National Technical University of Athens
School of Electrical and Computer Engineering
Multimedia, Communications & Web Technologies

HEALLINK
Hellenic Academic Libraries Link

Transient and Persistent RDF Views over Relational Databases in the Context of Digital Repositories

Nikolaos Konstantinou, Dimitrios-Emmanuel Spanos, Nikolas Mitrou

European Union
European Social Fund

OPERATIONAL PROGRAMME
EDUCATION AND LIFELONG LEARNING
investing in knowledge society
MINISTRY OF EDUCATION & RELIGIOUS AFFAIRS
MANAGING AUTHORITY
Co-financed by Greece and the European Union

NSRF 2007-2013
programme for development
EUROPEAN SOCIAL FUND

Outline

- Introduction
- Evaluation
- Conclusions

(Linked) Open Data (1/2)

- A shift toward openness in numerous domains
  - Cultural heritage (europeana.eu)
  - Governance (data.gov.uk)
  - News (guardian.co.uk/data)
- Mature technological building blocks
  - W3C Recommendations
    - HTTP, XML, RDF, SPARQL, R2RML

(Linked) Open Data (2/2)

- Richer expressiveness
  - Describing and querying information
- Ease of synthesis (integration, fusion, mashups)
- Semantic enrichment
- Inference (implicit vs explicit facts)
- Reusability by third parties
- Content can be linked
  - And be part of broader contexts

The Problem: Data Mapping

- Data mapping and synchronization between databases and RDF
- R2RML (RDB to RDF Mapping Language)
  - A standardized way to express relational-to-RDF mappings
  - Relatively new standard
    - W3C recommendation as of Sept. 2012
  - Reusable mapping definitions
  - Supported by numerous tools
    - Db2triples, D2RQ, Ultrawrap, Virtuoso, R2RML Parser etc.

Methodological Approach (1/2)

- Dilemma: Transient or Persistent RDF views?
- Transient RDF Views
  - Offered on top of the data
  - The RDF graph is implied (not materialized)
  - Queries on the RDF graph are answered with data originating from the actual dataset
  - Similar to the concept of SQL views
  - Typically involve SPARQL-to-SQL query translation

Methodological Approach (2/2)

- Persistent RDF Views
  - The data is exported (dumped) asynchronously
  - Similar to the materialized view in databases
  - Need for manual synchronization
  - Queries on the RDF graph are answered on the dump, therefore
  - Results from the RDF graph may differ from actual dataset

Outline

- Introduction
- Evaluation
- Conclusions

Experiments Setup

- Linux Server
- 3 separate DSpace (dspace.org) installations
  - 1k, 10k, 100k items and users, respectively
  - Random-generated text-values in metadata field values and person names
- Open-source tools involved
  - Postgresql
  - D2RQ experimental
  - R2RML Parser
  - Virtuoso Universal Server

D2RQ Experimental

- Open-source, written in Java, available at d2rq.org/download
- Offers transient RDF views over relational databases, runs as a server
  - Supports D2RQ Mapping language and R2RML
- Allows dumping relational database contents as persistent RDF based on the mappings
- R2RML support is still experimental
  - http://sourceforge.net/mailarchive/message.php?msg_id=30185355

R2RML Parser (1/2)

- An open-source R2RML implementation
- A command-line tool
  - In Java, uses the Jena Semantic Web framework
  - Exports relational database contents into RDF graphs, based on an R2RML mapping document
- Supports MySQL and Postgresql
- Output can be written in RDF or relational database
- See https://github.com/nkons/r2rml-parser

R2RML Parser (2/2)

- Allows arbitrary SQL queries to be used as logical views, including SQL functions and foreign keys
- Limitations
  - No SQL query nesting, union, intersection or difference
  - No multiple graphs from a single execution
  - Covers not all but most of the R2RML constructs (See https://github.com/nkons/r2rml-parser/wiki)
- Does not support transient RDF Views, (i.e. no on-the-fly SPARQL-to-SQL translations)

Virtuoso Universal Server

- Mature, enterprise-level software
- Open-source and commercial version
- Extensible, includes Sponger RDF-iser, a reasoning engine, supports clustering, etc
- Can be used as a relational database and/or a triplestore
- Offers RDF Views using R2RML
  - Subject to several limitations

Simple R2RML Mapping Example


```
@prefix map: <#>.
@prefix rr: <http://www.w3.org/ns/r2rml#>.
@prefix dcterms: <http://purl.org/dc/terms/>.
map:persons-groups
  rr:logicalTable [ rr:tableName '"epersongroup2eperson"'; ];
  rr:subjectMap [
    rr:template 'http://data.example.org/repository/group/{"eperson_group_id"}';
    ];
  rr:predicateObjectMap [
    rr:predicate foaf:member;
    rr:objectMap [ rr:template
'http://data.example.org/repository/person/{"eperson_id"}';
    rr:termType rr:IRI; ] ].
```

Table epersongroup2eperson

	id [PK] integer	eperson_group_id integer	eperson_id integer
1	499501	1	1
2	499502	1	2
3	499503	1	3
4	499504	1	4
5	499505	1	5
6	499506	1	6

```
<http://data.example.org/repository/group/1> foaf:member
      <http://data.example.org/repository/person/1> ,
      <http://data.example.org/repository/person/2> ,
      <http://data.example.org/repository/person/3> ,
      <http://data.example.org/repository/person/4> ,
      <http://data.example.org/repository/person/5> ,
      <http://data.example.org/repository/person/6> .
```

Complex R2RML Mapping Example


```
map:dc-contributor
  rr:logicalTable <#dc-creator-view>;
  rr:subjectMap [
    rr:template
'http://data.example.org/repository/item/{"handle"}';
  ];
  rr:predicateObjectMap [
    rr:predicate dcterms:creator;
    rr:objectMap [ rr:column '"text_value"' ];
  ].
```

Arbitrary SQL query results

handle character varying(256)	text_value text
123456789/3	krrvwkqxfdtmctv vtczgnkolzc m
123456789/3	eixfkv bvvnqecsdlnygbwldrxaelcxpx fqydnh
123456789/4	itc kcoffmphjbqpcz squgsonbuzqbij
123456789/4	kfitk zi

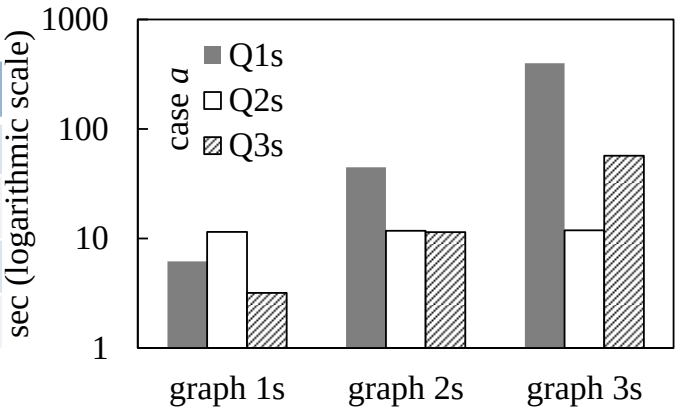
```
<http://data.example.org/repository/item/123456789/3>
  dcterms:creator
    "krrvwkqxfdtmctv vtczgnkolzc m" ,
    "eixfkv bvvnqecsdlnygbwldrxaelcxpx fqydnh" ;

<http://data.example.org/repository/item/123456789/4>
  dcterms:creator
    "itc kcoffmphjbqpcz squgsonbuzqbij" ,
    "kfitk zi" ;
```

```
<#dc-creator-view>
  rr:sqlQuery """
  SELECT h.handle AS handle,
         mv.text_value AS text_value
FROM handle AS h, item AS i,
  metadatavalue AS mv,
  metadataschemaregistry AS msr,
  metadatafieldregistry AS mfr
WHERE
   i.in_archive=TRUE AND
   h.resource_id=i.item_id AND
   h.resource_type_id=2 AND
   msr.metadata_schema_id=mfr.metadata_schema_id AND
   mfr.metadata_field_id=mv.metadata_field_id AND
   mv.text_value is not null AND
   i.item_id=mv.item_id AND
   msr.namespace=
   'http://dublincore.org/documents/dcmi-terms/' AND
   mfr.element='creator' AND
   mfr.qualifier IS NULL
   """.
```

Simple mapping results

- Case *a*: Transient views, using D2RQ, over PostgreSQL, and an R2RML mapping
- Case *b*: Persistent RDF views, using Virtuoso, over an RDF dump of the database
- Case c: Transient views, using Virtuoso, over its relational database backend, and an R2RML mapping

	Graph 1s			Graph 2s			Graph 3s		
Q1s	**6.18**	0.1	0.56	**44.75**	0.31	0.88	**398.74**	2.31	3.8
Q2s	**11.4** 8	0.07	2310	**11.76**	0.08	3522	**11.91**	0.12	4358
Q3s	**3.18**	0.04	0.22	**11.44**	0.04	0.68	**57.08**	0.04	1.28
	a	*b*	*c*	***a***	*b*	*c*	***a***	*b*	*c*



Complex mapping results

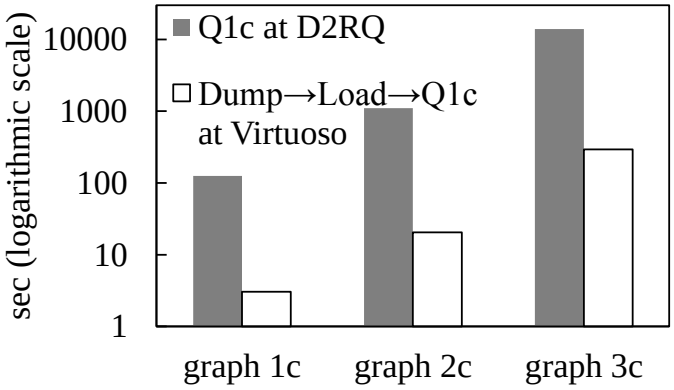
- Case 1: D2RQ (transient RDF view)
- Case 2: Export data into RDF using R2RML Parser, load it into Virtuoso (persistent RDF view), then execute SPARQL query

Export database into RDF

Graph	Triples	D2RQ	R2RML Parser
1c	16,482	3.15	**0.914**
2c	159,840	28.96	**7.732**
3c	1,592,790	290.92	**80.442**

Load into Virtuoso

Graph	Load into Virtuoso
1c	**1.87**
2c	**11.04**
3c	**201.03**



SPARQL query

	Graph 1c		Graph 2c		Graph 3c	
Q1c	**125.34**	**0.27**	**1100.58**	**1.77**	**13921.64**	**11.18**
Q2c	0.34	0.048	0.35	0.05	1.04	0.05
Q3c	144.01	0.13	1338.84	2.19	>6h	10.19
	D2RQ	Virtuoso	D2RQ	Virtuoso	D2RQ	Virtuoso

Outline

- Introduction
- Evaluation
- Conclusions

Conclusions (1/2)

- On-the-fly SPARQL-to-SQL conversions still are slow
  - There is much room for improvement in SPARQL-to-SQL translations
- Queries over RDF dumps perform significantly faster
  - Especially when SPARQL queries involve many triple patterns that are translated to many `JOIN` statements

Conclusions (2/2)

- Virtuoso transient RDF views perform well, but
  - Open-source version does not allow connection to external databases
  - No arbitrary SQL queries as logical tables
- In digital repositories:
  - Persistent RDF views (dumps) are preferable to transient (on-the-fly SPARQL-to-SQL translations)
  - Changes are not as frequent as to justify the burden caused by round-trips to the database
  - The trade-off in data freshness is remedied by the improvement in query answering

Open Research

- Reproducible results
- Datasets and software tools used for this work are online
- You can find here:
  - The software that was used
  - Database SQL dumps
  - The R2RML mapping files
  - The RDF graphs that were generated
  - The SPARQL queries that were used to evaluate the results

Thank you for your attention!

Questions?