
Level 0: 1
Level 1: 25\(^1\) (avg), 105\(^1\) (worst)
Level 2: 25\(^2\) (avg), 105\(^2\) (worst)
Level 3: 25\(^3\) (avg), 105\(^3\) (worst)
Level k: n\(^k\)
Linked Data Access Continuum

Materialise

LDSpider
[Isele et al. 2010]

swget
[Fionda et al. 2011]

[Hartig et al. 2009]
[Harth et al. 2010]
[Ladwig et al. 2010]
[Schmedding 2011]

Distributed Query Processing
Querying Data Across Sources

- Data warehousing or materialisation-based approaches (MAT)

- Distributed query processing approaches (DQP) over Linked Data sources

- DQP over RDF stores (not covered)
(Linked Data) Crawling

1. Get URI from a queue
2. Open connection and fetch content
3. Process and store content
4. Extract new links and put into queue
5. At defined intervals: schedule URIs in queue

LDSpider [Isele et al. 2010]

- API to access Linked Data
- Lightweight implementation in Java
- GPL at Google Code (http://code.google.com/p/ldspider)
- Multithreading
- Including RDF/XML parser
- Adheres to robots.txt protocol
- Flexible configuration/application with callbacks and hooks

- Link type specification to focus on certain parts of the Great Global Graph
- Heuristics to direct crawler to high quality sites (sort URIs in queue based on inlink count)
- Heuristics to direct crawler deeper into the web (cutting off no of URIs per pay-level domain)
Pay-Level-Domain Starvation

Fig. 8. Breakdown of PLDs per crawl hour.

[Hogan et al. 2011]
Swget [Fionda et al. 2011]

- Controlled exploration and retrieval of RDF sources
- Expanding paths (e.g., foaf:knows.foaf:name) specified via regular expressions
- User interface to tune parameters and view results
- More at http://swget.wordpress.com/ (and at the ISWC Poster Session!)

Fig. 1. The swget architecture and interaction scenario.
Conjunctive Queries

?x foaf:maker ?y .

- Built on triple patterns containing variables (?, ?, ?), (s, ?, ?), (?, p, ?),
  (?, ?, o), (s, p, ?), (?, p, o), (s, p, o)
- Variables are bound during query evaluation
- Query evaluation results in a set of variable bindings

<table>
<thead>
<tr>
<th>?x</th>
<th>?y</th>
<th>?z</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.w3.org/People/Berners-Lee/card#i">http://www.w3.org/People/Berners-Lee/card#i</a></td>
<td><a href="http://www.w3.org/DesignIssues/LinkedData.html">http://www.w3.org/DesignIssues/LinkedData.html</a></td>
<td>Linked Data - Design Issues - W3C</td>
</tr>
</tbody>
</table>
Join Tree

Traditional query evaluation: bind variables for triple patterns (based on indices), join results

- Often implemented via iterators with hasNext/getNext operations
Challenges for Link Traversal Evaluation

- Difficulty: interweave query processing with resource discovery (plan is known a-priori, sources are not)
- Dominant cost: network lookup and parsing
- But: QP has high complexity, may be slow on local storage, even slower over the web (orders of magnitude)
Iterator-based Implementation [Hartig et al. 2009]

- Traditional iterator-based query processor
- Feed URIs to lookup component
- If hasNext() returns false, feed IRIs into lookup component and defer evaluation of triple pattern
- Pro: paradigm fits current query processors (e.g., ARQ)
- Pro: (limited) multithreading via lookup component
- Con: potentially incomplete due to scheduling of lookups and variable bindings
Implementation with Symmetric Hash Joins [Ladwig et al. 2010]

- Operator is own thread
- Push triples into operator tree, bindings are stored in operator tree
- Joins carried out using Symmetric Hash Joins
- Pro: fully multithreaded
- Pro: avoids incompleteness due to scheduling lookups and deriving bindings
- Con: memory overhead
- Con: not all query patterns possible
Approximate Data Summaries [Harth et al. 2010]

- Combined description of schema-level and instance-level
- Use approximation to reduce index size (incurs false positives)
- Possible to use entire query for source selection
- Parallel lookups since sources can be determined for the entire query

- Can answer all triple patterns
- and combinations of triple patterns via join estimation
FUTURE WORK AND SUMMARY
Linked Data in the Smart Energy Grid [Wagner et al. 2010]

- Large, decentralised system
- Difficult to materialise data at central point
- Will have heavily coordinated parts, but should also be flexible
- Heavily regulated: privacy and security are of prime concern
- Resilience needed (no single point of failure)
Linked Data in the Internet of Things
Summary

- The Linked Data Web is a large, decentralised and complex system built on simple principles
  - identify resource via HTTP URIs
  - provide RDF that links to other URIs upon lookup
- Current trend around Linked Data allows for a re-think of components in Semantic Web Layer Cake
- Data publishers and consumers coordinate little
- Web of Data grows rapidly and covers a large variety of domains
- Algorithms operating over a common access protocol and data model
- First commercial applications emerging
References I


References II


Attribution

- Slides from my SWT-2 lectures, WWW 2010 SILD tutorial and INFORMATIK 2011 tutorial
- Images of Berlin, Hegel and Dietrich via Wikipedia