OPEN ACCESS
知 识 管 理 论 坛
新出网证（京）字 058 号
ISSN 2095-5472
www.kmf.ac.cn

近年来 SPARQL 查询技术的研究热点及进展

汤怡洁* 杨锐 刘毅

中国科学院国家科学图书馆武汉分馆/中国科学院武汉文献情报中心 武汉 430071
*通讯作者 E-mail: tangyj@mail.whlib.ac.cn

收稿日期：2013.09.14 录用日期：2013.11.25 发表日期：2014.01.06
本文网址：http://www.kmf.ac.cn/tabid/583/InfoID/2722/frtid/911/Default.aspx

摘要：
首先针对 SPARQL 查询语言的语法和查询过程等理论研究进行阐述，接着分析当前 SPARQL 查询技术的实现和扩展。最后通过对 Science Direct、IEEE 等数据库、语义网会议和相关大型项目的研究，总结出当前 SPARQL 的研究热点问题以及相关的研究进展，如关联数据的 SPARQL 查询、移动终端的 SPARQL 应用等。

关键词： SPARQL 语言 语义查询 关联数据 语义 Web 服务

基金项目： 本文系国家科技支撑计划课题“科技知识组织体系共享服务平台建设”子课题“科技知识组织体系（STKOS）的开放查询和推理接口建设”（项目编号：2011BAH10B03-5）研究成果之一。

1 引言

随着网络的发展，数字化资源呈现爆炸式增长，如何提高信息检索的质量，推出令人满意的检索技术成为全球范围内的研究重点和热点。由于网络资源具有分布式、异构性、易变性、更新快等特点，传统的基于字符串匹配的关键词检索技术在查全率和查准率方面难以满足用户的真实需求。因此，语义网概念和相关语义技术一经提出，得到了广泛关注——语义技术提供了针对信息组织、知识表示、机器理解等问题的解决方案。SPARQL 作为 W3C 推荐的语义检索的标准，可以使得检索操作以机器可理解的方式在语义层面上进行，实现语义检索进而提高检索的查全率和查准率。

2 相关理论研究

SPARQL（Simple Protocol and RDF Query Language）是一种用于 RDF 的查询语言，是基于之前的 RDF 查询语言（rdfDB、RDQL 和 SeRQL）发展而来的，用于访问任何可以映射到 RDF 模型的数据资源（包括本地的和远程的）。2008 年 1 月，SPARQL 正式成为一项 W3C 推荐标准，由三个独立的规范（查询语言规范、SPARQL 数据访问协议、XML 格式的查询结果）构成。2012 年 11 月，在 SPARQL 1.0 的基础上 SPARQL 工作组提出了全功能标准体系 SPARQL 1.1，并于 2013 年 1 月最终发布了一系列 SPARQL1.1 标准[1]。

2.1 SPARQL 语法

SPARQL 提供了 4 种不同形式的查询，SELECT、ASK、DESCRIBE 和 CONSTRUCT。其中，SELECT 查询形式

用于标准查询，以标准的 SPARQL XML 结果格式返回查询结果。ASK 查询返回结果是 yes 或 no，没有具体内容。DESCRIBE 用于提取本体和实例数据的一部分，返回一个图形，其中包含和图形模式匹配的节点的相关信息。CONSTRUCT 用来为每个查询结果输出一个图形模式，这样就可以直接从查询结果创建新的 RDF 图。

在 SPARQL 各种查询类型中，查询语句的构建必须遵循 SPARQL 基本查询语法，SPARQL 的语法和传统 SQL 的语法有相似之处，具体语法如表 1 所示：

表 1 SPARQL 基本语法

序号	语法	含义
1	BASE	根 IRI，其他以此为根的 IRI 可以写成相对形式
2	PREFIX	IRI 前缀的缩写
3	SELECE、ASK、DESCRIBE、CONSTRUCT	查询关键字
4	?person ?name ?age	要查询的变量，使用?标识变量
5	FROM	从何处查询，可以一次查询多个 RDF 数据集
6	WHERE	过滤条件集合，等同于 SQL 的 WHERE 子句
7	?person foaf:name ?name	具体的过滤条件，使用 Turtle 语法
8	OPTIONAL	可选过滤条件
9	FILTER(REGEX(?name, “Jack”))	明确化的过滤条件，类似 SQL 中的 LIKE 等
10	ORDER BY	可指定排序，与 SQL 中类似
11	LIMIT 10	限定返回结果记录数，类似 SQL 中的 TOP 10
12	OFFSET 10	翻页功能，掠过前 10 条，从 11 条开始返回结果

2.2 SPARQL 查询过程

SPARQL 查询过程[2]中，用户（包括人和机器）通过一系列接口与系统进行交互，接口将查询请求送入 SPARQL 查询处理器，调用底层的 RDF 存储获取相关的结果记录。SPARQL 通过查询器扫描关键词，并且根据标准解析查询序列验证 RDF 三元组的有效性。如果查询不正确，则在处理的过程中及时通过接口为用户返回错误信息，如图 1 所示：

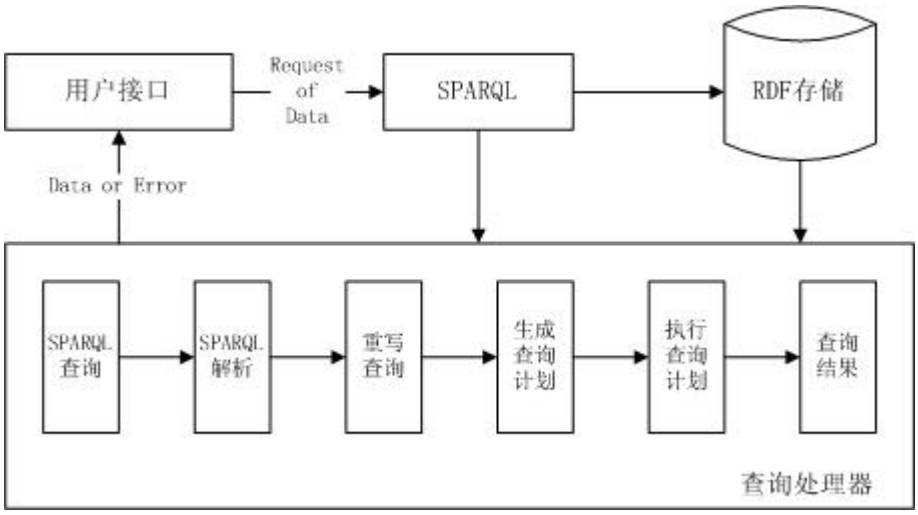

图 1 SPARQL 查询过程

在 SPARQL 处理器中，首先利用解析器对查询语句进行解析，判断是否存在语法错误。接着在重写查询阶段，以规则为基础重新优化查询语句。最后通过执行 QEP（查询执行计划）发生器产生的计划获取 RDF 数据并通过接口返回给用户。

3 SPARQL 查询技术的实现和扩展

3.1 基于 RESTful 协议的 SPARQL 终端

SPARQL Endpoint 是遵循 SPARQL 协议的查询终端，用户可以通过 SPARQL 查询语言对知识库进行查询操作[3]。SPARQL 终端提供 Web 交互界面，支持用户直接输入查询语句进行查询，同时也支持机器用户通过 HTTP URI+SPARQL 语句的 Restful 方式调用。SPARQL Endpoint 支持 GET/POST 请求方式；支持各种查询类型，如 CONSTRUCT（构建）、DESCRIBE（描述）；查询结果以 XML/JSON/N3 等多种方式输出。

多个数据集之间的联合查询可以利用数据集提供的远程 SPARQL 终端进行，SPARQL 工作组提出的最新 SPARQL 1.1 标准体系专门针对 SPARQL 联合查询以及扩展进行说明，利用 SERVICE 关键词指向 SPARQL 联合查询处理器，允许调用远程 SPARQL 终端，并通过 OPTIONAL 在多个数据集中关联查询，挖掘内部关联关系。

目前已经出现大量 SPARQL 查询工具，如 Jena ARQ Processor、Twinkle、ViziQuer 等支持用户操作[4]。其中 Jena ARQ Processor 以命令行的形式提供用户使用，对用户要求较高，使用较为复杂。Twinkle 较之 Jena ARQ Processor 来说，提供较好的图形化界面供用户使用，可以加载、编辑和保存 SPARQL 查询语句，支持查询 RDF 文件和关系型数据库等。ViziQuer[5]能够帮助用户连接远程 SPARQL 终端，浏览检索包含 rdf:type 关系的数据。通过工具连接并抽取 SPARQL 终端的基础数据架构，支持用户针对抽取的基础数据架构进行可视化浏览，在此基础上用户可以通过 ViziQuer 构建 SPARQL 查询，获取远程数据集中的相关数据。

3.2 SPARQL 与 SQL 查询语言转换

SPARQL 查询是针对 RDF 数据的语义查询技术，检索以机器可理解的方式在语义层次上进行。但是目前大多数的应用系统仍沿用传统关系型数据库存储，采用元数据字段组织数据的方式。关系型数据库查询语言 SQL 语言和采用视图匹配方式进行本体查询的 SPARQL 语言有较大的差异，因此，如何有效地将用户语义检索的请求转换为对传统关系型数据库的查询是语义查询技术的核心问题之一。

针对 SPARQL 访问现有关系型数据库，SPARQL-SQL 查询语言转换的解决思路是在现有关系型数据库的数据结构基础上，通过映射规则生成相应的 RDF 本体知识组织结构支持 SPARQL 的访问查询，代表应用有 D2RQ 等[6]。这种方法从本质上讲并没有改变底层数据存储模式，只是在上层进行了数据组织形式转换。

在语义存储的三种模式中基于传统数据的存储模式目前应用较为广泛，如 Jena、Sesame 都是利用关系型数据库作为底层存储的。基于这种思路的研究[7]都有一个共同的设计模式，利用各种不同的映射算法匹配传统数据库存储形式和 RDF 数据格式，如结构映射算法、数据映射算法、查询映射算法等。

4. SPARQL 技术热点应用

4.1 Linked Data 查询处理

Linked Data（关联数据）简单地说，就是按标准的 RDF 格式定义组织数据，用三元组（主体、谓词、客体）形式来表示资源。其本身并不具备语义特征，但是它可以在数据层面建立关联，支持语义网的各种语义操作。关联数据的服务功能需要检索和解析 RDF 数据，通常采用 SPARQL 作为标准的 RDF 解析语言，同时提供 SPARQL Endpoint 服务是关联数据应用的重要方式之一。图 2 以 LOD 云图为基础，具体描述了关联数据集和 SPARQL Endpoint 之间的关系[8]。DBpedia 提供了自身的 SPARQL 终端，支持外部服务通过 SPARQL 查询相关数据；Virtuoso LOD SPARQL 终端提供了多个关联数据集的检索服务，利用分布式子查询技术在各个数据集中获取检索结果。图中还描绘了 Google 搜索引擎在未来将利用 SPARQL 终端涵盖所有 LOD 云中的数据集。

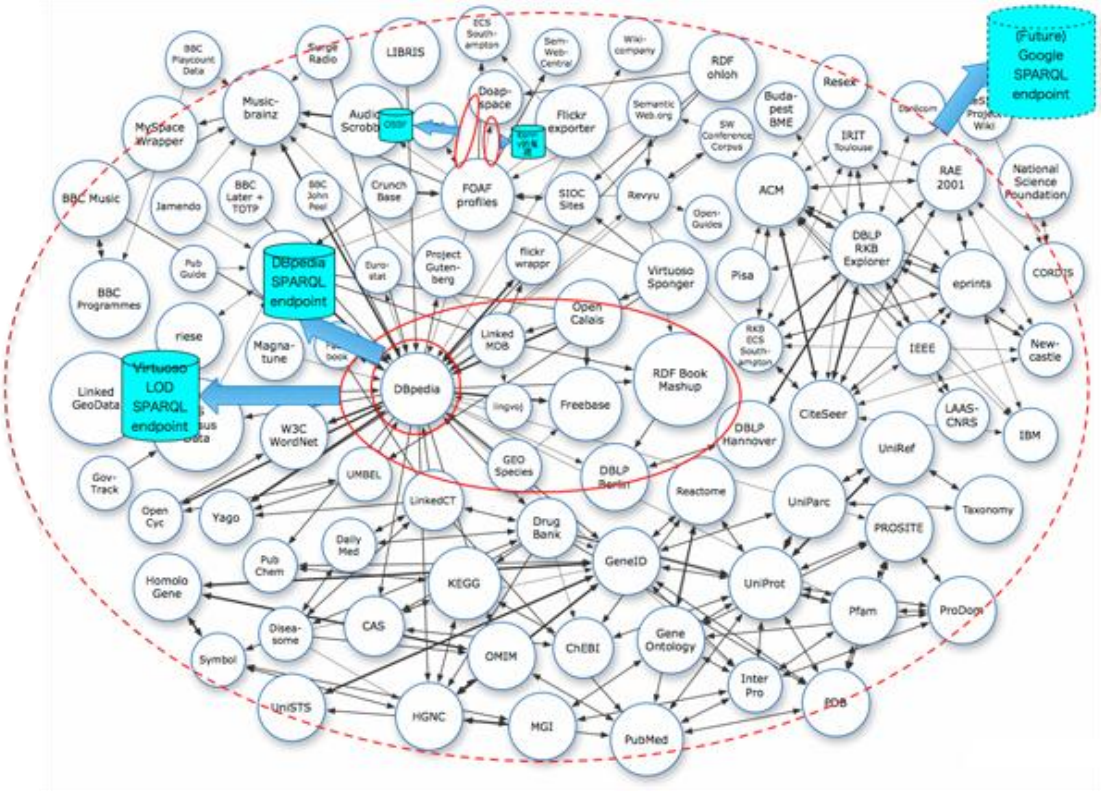

图 2 LOD 云图与 SPARQL Endpoint

目前针对 Linked Data 的 SPARQL 研究主要集中在多数据集查询方式、分布式 SPARQL 查询算法优化等方面。在 2012 年 7 月举办的 ICWE 2012（12th International Conference on Web Engineering）会议上，来自德国柏林洪堡大学（Humboldt-Universität zu Berlin）的 O. Harting 提出了针对多个关联数据集之间的 SPARQL 查询处理方法，主要包括数据仓库、联合查询和关联数据查询处理[9]。

数据仓库是针对统一的中心数据库内数据的查询处理方法，可以通过 SPARQL Endpoint 访问集成多个数据源的公共数据集（如 Sindice）。这种方法具有较高的查询执行效率，但是查询的数据无法保证实时更新，并且无法涵盖所有的数据信息。联合查询方法是基于关联数据出版商提供的查询服务进行分布式查询处理，分发器分析查询请求并将之分解为若干个子查询请求，分别发送到不同的查询服务中执行并返回结果。这种方法能保证查询数据的实时更新，但是所有数据集都必须提供标准的 SPARQL Endpoint 接口。关联数据查询处理是依赖于关联数据原则以执行查询中的数据链接遍历为处理方式的关联数据查询方法。这种方法可以保证查询数据的实时更新，而且不需要标准的 SPARQL Endpoint，直接支持关联数据的表示形式。

在联合查询和关联数据查询处理两种方法中，分布式 SPARQL 查询和关联数据的遍历算法是提高整体查询效率的关键，目前分布式查询算法大多数都是基于最小生成树算法（MST-based algorithm）的，比较常用的有 Boruvka’s algorithm、Prim’s algorithm、Edmonds’ algorithm、Kruskal’s algorithm 等。在算法优化方面，欧盟 FP7 的 SHARE 项目中提出了两种优化方法：一种是利用最小生成树标准图算法在查询执行之前计算出一个静态的查询计划；另一种是在查询执行的过程中，利用统计预测等相关算法进行动态分析并及时制定符合要求的执行计划。

4.2 语义 Web 服务发现

在 Web 服务研究中，如何使机器自动、准确、高效地进行服务发现、匹配、组合、监控和调用一直都是热点和难点问题。语义 Web 服务提供了一种新的解决思路。语义 Web 服务是以一种明确的、计算机能够理解的语言来描述 Web 服务功能和内容，同时增强 Web 服务的操作性能和健壮性。Web 服务利用语义网丰富的语义描述能力和强大的逻辑推理能力来表述其含义，通过这些带有语义信息的描述来实现服务的自动发

现、匹配、组合、监控和调用。

基于描述逻辑推理的语义 Web 服务发现方法在 Web 服务数量大幅增加和本体复杂度不断提升的情况下，会带来语义 Web 服务发现效率和可扩展性方面的问题。如何提高语义 Web 服务发现的性能是研究的重点之一。通过调研相关项目及其研究成果发现，在语义 Web 服务发现过程中加入 SPARQL 语言处理是解决上述问题的途径之一[10]。有学者指出在语义 Web 服务匹配算法之前先通过基于 SPARQL 查询的预处理策略过滤 Web 服务仓库并消除服务描述。这种方法降低了发现机制的搜索范围，从而提升了整体性能。这种解决方案不独立提供新型的发现机制，而是较好地适用于目前现有的各种发现机制中。

在现有的服务发现方法中，针对服务请求和服务广告都采用相同的发现机制。但是服务广告是服务的具体描述，信息丰富完整；服务请求是针对服务的某些特性，并非构造一个完整的服务描述。研究人员认为将服务请求和服务广告分开处理能够有效地提高服务发现效率——使用语义 Web 查询语言 SPARQL-DL 作为服务请求的描述语言来获取已发布的服务，采用 OWL-S 描述服务广告支持有效的服务发现。

在组合 Web 服务方面，WS-BPEL（Web 服务业务流程执行语言）是最为成熟和被广泛支持的技术，是一种可执行的 XML 语言，被描述的业务流程中每个单元都是由 Web 服务实现的。在服务组合中，Web 服务的动态选择必须能够适应服务的各种变化，然而在 WS-BPEL 中大多数语言不支持动态选择。为了解决这一问题，研究人员提出了采用 SPARQL 和 WS-BPEL 开发支持 Web 服务的动态选择模型[11]。在这个新型模型中，用户可以在运行环境中通过查询随时提出服务需求，以保证在不断变化的环境中，系统自动选择适应需求，而无需重构整个服务组合。

4.3 移动终端的 SPARQL 应用

随着传感网络和移动通讯的不断发展，越来越多的智能终端承担着数据处理的功能，在智能终端的数据计算增强了用户间的通讯与合作。具体而言，智能终端依赖于一个小型计算平台，采用语义技术进行处理、利用、揭示相关知识。目前各类语义工具，如三元组存储、推理、查询等基本上都是面向大型应用程序设计的，在高性能服务器上进行计算，因此建立智能终端的小型计算平台在硬件性能上将会是一个巨大的考验。

针对移动智能终端的语义技术和知识的处理方式基本上是在智能终端自身进行中小数量级的知识计算，辅以大数量级知识计算的远程接口调用[12]。利用支持中小数据量语义应用的基准构建轻量级系统架构，在移动设备上部署三元组存储，提供基于移动设备的可共享本地数据仓库。在共享数据仓库的基础上构建语义应用，并通过可对外提供访问服务的 SPARQL Endpoint 支持网络联合查询。同时语义处理可以应用于远程服务器上，终端设备只包含语义数据接口，通过接口从服务器远程获取语义数据。在这种情况下，必须实行相关的机制以确保在不同环境的终端设备中数据的有效性、安全性和保密性。

目前已经开发出能与三元组存储系统交互的基于事件的组件和移动终端 API，如 Qsparql、Soprano、sparqlpush 等[13]，现有的移动应用开发框架（QT、Python）通过 API 调用数据商或应用软件提供的 SPARQL 终端查询数据资源。其中 sparqlpush 可以通过对用户提交的 SPARQL 查询请求的登记，在 PuSH 服务器端记录该请求。当数据集发生变化时，sparqlpush 自动进行查询请求操作，记录下所有有结果变化的 SPARQL 查询请求，在 PuSH 服务器端通过 PubSubHubbub 协议向相关移动设备发推送通知，类似于 RSS 服务。

5. 结论

通过对 Science Direct、ACM、IEEE 等多个数据库中的文献、ISWC、ESWC 等语义网的专业会议及国际上大型项目的调研分析发现，在语义查询方面针对 SPARQL 查询语言的研究和应用的关注重点不再是理论探讨和基本应用，而是基于当前语义网各种技术和应用的发展如何进一步拓展 SPARQL 技术发展。正如本文中描述的如何在海量 RDF 数据和关联数据集中组合 SPARQL 关联查询以及在移动应用中如何集成 SPARQL 查询服务等都成为了当前 SPARQL 查询技术的研究热点。

参考文献：

1. Last SPARQL 1.1 proposed recommendations published [EB/OL]. [2013-03-19]. http://www.w3.org/blog/SW/2013/01/29/missing-sparql-1-1-proposed-recommendations-published/.
2. Search RDF data with SPARQL[EB/OL]. [2013-03-23].http://www.ibm.com/developerworks/xml/library/j-sparql/.
3. SparqlEndpoints [EB/OL]. [2013-03-25].http://www.w3.org/wiki/SparqlEndpoints.
4. Gupta R, Malik S K. SPARQL Semantics and execution analysis in Semantic Web using various tools[C]//2011 International Conference on Communication Systems and Network Technologies. Jammu：IEEE，2011：278-282
5. Zviedris M, Barzdins G. ViziQuer: A tool to explore and query SPARQL endpoints[C]//European Semantic Web Conference 2011, LNCS 6644. Berlin：Springer-Verlag ,2011: 441-445.
6. Kashlev A, Chebotko A. SPARQL-to-SQL query translation: Bottom-Up or Top-Down?[C]//2011 IEEE International Conference on Services Computing. Washington DC：IEEE，2011 ：757-758
7. Chebotko A, Lu Shiyong . Querying the Semantic Web: An efficient approach using relational databases [M]. Saarbrücken: LAP Lambert Academic Publishing, 2009.
8. 吕康豪. RDF Web 与 SPARQL 的应用 [EB/OL]. [2013-04-02].http://semwebtw.openfoundry.org/2010/Talks/0319-semweb-kennyluck/.
9. Hartig O. An introduction to SPARQL and queries over linked data [C]// 11th International Semantic Web Conference(ISWC 2012), LNCS 7387. Berlin：Springer-Verlag , 2012: 506－507.
10. Garcia J M, Ruiz D, Ruiz-Cortes A. Improving semantic web services discovery using SPARQL-based repository filtering [J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2012,17(4):12－24.
11. Tizzo N P, Coello J M A, Cardozo E. Improving dynamic Web service selection in WS-BPEL using SPARQL [C]// Systems, Man, and Cybernetics (SMC), 2011 IEEE International Conference on 2011. Anchorage：IEEE，2011 :864-871
12. Aquin M, Nikolov A, Motta E. Building SPARQL-Enabled applications with android devices[EB/OL].[2013-04-02].http://iswc2011.semanticweb.org/fileadmin/iswc/Papers/PostersDemos/iswc11pd_submission_89.pdf.
13. Smart－M3 Storage Solutions [R/OL]. [2013-04-22]. http://www.diem.fi/files/deliverables/D5.6.3_Storage-solutions.

（本文责任编辑：刘远颖）