语义仓储构建技术研究进展 1)

邹益民 [1,2] 张智雄 [1] 钱力 [1,2] 王颖 [1]

[1]（中国科学院国家科学图书馆，北京 100190）

[2](中国科学院研究生院，北京 100190)

摘要 如何对海量的 RDF 数据进行存储、查询、存取和推理是 RDF 数据管理研究领域最关心的问题之一，文章界定了语义仓储的概念及其与关系数据库管理系统的区别，根据语义存储介质和组织方式的不同对语义仓储进行了分类，结合实际的案例对基于内存、基于传统数据库和原生方式存储模式的语义仓储的优缺点、适用范围和不同存储模式之间的区别和联系进行了分析，在语义仓储的分布式存储策略上，对集中式语义仓储和自组织语义仓储这两种网络结构的组织形式和应用系统做了综述，还对语义仓储测试基准及应用系统的研究进展进行了分析，讨论存在的问题及未来可能的研究方向。

关键字 RDF 存储；语义仓储；存储模式；分布式语义仓储；测试基准

Review on Techniques of Semantic Repository

Yimin Zou[1,2] Zhixiong Zhang[1] Li Qian[1,2] and Ying Wang[1]

[1]（*National Science Library, Chinese Academy of Sciences, Beijing 100190*）

[2]*(Graduate University of Chinese Academy of Sciences, Beijing 100190)*

Abstract Data storage, query, access and reasoning on vast amounts of RDF is one of the greatest concern of the RDF data management research areas. Comparing with RDMS, this paper gives the explicit concept of Semantic Repository, and divides the Semantic Repository storage model into memory model, traditional database model and native model based on the storage media and organization, and analyzes these model with application systems separately. Distributed storage strategy of the Semantic Repository, Clustered Semantic Repository and Self-Organized Semantic Repository are reviewed. It also descrips benchmarks and application systems, and points the problems and trends of this domain.

Key words RDF Store; Semantic Repository; Storage Mode; Distributed Semantic Repository; Benchmark

1 引言

随着语义网的迅速发展，大量的 RDF（Resource Description Framework）数据被生成、发布、检索和重用，在 2011 年 9 月关联开放数据（Linking Open Data，LOD）项目发布的关联开放数据云图（LOD Cloud Diagram)中包含了 295 个数据集和超过 310 亿个三元组[1]，而在 OpenLink 公司的关联数据云缓存（LOD Cloud Cache）中截至到 2012 年 3 月也包含超过 510 亿个三元组[2]。如何对这些海量的 RDF 数据进行管理，提供高效的数据存储、查询、存取甚至推理机制是目前 RDF 数据管理研究领域最关心的问题之一。

在目前的研究中，针对 RDF 数据管理系统出现了不同的名称，例如：三元组存储（Triplestore）、RDF 存储（RDF Store）、语义仓储（Semantic Repository）等，维基百科定义三元组存储为“一种为存储和检索

收稿日期：2012 年 5 月 4 日

作者简介：**邹益民**，1983 年生，博士生，主要研究方向：智能信息处理、知识组织等。E-mail：zou_yimin@163.com。**张智雄**，1971 年生，研究馆员，博士生导师，博士，主要研究方向：知识抽取，智能信息处理，知识组织等。**钱力**，1981 年生，博士生，馆员，主要研究方向：信息可视化、知识组织等。**王颖**，1982 年生，博士，馆员，主要研究方向：本体映射，知识组织等。

1) 本文系国家十二五科技支撑计划项目“面向外文科技文献信息的知识组织体系建设与应用示范”（项目编号：2011BAH10B00）课题三“科技知识组织体系共享服务平台建设” 以及中国科学院国家科学图书馆青年人才项目（项目编号：青 1105）的研究成果之一。

三元组而特别设计的数据库”[3]，欧洲数字图书馆的 B. Haslhofer 等人[4]则认为用 RDF 存储更能够全面地描述那些有能力处理 RDF 数据的系统，不但包括三元组存储，还包括四元组存储（Quadstore）等，定义 RDF 存储为“允许摄入和检索序列化 RDF 数据的系统”。而对于语义仓储目前还没有正式的定义，Ontotext 实验室的 OWLIM 工作组认为“语义仓储是一种数据库管理系统，能够被用来存储、查询和管理遵循 RDF(S)标准的结构化数据。”[5]，并且认为语义仓储与关系数据库管理系统相比最大的不同在于：1）用本体作为语义模式，并允许在这些数据上进行推理；2）采用了更加灵活和通用的数据模式（例如：图），这使得其能够快速地解释和适应新的本体和元数据模式[5]。笔者认为 Ontotext 对语义仓储的理解更加准确和全面，本文采用语义仓储来表示 RDF 数据管理系统。通常语义仓储的功能包括：数据存储、查询处理以及推理等，本文将主要围绕数据存储的相关技术进行总结和分析，在存储模式、分布式存储策略、测试基准以及应用系统等方面进行综述，以期对关注 RDF 数据管理的研究人员提供帮助。

2 存储模式

存储模式指逻辑上和物理上数据在存储设备上的组织方式，是数据存储面临的基本问题，根据存储介质的不同，笔者对语义仓储提出了如图 1 的分类体系，该体系中不包括那些不能提供语义查询的存储模式，例如：基于 Web 存储模式的语义 Web 搜索引擎 Swoogle [6]。

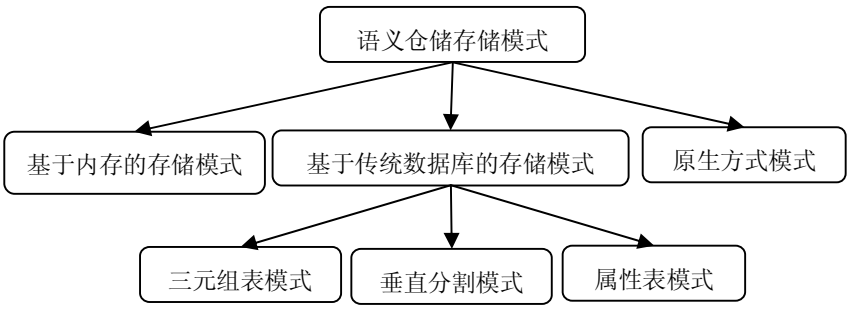

图 1 语义仓储存储模式分类体系

2.1 基于内存的存储模式

这类语义仓储的特点是将整个 RDF 图全部导入内存，按照某种数据结构对 RDF 数据进行组织，在内存结构上完成数据的存储、查询和推理操作。由于是在内存中进行数据管理，不存在磁盘更新的问题，因此该模式在数据加载、查询和推理方面具有很高的效率，但只能处理有限规模的数据。Ontotext 实验室的 SwiftOWLIM 是该模式的典型代表，它是目前最快的语义仓储系统，在普通电脑上使用非平凡推理加载数据的速度超过每秒 5 万个三元组，但最多只能处理不超过 1 亿个三元组[5]，此外基于该模式的语义仓储还有 Sesame-Memory[7]、Jena-Memory[8]以及 OWLJessKB[9]等。

2.2 基于传统数据库的存储模式

此类语义仓储充分利用传统数据库管理系统对 RDF 图进行组织、操作和管理。按照使用传统数据库的方式不同，可以将该模式分为：1）基于第三方数据库的语义仓储，例如：基于关系数据库 MySQL 的 3Store[10]、基于对象关系数据库 PostgreSQL 的 Sesame[7]以及基于 Key-Value 数据库的 CumulusRDF [11]等，基于该模式的语义仓储在功能上多是对现有数据库技术进行封装的智能中间件；2）在传统数据库上增加对 RDF 数据

的支持，例如：对象关系数据库 Virtuoso[12]和关系数据库 Oracle 都增加了对 RDF 数据的支持[13]。基于传统数据库的存储模式可以充分利用传统数据库的事务处理、查询优化、访问控制、日志和数据恢复等功能，技术成熟度高、系统稳定，并可实现语义数据和其他数据的无缝连接，但传统的数据库模式与 RDF 图的数据模式不一致，模式之间转换导致的存储和查询开销较大，即存在“阻抗失配”效应[14]。由于目前主流的基于传统数据库的模式是将（对象）关系数据库作为底层存储，因此本节将重点论述该存储方式，并根据逻辑组织的不同，将其分为：三元组表模式（Triple Table）、垂直分割模式（Vertical Partitioning）以及属性表模式（Property Table）。

（1）三元组表模式

三元组表模式是将 RDF 数据映射到（对象）关系数据库上最直接的方式，也是目前基于传统数据库的语义仓储中使用最多的一种模式，其设计思想简单，利用一张拥有三列的关系表对所有的三元组进行存储，表中的每列分别对应三元组的主语、谓语和宾语，如表 1 所示。这种方案简洁明了，易于在关系数据库上实现，且独立于具体的本体模式，但存在查询效率低的问题，这是因为查询时一个复杂的 SPARQL 查询将会被转换成一个存在大量自连接的 SQL 查询，如果三元组的数量巨大，随着自连接次数的增加，查询时间也会随之显著增长。为了改善查询效率，基于该模式的语义仓储多是在三元组表下增加索引，将三元表中的每列都放在索引中，以此来减少自联接查询带来的开销。

表 1 三元组表模式

Subj.	Prop.	Obj.
ID1	type	Student
ID1	name	“Zou”
ID1	school	“cas”
ID1	born	“1983”
ID2	type	Teacher
ID2	name	“Zhang”
ID2	post	“professor”
ID2	born	“1971”
ID2	sex	“male”
ID3	type	Student
ID3	name	“Wang”
ID3	sex	“female”
ID4	type	Artist
ID4	name	“Qian”
ID5	type	Teacher
ID5	name	“Huang”
ID5	sex	“female”

而在实际应用系统中，多是采用基于该模式的改进方案，例如：Virtuoso[12]将三元组表扩展成四元组表，增加了一列 G 用于对命名图（Name Graph）的支持，并增加了新的数据类型 IRI_ID 和 ANY 用于和辅组表中的数据相对应，来减少存储空间。而 Minerva 则将类和属性信息与实例信息相分离进行单独存储，将 typeOf 类型和非 typeOf 类型的三元组分别存放在不同的三元组表中，这种改进的三元组表模式在执行查询时将在一张包含大量三元组表的自联接查询转化为若干个小规模表上的外联接查询，以此来提高查询的效率[15]。

（2）垂直分割模式

垂直分割模式是 Abadi 等提出的一种用于存储 RDF 数据的新模式[16]，该模式根据三元组谓语不同，将拥有相同谓语的三元组存储在同一张表中。由于每张表的谓语都相同，因此可以把谓语去掉，仅保留两列，分别对应主语和宾语，垂直分割模式有时也被称为二元表模式，如表 2 所示。在实际操作中，可以利用完全存储分解模型（Fully Decomposed Storage Model，DSM）对 RDF 数据进行分割，存放于 n 个二元表中，其中 n 是数据集中谓语的数量，DLDB-OWL [17]和基于 PostgreSQL 的 Sesame[7]都是采用了这种模式。这种模式在执行查询时通过减少遍历空间，来提高数据存取的效率，但其灵活性较差，当本体模式发生改变时

需要在数据库中增加和删除相应的表，以适应新的本体模式。所以，该模式不适合存储那些包含数万个类的本体，例如：SnoMed 本体，太多的表会严重增加数据库的开销[18]。

表 2 垂直分割模式

Type

ID1	Student
ID2	Teacher
ID3	Student
ID4	Artist
ID5	Teacher

Name

ID1	“Zou”
ID2	“Zhang”
ID3	“Wang”
ID4	“Qian”
ID5	“Huang”

Born

ID1	“1983”
ID2	“1971”

Sex

ID2	“male”
ID3	“female”
ID5	“female”

School

ID1	“cas”

Post

ID2	“professor”

为了提高存储的性能，Olivier Cur´e 等人提出了一种只针对顶级属性分割的方法，并为二元表增加一列谓语，这种方法结合了垂直分割模式和三元组模式的优点，并以此实现了 roStore 系统[19]。Abadi 还提出了在列数据库上来实现垂直分割模式[16]，如表 3 所示，在列存储数据库中，即使实例在某个属性上的取值为空值，在对应表中也需为其保留一行位置[20]，为了减少空值和重复值对存储空间和查询性能的影响，Google 的 BigTable[21]以及 HBase [22]等列数据库采用数据压缩技术来处理数据稀疏的问题，Abadi 等人的研究也表明采用数据压缩技术的列数据库能够显著提高 RDF 数据的查询效率[16]。另外，Ontotext 实验室的 Atanas Kiryakov 等人还曾用“语义仓储=推理引擎+列数据库”的等式来表述语义仓储和列数据库之间的关系[23]，可见列数据库在处理数据模式迅速改变的情况的确有自己的优势。

表 3 基于列数据库的垂直分割模式

Subj.	Type	Name	Born	School	Post	Sex
ID1	Student	“Zou”	“1983”	“cas”	NULL	NULL
ID2	Teacher	“Zhang”	“1971”	NULL	“professor”	“male”
ID3	Student	“Wang”	NULL	NULL	NULL	“female”
ID4	Artist	“Qian”	NULL	NULL	NULL	NULL
ID5	Teacher	“Huang”	NULL	NULL	NULL	“female”

（3）属性表模式

如果将数据集中的主语和谓语都当成一个独立的列，就可以利用一张表对所有的 RDF 数据进行存储，这种存储方式就是属性表模式，如表 4 所示。这种设计模式的最大优点是减少了查询带来的自联接问题，但和垂直分割模式一样都是对数据集结构敏感的一种实现模式，不适合存储包含太多类和属性的数据集，限制了其灵活性；其次，该模式不能对多值属性进行处理，而多值属性在 RDF 数据集中正频繁出现；另外，空值太多会造成表中数据的稀疏，导致存储空间的浪费，所以这种模式比较适合存储属性较少且结构相对稳定的数据集。

表 4 属性表模式

Subj.	Type	Name	Born	School	Post	Sex
ID1	Student	“Zou”	“1983”	“cas”	NULL	NULL
ID2	Teacher	“Zhang”	“1971”	NULL	“professor”	“male”
ID3	Student	“Wang”	NULL	NULL	NULL	“female”
ID4	Artist	“Qian”	NULL	NULL	NULL	NULL
ID5	Teacher	“Huang”	NULL	NULL	NULL	“female”

在实际的使用中，Jena2 首先提出了两种优化策略[16]：1）聚类属性表（Clustered Property Table），对拥有相似属性特征的主语进行聚类，建立属性表，如表 5（a）所示，type、name 和 sex 属性被定义为属性表中的列，而剩余的三元组则单独存储在另一个三元组表中。2）属性－类表（Property-Class Table），利用 rdf:type 属性对相似的主语进行聚类，并放在同一张表中，这样就避免了针对同一主语查询的自联接，如表 5（b）所示。

表 5 (a) Jena2 的聚类属性表

Property Table

Subj.	Type	Name	Sex
ID1	Student	“Zou”	NULL
ID2	Teacher	“Zhang”	“male”
ID3	Student	“Wang”	“female”
ID4	Artist	“Qian”	NULL
ID5	Teacher	“Huang”	“female”

Left-Over Triples

Subj.	Prop.	Obj.
ID1	school	“cas”
ID1	born	“1983”
ID2	born	“1971”

表 5(b) Jena2 的属性-类表

Class: Student

Subj.	Name	Born	School
ID1	“Zou”	“1983”	“cas”
ID3	“Wang”	NULL	NULL

Class: Teacher

Subj.	Name	Born	Sex
ID2	“Zhang”	“1971”	“male”
ID5	“Huang”	NULL	“female”

Left-Over Triples

Subj.	Prop.	Obj.
ID2	post	“professor”
ID3	sex	“female”
ID4	type	Artist
ID4	name	“Qian”

2.3 原生方式存储模式

原生方式也称为 Native 方式，是建立在文件系统之上，专门针对 RDF 图的特点设计的存储模式，能够灵活地适应数据模式的变化，摆脱传统数据库普遍存在的“阻抗失配”问题。在该模式中，完全索引是一种较有影响力的设计方法，通过建立 PSO、POS、SPO、SOP、OPS 和 OSP（S 代表主语、P 代表谓语、O 代表宾语）6 个索引，来包含三元组中三个元素所有可能的排序方式，根据这一思想实现的语义仓储 YARS 对三元组及其上下文信息建立了六个 B+树索引[24]，而 Kowari 则利用 AVL 和 B 树来代替 B+树建立索引，完全索引模式是一种牺牲存储空间和数据更新速度来保证查询效率的一种方式。

而其它原生方式存储模式也是围绕如何根据 RDF 图的特点设计更加高效的索引方式展开研究，例如：4Store 通过 radix 树建立 R 索引、M 索引以及 P 索引来实现 RDF 数据存储和高效查询[25]；Ralf Heese 等人将相关的三元组放在同一个数据库页面上，通过建立位集合索引（bitset index）来指示三元组所在的数据库页面，以此提高查询效率[26]；Ying Yan 等为了使 RDF 查询引擎高效快速地在小范围内定位到查询结果，把 RDF 图分成若干个子图，分别进行存储，并建立 URI 签名树，签名树通过 URI 对包含查询结果的子图进行快速的定位[27]；HStar 提出了层次存储系统，将 typeOf 类型的三元组在类的层次上进行存储，而非 typeOf 类型的三元组则在属性的层次上进行存储，并利用 B+树对三元组进行索引，提高了查询效率[28]。

基于原生方式存储的语义仓储还有：AllegroGraph[29]、BigOWLIM [5]、Bigdata[30]和 Jena TDB[31]等。与基于传统数据库相比，基于原生方式存储具有更大的灵活性，能够减少数据加载和更新时间。但 Ma 的研究员也表明，该模式在三元组的顺序发生变化时，会导致查询时间增加数 10 倍（甚至更多）[40]，所以在查询过程中必须实施高效的优化策略，除此之外基于原生方式的语义仓储还要实现传统数据库的很多其它功能，例如：事务处理、访问控制、日志和数据恢复等。随着相关技术的不断完善，基于原生方式的语义仓储处理 RDF 数据的能力已经有了很大的提高，例如：AllegroGraph 的处理能力超过 1 万亿个三元组[32]，笔者认为采用原生方式的存储可能是未来语义仓储发展的方向。

3 分布式存储策略

为了突破 I/O 和主存对 RDF 数据存储和查询能力的限制，进一步提高系统的可扩展性以及数据访问性能，研究人员将目光转向了分布式存储策略上，对数据进行并行处理，根据分布式网络的拓扑结构和各个结点在网络中所充当的角色不同，分布式存语义仓储主要有两种组织形式：集中式语义仓储（Clustered Semantic Repository）和自组织语义仓储（Self-Organized Semantic Repository）。

3.1 集中式语义仓储

集中式语义仓储又称为集群语义仓储，是目前分布式语义仓储中采用最多的方法，在集中式分布网络中通常具有两种类型的结点：存储结点和控制结点，存储结点主要负责对 RDF 数据进行存储；而控制结点则在前端对用户提供服务，例如：当用户查询时，控制结点对查询请求进行分发，并对各个查询结点的响应结果进行组配.集中式语义仓储的一般模式如图 2 所示，C 为控制结点，S1、S2、S3、S4 和 S5 为存储结点。该类型的语义仓储一个显著缺点就是系统的健壮性问题，一个结点的故障可能导致所有的结点不能正常工作；另外控制结点需对所有的用户请求做出反应，易造成系统的瓶颈。

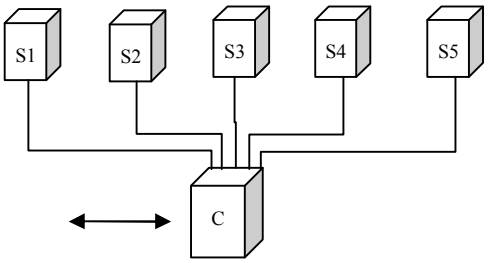

图 2 集中式语义仓储一般模式

集中式语义仓储在具体的实施过程中，不同的系统又有自己不同的特点，例如： 4Store[25]、Virtuoso[12]和 YARS2[34]等利用哈希分割算法把索引映射到不同的存储结点上，查询时根据用户请求控制结点使用同样的哈希函数来确定所涉及的三元组所在的存储结点，取出相应的数据进行响应。利用哈希函数进行索引分割的语义仓储系统是目前主流的分割算法，除此之外还有一些其它的分割算法，例如： Jiewen Huang 等人提出的图分割算法[35]，根据图数据模型对索引进行分割，这种方法将图中相近的三元组存储在同一结点上，为了最大化的对查询进行并行处理，算法允许各个结点拥有重叠的数据，并把查询分解成若干能够独立执行的块，随后利用 Hadoop MapReduce 架构对这些块的返回结果进行重新组装，这样在各个结点之间就无需进行通信，减少了查询响应时间。

值得一提的是 BigOWLIM 语义仓储系统，其并没有采用分割算法对索引进行分割[36]，而是将全部的索引都存放在各个工作结点（Workers Node）上，系统中的控制结点（Masters Node）作为集群的网关，所有的读/写请求都要通过这些结点。各个工作结点都是独立的 OWLIM-Enterprise 实例，所以集群在只有一个工作结点的情况下仍然可以正常工作，各个控制结点通过监测各个工作结点的查询请求队列，将新的查询请求分发给等待时间最短的工作结点。

3.2 自组织语义仓储

为了应对集中式系统中控制结点容易造成系统瓶颈的问题，研究者们提出了自组织语义仓储，其中各结点之间通过协调工作来执行集中式系统中控制结点所承担的高效控制集群结点的任务，在自组织网络中根据网络结构不同分为：非结构化重叠网络（Unstructured Overlay Network）、结构化重叠网络（Structured Overlay Network）以及两者的混合模式（Hybrid Overlay Network）[37]。在非结构化重叠网络中，各个结点之间通过无约束的随机连接进行通信，虽然这种组织模式简单，但其也限制了系统的可扩性、系统的查找时间较长，并且维护成本也较高，基于这一模式的语义仓储有：Bibster 和 S-RDF 等[37]；在结构化重叠网络中，结点被组织成明确的几何拓扑结构，例如：环形、立方体和树形等，与非结构重叠网络相比其更能对查询响应时间和结点的维护进行保证，基于这一模式的系统有：Edutella、RDFPeers 和 GridVine 等[33]；而

在混合模式的架构中，其部分采用了非结构化重叠网络中的随机连接方法，部分采用了结构化重叠网络中的拓扑结构，其中特定的拓扑结构是混合模式架构的核心，例如：BT Loo 等人提出的混合策略[37]。在这三种模式中又以结构化重叠网络的模式使用最为普遍，下面将以 RDFPeers 为例对这种模式进行分析。

RDFPeers 系统[38]是第一个提出使用 DHTs（Distributed Hash Tables）的 P2P（Peer-to-Peer）语义仓储系统，DHTs 被用于对 RDF 数据进行分割和定位三元组所在的存储结点，其网络结构是基于环形的拓扑结构，所有结点至少与环中的一个结点相连接，任何一个结点接收到的请求都会提交到查询结果所在的结点进行处理，例如：图 3 中 S6 接收到对关键字 1.1 的请求，会通过 S1 传递到 S2 进行处理[37]。

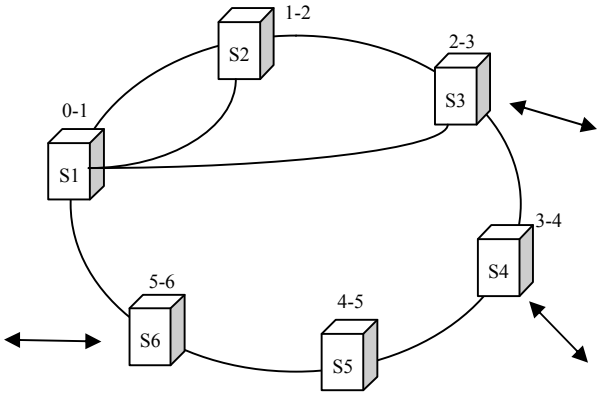

图 3 RDFPeers 系统架构[37]

注：此图来源于 I. Filali 和 F. Bongiovanni 等人的 A Survey of Structured P2P Systems for RDF Data Storage and Retrieval。

4 测试基准

语义仓储的迅速发展不但受益于人们对关联数据的热情、数据整合的需要、相关标准的制定以及硬件性能的提高等方面，测试基准的发展和规范化也是促进语义仓储发展的一个重要方面，也是语义仓储领域相关研究逐渐走向成熟的一个标志。测试基准可以在数据的加载速度、系统扩展性、查询速度以及推理能力等方面对语义仓储的性能进行评估，下面对目前较为流行的测试基准做简单地分析。

4.1 里海大学测试基准

里海大学测试基准（Lehigh University Benchmark，LUBM）是里海大学开发的一个测试基准，测试数据集可通过数据集生成器和一个固定的 OWL 大学领域本体自动生成[39]，并可通过配置大学的数量来生成相应规模的数据集，例如：LUBM（50）和 LUBM（90K）等，LUBM 定义了 14 个查询语句（Q1－Q14）来测定语义仓储的查询性能。但在 LUBM 数据集中的数据关联度不高，与基于真实数据集的测试结果存在偏差，基于此 IBM 中国研究院对 LUBM 进行了扩展提出了 UOBM，通过完善不同学校的学生之间的联系，使得数据更接近于真实世界数据[40]。

4.2 基于事实的测试基准

基于事实的测试基准 DBPedia 是从维基百科中抽取的结构化数据集，在他们最新发布的版本中，DBPedia 对超过 364 万个事物进行了描述，包括人物、地点、音乐专辑和电影等，包含了超过 10 亿个三元组，其中的 3.85 亿来自于对英文版维基百科的抽取，目前 DBPedia 作为关联开放数据云的中心存在，DBpedia

包含的信息的真实性、多样性和广博性，使其作为语义仓储的测试基准拥有得天独厚的优势[41]。

4.3 柏林测试基准

柏林测试基准 BSBM（Berlin SPARQL Benchmark）建立在电子商务用例的基础上，数据集包括来自各类站点的不同供应商和消费者提供的产品数据集，利用混合查询（Query Mix，每个包含 25 个查询）来模拟消费者在选择商品时的搜索和导航行为。BSBM 通过 SPARQL 端点（SPARQL Endpoint）对原生方式语义仓储，基于关系数据库的语义仓储以及任何提供 SPARQL 查询的仓储进行评估，BSBM 还可以通过改变数据集的大小和仿真客户端的数量来测试语义仓储在不同数据集量级和访问压力下的性能。目前 BSBM 共发布了三次测试报告，在最近的 2011 年的测试中，BSBM 对 Virtuoso、BigOWLIM、4Store、BigData 以及 Jena TDB 分别在包含 1 亿和 2 亿三元组的数据集上进行了测试[42]。

5 应用系统

目前，基于语义仓储的研究和实践已经取得了一定的成果，其中典型的应用系统有：Virtuoso、BigOWLIM、BigData、3/4/5Store、AllegroGraph、Oracle 11g、Jena TDB、Jena SDB、Sesame 和 Redland 等，这些系统已经从理论研究发展到实际的应用阶段，例如：BigOWLIM 被用于 BBC 的 2010 年世界杯网站和英国报业协会等[36]；VIVO[43]计划在 1.5 版本中采用 Virtuoso 作为底层语义仓储；另外，在笔者参加的国家科技支撑计划项目中也拟采用 Virtuoso 对超级科技词表和本体进行存储和管理，以支持在海量文献信息中进行知识发现和推理等。在这些系统中 Jena、Sesame 和 Redland 则更多地作为语义架构被使用，它可以独立于任何具体存储底层，例如：Virtuoso 和 BigOWLIM 都实现了 Jena 和 Sesame 的底层存储接口，各个语义仓储系统之间的关系如图 4 所示[44]。

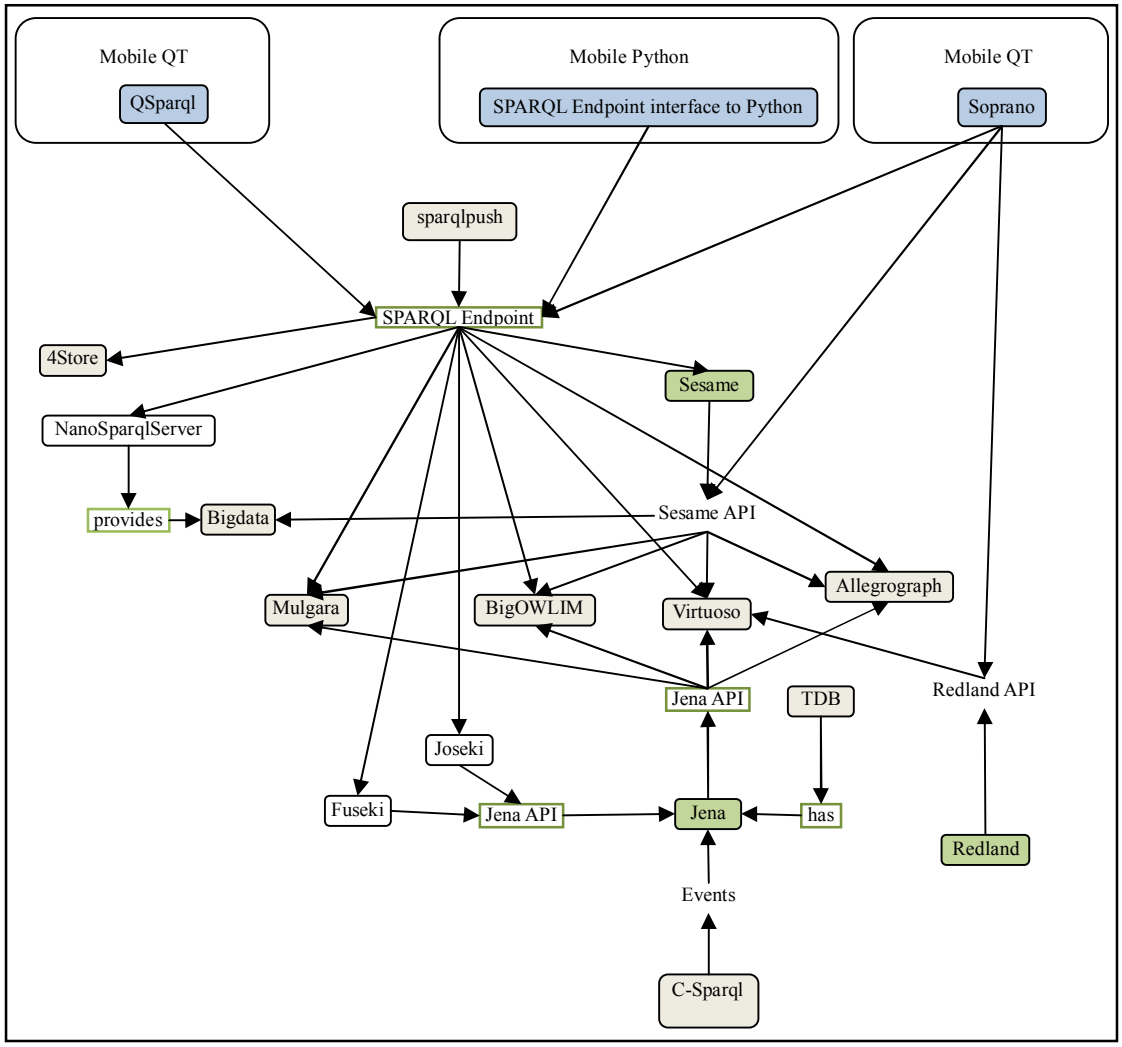

图 4 各个语义仓储系统之间的关系[44]

注：此图来源于 Smart - M3 Storage Solutions，原文地址：http://www.diem.fi/files/deliverables/D5.6.3_Storage-solutions.pdf。

6 总结

本文主要对语义仓储的存储模式、分布式存储策略、测试基准和应用系统的研究进展进行了分析，但限于篇幅本文没有涉及到语义仓储的查询处理和优化、推理机制以及对各个语义仓储在加载、查询和推理上的各种评测结果等方面的研究成果。就现阶段而言，笔者认为比较成熟的语义仓储构建技术是在现有的（对象）关系数据库上增加 RDF 数据存储和管理机制，例如：Virtuoso 和 Oracle RDF，这种方式可以充分利用现有关系数据库的查询优化、事务处理、访问控制、日志和数据恢复等技术。就将来的发展趋势来看，根据 RDF 图的特点设计的原生方式应该有更大的发展空间，其能够处理更加全面的模式和本体，加载更多的数据，但原生方式语义仓储还需在查询处理和优化技术、索引设计、推理功能以及分布式存储技术等方面进行深入的研究。

参考文献：

[1] LinkingOpenData. [EB/OL]. [2012-04-22]. http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData.
[2] LOD Cloud Cache. [EB/OL]. [2012-04-22]. http://lod.openlinksw.com/.
[3] Triplestore [EB/OL]. [2012-04-22]. http://en.wikipedia.org/wiki/Triplestore.

[4] B. Haslhofer, E. Momeni, B. Schandl, et al. Europeana RDF store report[R]. Europe: Europeana, 2011.
[5] OWLIM. [EB/OL]. [2012-04-22]. http://www.ontotext.com/owlim.
[6] Swoogle Semantic Web Search. [EB/OL]. [2012-04-22]. http://swoogle.umbc.edu/.
[7] Sesame. [EB/OL]. [2012-04-22]. http://www.openrdf.org.
[8] Jena. [EB/OL]. [2012-04-22]. http://incubator.apache.org/jena/.
[9] OWLJessKB: A Semantic Web Reasoning Tool. [EB/OL]. [2012-04-22]. http://edge.cs.drexel.edu/assemblies/software/owljesskb/.
[10] S. Harris, D.N. Gibbins. 3store: Efficient bulk RDF storage[C],// Proceedings of the 1st International Workshop on Practical and Scalable Semantic Systems(PSSS'03), Sanibel Island, Florida, USA, 2003:1-15.
[11] G. Ladwig, A. Harth. CumulusRDF: Linked Data Management on Nested Key-Value Stores[C],// Proceedings of the 7th International Workshop on Scalable Semantic Web Knowledge Base Systems(SSWS'11), Bonn, Germany, 2011:30-43.
[12] Virtuoso. [EB/OL]. [2012-04-22]. http://virtuoso.openlinksw.com/.
[13] Oracle. [EB/OL]. [2012-04-22]. http://www.oracle.com/technetwork/database/options/semantic-tech/index.html.
[14] 吴刚. RDF 图数据管理的关键技术研究[D]. 北京: 清华大学, 2008.
[15] J. Zhou, L. Ma, Q. Liu, et al. Minerva: A scalable OWL ontology storage and inference system[C],//Proceedings of the 1st Asian Semantic Web Conference(ASWC'06), Beijing, China, 2006:429-443.
[16] D.J. Abadi, A. Marcus, S.R. Madden, et al. Scalable semantic web data management using vertical partitioning[C],// Proceedings of the 33rd international conference on very large data bases(VLDB '07), Vienna, Austria, 2007:411-422.
[17] DLDB-OWL. [EB/OL]. [2012-04-22]. http://swat.cse.lehigh.edu/downloads/dldb-owl.html.
[18] S. Heymans, L. Ma, D. Anicic, et al. Ontology reasoning with large data repositories[J]. Ontology Management, 2008(7):89-128.
[19] O. Curé, D. Faye, G. Blin. Towards a better insight of RDF triples Ontology-guided Storage system abilities[C],//Proceedings of the 6th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS'10), Shanghai, China, 2010:1-16
[20] 杜小勇, 王琰, 吕彬. 语义 Web 数据管理研究进展[J]. 软件学报, 2009(20):2950-2964.
[21] Bigtable. [EB/OL]. [2012-04-22]. http://research.google.com/archive/bigtable.html.
[22] HBase. [EB/OL]. [2012-04-22]. http://hbase.apache.org/.
[23] A. Kiryakov, M. Damova. Storing The Semantic Web: Repositories.. Chapter 7 in: Semantic Web Handbook[M]. Heidelberg,Germany: Springer Verlag, 2011.
[24] C. David, C. Olivier, B. Guillaume. A survey of RDF storage approaches[J]. ARIMA Journal, 2012(15):11-35.
[25] S. Harris, N. Lamb, N. Shadbolt. 4store: The design and implementation of a clustered rdf store[C],// Proceedings of the 5th International Workshop on Scalable Semantic Web Knowledge Base Systems (SSWS'09), Washington DC, USA, 2009:94-109.
[26] R. Heese, M. Znamirowski. Resource centered RDF data management[C],// Proceedings of the 7th International Workshop on Scalable Semantic Web Knowledge Base Systems(SSWS'11), Bonn, Germany, 2011:138-153.
[27] Y. Yan, C. Wang, A. Zhou, et al. Efficient indices using graph partitioning in rdf triple stores[C],//Proceedings of the 25th IEEE International Conference on Data Engineering(ICDE'09), California, USA, 2009:1263-1266.
[28] Y. Chen, J. Ou, Y. Jiang, et al. HStar–a semantic repository for large scale OWL documents[C],//Proceedings of the First Asian Semantic Web Conference (ASWC'06), Beijing, China, 2006:415-428.
[29] AllegroGraph. [EB/OL]. [2012-04-22]. http://www.franz.com/agraph/allegrograph/.
[30] Bigdata. [EB/OL]. [2012-04-22]. http://www.bigdata.com/blog/.
[31] Jena SDB. [EB/OL]. [2012-04-22]. http://incubator.apache.org/jena/documentation/sdb/index.html.
[32] LargeTripleStores. [OL]. [2012-04-22]. http://www.w3.org/wiki/LargeTripleStores.
[33] H. Mühleisen, T. Walther, R. Tolksdorf. A survey on self-organized semantic storage[J]. International Journal of Web Information Systems, 2011(7):205-222.
[34] YARS2. [EB/OL]. [2012-04-22]. https://grenada.lumc.nl/LOVD2/mendelian_genes/home.php?select_db=YARS2.
[35] J. Huang, D.J. Abadi, K. Ren. Scalable sparql querying of large rdf graphs[J]. Proceedings of the VLDB Endowment, 2011(4).
[36] B. Bishop, A. Kiryakov, D. Ognyanoff, et al. OWLIM: A family of scalable semantic repositories[J]. Semantic Web, 2011(2):33-42.
[37] I. Filali, F. Bongiovanni, F. Huet, et al. A Survey of Structured P2P Systems for RDF Data Storage and Retrieval[J]. Transactions on Large-Scale Data-and Knowledge-Centered Systems III, 2011:20-55.
[38] M. Cai, M. Frank. RDFPeers: a scalable distributed RDF repository based on a structured peer-to-peer network[C],// Proceedings of the 13th international conference on World Wide Web(WWW'04), 2004:650-657.
[39] SWAT Projects - the Lehigh University Benchmark (LUBM) [EB/OL]. [2012-04-22]. http://swat.cse.lehigh.edu/projects/lubm/.
[40] L. Ma, Y. Yang, Z. Qiu,et al. Towards a complete OWL ontology benchmark[J]. The Semantic Web: Research and

Applications, 2006:125-139.
[41] DBPedia. [EB/OL]. [2012-04-22]. http://dbpedia.org/About.
[42] Berlin SPARQL Benchmark. [EB/OL]. [2012-04-22]. http://www4.wiwiss.fu-berlin.de/bizer/berlinsparqlbenchmark/.
[43] VIVO. [EB/OL]. [2012-04-22]. http://vivoweb.org/.
[44] Smart - M3 Storage Solutions. [R/OL]. [2012-04-22]. http://www.diem.fi/files/deliverables/D5.6.3_Storage-solutions.pdf.

（责任编辑 化柏林）