QUERYING RDF DATA IN THE GRID

◄WS-DAI RDF(S) Querying Specification►

OpenGridForum OPEN FORUM | OPEN STANDARDS

Said Mirza Pahlevi and Isao Kojima, *Grid Technology Research Center, AIST, Tsukuba, Ibaraki 305-8568, Japan http://dbgrid.org {mirza|kojima}@ni.aist.go.jp*

The Grid Meets the Semantic Web: Motivation

The semantic Web provides a common framework that allows data to be shared and reused across applications, enterprises, and community boundaries. It is based on the Resource Description Framework (RDF), which is intended to provide a simple way to make *statements (triples) about resources*. RDF is based on the idea that the things being described (subjects) have properties (predicates), which have values (objects). Since objects can refer to the subjects of other resource statements, RDF creates *relationships between resources*.

Large amounts of RDF data have already been created and stored in various applications and projects worldwide. For example, the UniProt Protein Database [1] contains 262 million triples, while the IngentaConnect bibliographic metadata storage [2] contains over 200 million triples. With the rapid proliferation of semantic Web technologies, there is an increasing demand for access to the large amounts of distributed RDF data. We argue that a scalable, robust access mechanism should be based on grid computing technologies.

Recently, grids have applied semantic Web technologies to discover resources, manage policies, and describe provenance information, among other actions. Like the semantic Web applications, semantic grid applications usually handle a large amount of RDF data. An example is the CombeChem semantic Grid project [3], which stores more than 80 million triples in multiple databases. Therefore, providing a standard grid-based interface for accessing (the large amounts of) RDF data in the grid is important.

Data Resource Access Interfaces: the Standard

The Database Access and Integration Service Working Group (DAIS-WG) of the Open Grid Forum (OGF) has started to define a collection of querying interfaces for RDF data resources‡ [4]. Similar to the standard interfaces for relational data resources (WS-DAIR specification) and for XML data resources (WS-DAIX specification), the querying interfaces extend the basic interfaces defined in the core specification (WS-DAI). An RDF data resource is a data source/sink that is based on the RDF model, together with any associated management infrastructure. An example is RDF data managed by an RDF repository system (a.k.a. triplestore), such as Sesame [5] and Boca [6]. As shown in the table, the querying specification provides two types of interface: those for *data resource management* and those for *data resource access*. The latter use the *de facto* standard RDF query language SPARQL [7] for data retrieval and provide direct and indirect access to the data resources.

> **RDF DATA RESOURCE MANAGEMENT INTERFACE**
>
> **RDFSCollectionAccess** (*AddGraphs*, *GetGraphs*, & *RemoveGraphs*)
> → create, retrieve & remove RDF data in an RDF data resource.
>
> **RDF DATA RESOURCE ACCESS INTERFACES**
>
> **SPARQLAccess** (*SPARQLExecute*)
> → direct access to an RDF data resource using a SPARQL query.
>
> **SPARQLFactory** (*SPARQLExecuteFactory*)
> → indirect access to an RDF data resource using a SPARQL query.
>
> **SPARQLResultsSetAccess & SPARQLTriplesSetAccess** (*GetResults*)
> → access SPARQL query results returned by SPARQLExecuteFactory.

Relation to W3C's SPARQL Specification

The SPARQL specification of W3C consists of a query language, a means of conveying a query to a SPARQL processor, and the XML format in which the query results will be returned. Nevertheless, the SPARQL specification does not dictate indirect or direct access using a (stateful) Web service and data resource management, both of which grid applications generally require. The RDF querying specification does not reinvent the wheel for SPARQL-related components. While extending the WS-DAI core specification to fulfill the grid general requirements, in addition to using SPARQL query language for RDF data retrieval, it adopts parts of the SPARQL specification, such as message and XML formats.

REFERENCES

[1] http://dev.isb-sib.ch/projects/uniprot-rdf
[2] http://www.ingentaconnect.com
[3] http://www.combechem.org/
[4] https://forge.gridforum.org/sf/go/doc14074?nav=2
[5] http://www.openrdf.org/
[6] http://ibm-slrp.sourceforge.net
[7] http://www.w3.org/TR/rdf-sparql-query

‡ This activity also defines interfaces for accessing data resources using ontologically based primitives.

QUERYING RDF DATA: USE CASES

◄WS-DAI RDF(S) Querying Specification►

Said Mirza Pahlevi and Isao Kojima, *Grid Technology Research Center, AIST, Tsukuba, Ibaraki 305-8568, Japan http://dbgrid.org {mirza|kojima}@ni.aist.go.jp*

Semantic Matchmaking of Grid Resources

Motivation. The grid includes a large number of resources with various capabilities distributed across different organizations. To facilitate effective grid resource discovery and selection, semantic grid applications and architecture such as S-MDS [1], MyGrid [2], and S-OGSA [3] describes grid resources using a metadata framework such as RDF/OWL.

Goal. Given repositories and services that store the semantic descriptions created by various semantic grid applications, the goal is to query the semantic descriptions using a high-level RDF query language, SPARQL, and order the matched resources based on a specific ordering criteria such as class subsumption relationship.

Use Case. Figure 1 shows a SPARQL-based semantic matchmaking system designed for the querying specification. RDF data access services (RDAS) provide a standard access method for the RDF data resources. A requester sends a SPARQL query as a resource request to a matchmaker, and the matchmaker forwards the query to one or more repositories. On receiving query results, the matchmaker orders the results according to class subsumption relationships. Similar work has been proposed and implemented in a semantic web environment [4].

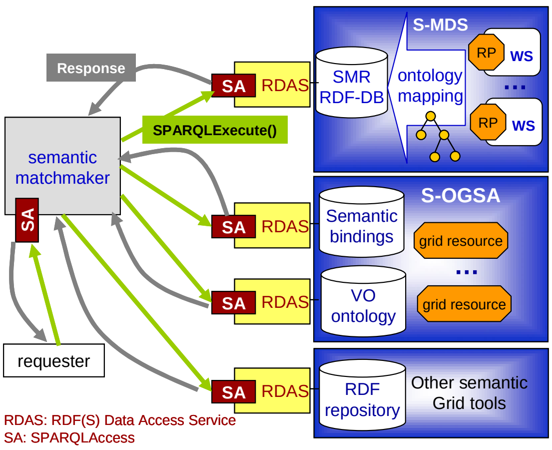

Figure 1. Semantic Matchmaking using SPARQL

Distributed RDF Storage for Ubiquitous Objects

Motivation. Ubiquitous code (ucode) [5] is a unique code in the form of 128-bit binary data assigned to real-world objects for identification purposes. It is stored as a ucode tag attached to the identified object; this is often physically implemented as an RFID tag. A relation between objects (ucodes), which is called a *ucode relation* (*UCR*), is modeled as a *triple*, consisting of *subject*, *predicate*, and *object*. For example, this apple (subject ucode) is produced by (predicate relation ucode) the JA Tsugaru-Minami Farm (object ucode). The triples, which are usually large in number, are stored in a wide area distributed database (UCR database). There have been efforts to implement UCR databases using SPARQL-supported RDF databases [6].

Goal. Given such RDF databases, the goal is to provide a robust and scalable federated database that supports seamless access over the heterogeneous RDF databases.

Use Case. Figure 2 shows an overview of a grid-based distributed RDF database, which federates various (UCR) RDF databases. The service-based SPARQL query interfaces provide a uniform access to the heterogeneous RDF databases for distributed query processing. The indirect data access of the proposed specification (*SPARQLAccess*) can be used in the distributed query processing, as was done in [7]. Another attempt at distributed SPARQL query processing is described in [8].

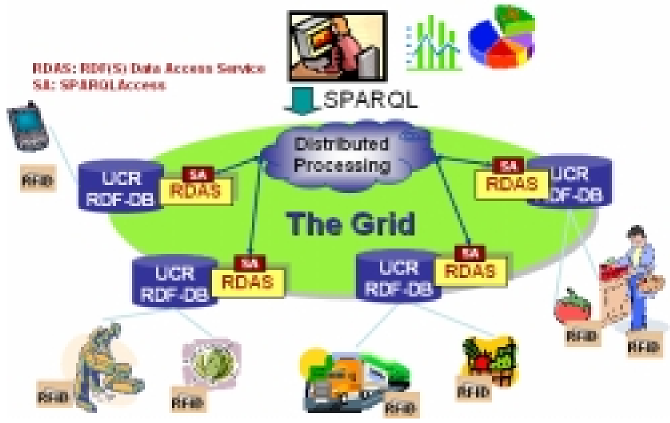

Figure 2. Large scale distributed RDF Database

REFERENCES

[1] M.P. Said et.al., "S-MDS: A Semantic Information Service for Advanced Resource Discovery and Monitoring in WS-Resource Framework", GGF16, 3rd Semantic Grid Workshop, 2006.

[2] MyGrid Project: http://www.mygrid.org.uk

[3] O.Corcho, et.al, "An Overview of S-OGSA: a Reference Semantic Grid Architecture", J. of Web Semantics, Vol. 4, pp. 102-115, 2006.

[4] M.P. Said, et.al, "SPARQL-based OWL-S Service Matchmaking", SMR 06 in conjunction with VLDB'06, pp. 35-50, 2006.

[5] T-Engine Forum, "Ubiquitous ID Architecture", UID-CO00002-0.00.24, Nov. 2006 (in Japanese).

[6] Nihon Unisys SSDB: http://dev.tyzoh.jp/trac/semi-structured-db/

[7] OGSA-DAI-RDF: http://www.dbgrid.org/OGSA-DAI-RDF/

[8] Federated SPARQL, http://www.w3.org/2007/05/SPARQLfed/