基于HBase的本体存储模型

宋华珠 段文军 刘 翔
(武汉理工大学计算机科学与技术学院 武汉 430070)

摘 要 本体是对某一特定领域的重要概念的形式化描述。合理地存储本体数据是发挥其共享性的重要前提,尤其是在当前分布式系统下其作用更为突出。通过分析目前的各种存储方式,并结合当前语义网、Hadoop的特点,提出了基于HBase的本体存储模型HBase-OntSM,该模型将本体的三元组数据集作为一个图,把图作为一条记录存储到数据库中;并给出了与该图相关的一系列基本定义和索引定义。最后以西藏文化本体中的一个片段为例,解释了该本体存储模型及其存储过程。

关键词 本体存储模型,HBase,Hadoop
中图法分类号 TP311 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.6.008

Ontology Storage Model Based on HBase

SONG Hua-zhu DUAN Wen-jun LIU Xiang
(School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China)

Abstract Ontology is a formal description of important concepts within a specific domain. Reasonable storage of ontological data is an important prerequisite to perform its sharing features, and its function is more outstanding especially under the current distributed systems. By analyzing different means of storing ontology data at present and combining the features of semantic Web and Hadoop, this paper came up with an ontology storage model based on HBase, called HBase-OntSM, which regards ontology's triple dataset as a graph storing it in a database as a record, and then presented a series of basic definitions and index definitions related to the graph. Finally, this paper took a segment of Tibetan culture ontology for an example, and explained the ontology storage model and its stored procedures.

Keywords Ontology storage model, HBase, Hadoop

在信息学科中,本体是知识的形式化表示,作为领域内的概念集及概念之间的关系,可以用于推理或描述领域的实体。本体作为语义网发展的支撑技术之一,语义网的快速发展加速了其在Web技术领域的研究进程。随着本体研究的深入,大型、复杂的本体不断被开发出来,构建的本体能否完好、合理地保存将直接影响到后期对本体的管理和复用。目前本体的存储方法主要有文本存储和数据库存储[1]。文本存储是将本体数据以文件的形式存储在本体文件系统中。这种存储方式简单、灵活,但存储效率很低且存储规模有限。数据库存储方式是将本体数据按照一定的组织策略存放在数据库中。由于数据库技术发展成熟,这种存储方式效率较高,但系统设计复杂、可扩展性差。

作为一个非关系数据库,HBase是一个比较适合非结构化数据存储的数据库,它依托于Hadoop的HDFS,可以方便地通过Hadoop的Map/Reduce框架对HBase进行数据操作。2010年,John Abraham等人提出了一种三表存储方式[2],他们将本体数据存储在Ta、Tp和To 3张表中。这种存储方式能够很快返回(S,?P,?O),(?S,P,?O),(?S,?P,O)模式的查询,但是针对(S,?P,O),(S,P,?O),(?S,P,O)模式的查询,就需要将列族中的数据按分割号分隔开,再次进行查询。Artem Chebotko等提出使用(S_PO,P_SO,O_SP,PS_O,SO_P,PO_S) 6个HBase表格来存储数据[3],每个表格只有一个名为VALUE的列族,针对不同的查询请求,使用这种表格设计可以明确查询的表格。但是由于不同表中的数据无法重用,因此会产生额外的开销,它需要的存储空间是原始数据占用空间的6倍之多。在Sun J等[4]的基础上,Papailiou N等提出用(SPO,POS,OSP)3张表来存储RDF三元组[5]。SPO表中的行键是(Subject,Predicate),列族存放宾语值;POS表的行键是(Predicate,Object),列族存放主语值;OSP表的行键是(Object,Subject),列族中存放的是谓语值。数据在HBase中存储时,是按照行键进行排序的,使用这种方案所占用的存储空间仅为六表存储方案的1/2。

1 基于HBase的本体存储模型

通过对各种存储方式的分析,受HeartProposal项目思想的启发,以及基于当前Web数据和Hadoop的特点,本文提

到稿日期:2015-06-15 返修日期:2015-08-24 本文受国家科技支撑计划项目(2012BAH33F03),中央高校基本科研业务费专项资金(2014-IV-148)资助。

宋华珠(1970—),女,博士,副教授,主要研究方向为智能方法、语义本体、数据挖掘,E-mail:shuaz@whut.edu.cn;段文军(1990—),男,硕士生,主要研究方向为智能方法、语义本体;刘 翔(1991—),男,硕士生,主要研究方向为智能方法、语义本体。

出了基于 HBase 的本体存储模型 HBase-OntSM，其将本体的三元组数据集作为一个图，把图作为一条记录存储到数据库中；并对该图定义了一系列的索引，加快了查询速度。本节将介绍这种存储模型及设计过程中的一些定义。

1.1 基本定义

定义 1 T 是一个 RDF 三元组，$T=(S,P,O)$，S，P 和 O 分别表示 RDF 三元组中的主语、谓词和宾语。

定义 2 G 是一个 RDF 本体图，$G=\{T_1,T_2,\cdots,T_i,\cdots,T_n\}$，$n=|G|$，其中 T_i 是 RDF 三元组，$T_i\in G$，$i=1,2,\cdots,n$。

定义 3 D 是一个本体三元组数据集，将 D 分成 n 个数据集，$D=\{G_1,G_2,\cdots,G_i,\cdots,G_n\}$，对于任一 $G_i\in D$，G_i 都有唯一标识 $G_i.id$，$G_i.id$ 是由本体名＋i 组成，$i=1,2,\cdots,n$。可以把 D 看作一个本体图，G_i 则是该本体图的一个子图。

定义 2 把本体看作是由一系列三元组组成的，这些三元组中的主语、宾语和谓语分别对应于本体关系图中的节点和边。在本体图中，每个节点都标有一个唯一的标识符（即统一资源标识符 URI）。本体图的边表示谓语，也即属性，在本体表示中每一个谓语也是有唯一的标识符。本体图的节点表示一个主语、宾语或是一些数据值，所以主语和宾语的数量会随着节点的增加而增加，主语和宾语之间通过谓语连接。使用数据集 P 来表示一个 RDF 数据集 D 中的所有谓词，$P=P_1\cup P_2\cup\cdots\cup P_i\cup\cdots\cup P_n$，其中 $P_i=\{p|(s_i,p_i,o_i)\in G_i\}$，$G_i\in D$，$D=\{G_1,G_2,\cdots,G_i,\cdots,G_n\}$，其中 $i=1,2,\cdots,n$。

本文使用函数 *num* 来返回三元组在一个 RDF 图 G 中的位置，函数定义为：int i *num*(T_i)，其中 $T_i\in G$，$G=\{T_1,T_2,\cdots,T_i,\cdots,T_n\}$，$i=1,2,\cdots,n$。此外，定义一个反函数 Triple t *inversenum*(i) 用于返回位置 i 存储的三元组 T_i，其中 $T_i\in G$，$G=\{T_1,T_2,\cdots,T_i,\cdots,T_n\}$，$i=1,2,\cdots,n$。

定义 4 基本图形模式（Basic Graph Pattern，BGP）是一个三元组模式集 $\{TP_1,TP_2,\cdots,TP_i,\cdots,TP_n\}$，其中 $n\geqslant 1$，$i=1,2,\cdots,n$，对任意 $TP_i\in BGP$，有 $TP_i=(sp_i,pp_i,op_i)$，sp，pp 和 op 分别表示主语模式、谓语模式和宾语模式。

1.2 索引定义

匹配一个 RDF 图的 BGP 需要在 RDF 三元组上匹配查询三元组。每个三元组模式会产生一个中间三元组集，这些中间数据需要经过进一步连接查找才能找到匹配的子集。为了加快这个计算过程，下面定义几个位图索引。

定义 5 属性索引 Ip 是 RDF 图 $G\in D$ 的一个元组集合 $\{(p_1,v_1),(p_2,v_2),\cdots,(p_i,v_i),\cdots,(p_n,v_n)\}$，其中 $p_i\in P$ 是 RDF 数据集 D 的谓词集合中的一个谓词，$n=|P|$，$i=1,2,\cdots,n$，v_i 是一个维度为 $|G|$ 的位向量，其中当且仅当三元组 $T_k=$ *inversenum*$(k)(T_k\in G)$ 且 $T_k.p=p_i$ 时，向量 v_i 中第 k 个位置的值为 1。

索引 Ip 有助于快速定位一个具有特定谓词的三元组在 RDF 图中的位置。$Ip(p)$ 表示谓词 p 的位向量。该位向量的维度等于 RDF 图中的三元组的个数 $|G|$。在索引中向量的数量等于谓词的个数 $|P|$，可以看到谓词个数是比较少的（$|P|<|G|$）。

同上，可以定义主语索引 Is 和宾语索引 Io。如果想在 RDF 图中找到一个主语、谓语已知的元组数据 (s_1,p_1)，可以对 Is，Ip 做逻辑与运算 $Is(s_1)\wedge Ip(p_1)$；若主语、谓语和宾语都已知即 (s_1,p_1,o_1)，则需进行两次逻辑与运算，计算公式为：$Is(s_1)\wedge Ip(p_1)\wedge Io(o_1)$。

通过定义 Is，Ip 和 Io，已经可以加速三元组的匹配。下面要定义的索引用来连接通过三元组匹配到的子图得到的中间结果。

定义 6 索引 Iss 是 RDF 图 G 的一个元组集合 $\{(1,v_1),(2,v_2),\cdots,(i,v_i),\cdots,(n,v_n)\}$，其中 $n=|G|$，$1,2,\cdots,n$ 为 G 中连续的三元组的位置，$i=1,2,\cdots,n$。v_i 是一个维度为 $|G|$ 的位向量，当且仅当 $T_k=$ *inversenum*$(k)(T_k\in G)$，$T_i=$ *inversenum*$(i)(T_i\in G)$ 且 $T_k.s=T_i.s$ 时，v_i 中第 k 个位置的值为 1。即，在 Iss 中若位向量 v_i 中第 k 个值为 1，说明 G 中第 i 个三元组和第 k 个三元组的主语相同。

定义 7 索引 Ioo 是 RDF 图 G 的一个元组集合 $\{(1,v_1),(2,v_2),\cdots,(i,v_i),\cdots,(n,v_n)\}$，其中 $n=|G|$，$1,2,\cdots,n$ 为 G 中连续的三元组的位置，$i=1,2,\cdots,n$，v_i 是一个维度为 $|G|$ 的位向量，当且仅当 $T_k=$ *inversenum*$(k)(T_k\in G)$，$T_i=$ *inversenum*$(i)(T_i\in G)$ 且 $T_k.o=T_i.o$ 时，v_i 中第 k 个位置的值为 1。即，在 Ioo 中若位向量 v_i 中第 k 个值为 1，说明 G 中第 i 个三元组和第 k 个三元组的宾语相同。

定义 8 索引 Iso 是 RDF 图 G 的一个元组集合 $\{(1,v_1),(2,v_2),\cdots,(i,v_i),\cdots,(n,v_n)\}$，其中 $n=|G|$，$1,2,\cdots,n$ 为 G 中连续的三元组的位置，$i=1,2,\cdots,n$，v_i 是一个维度为 $|G|$ 的位向量，当且仅当 $T_k=$ *inversenum*$(k)(T_k\in G)$，$T_i=$ *inversenum*$(i)(T_i\in G)$ 且 $T_k.o=T_i.s$ 时，v_i 中第 k 个位置的值为 1。即，在 Iss 中若位向量 v_i 中第 k 个值为 1，说明 G 中第 i 个三元组的主语和第 k 个三元组的宾语相同。

定义 9 索引 Ios 是 RDF 图 G 的一个元组集合 $\{(1,v_1),(2,v_2),\cdots,(i,v_i),\cdots,(n,v_n)\}$，其中 $n=|G|$，$1,2,\cdots,n$ 为 G 中连续的三元组的位置，$i=1,2,\cdots,n$，v_i 是一个维度为 $|G|$ 的位向量，当且仅当 $T_k=$ *inversenum*$(k)(T_k\in G)$，$T_i=$ *inversenum*$(i)(T_i\in G)$ 且 $T_k.s=T_i.so$，v_i 中第 k 个位置的值为 1。即，在 Iss 中若位向量 v_i 中第 k 个值为 1，说明 G 中第 i 个三元组的宾语和第 k 个三元组的主语相同。

对于一个给定的图 G，索引 Iss，Ioo，Iso 和 Ios 占用 $|G|\times|G|$ 位的数据空间。它们可以用于快速匹配，找出两个集合中主语与主语、宾语与宾语、主语与宾语、宾语与主语相同的三元组。直观地说，如果 RDF 数据集中位置 i 对应的三元组 $T_i=$ *inversenum*(i)，$T_i\in G$，那么通过 $Iss(i)$ 可以获得与 T_i 有相同主语的三元组集合。

本文称 Is，Ip 和 Io 为选择索引，称 Iss，Ioo，Iso 和 Ios 为连接索引。需要注意，Ios 索引可以从 Iso 索引中获得，即对每一个 Ios 中的每一个 v_i 进行转置就可以获得 Iso 中的 v_i，反之亦然。在实践中，可以在 Iso 与 Ios 中选择其一。

1.3 存储模式设计

HBase-OntSM 存储模型用一种稀疏多维排序的方式来将数据存储到 HBase 数据库中。HBase 是一种基于 HDFS 的 Bigtable 类型的数据库，它的表结构与传统的关系型数据库是不同的[6]。一个 HBase 表的每一行都有一个行键（row-key）和多个列族，同一列在两个不同的行中存储的数据类型不必相同。列族是在创建表时定义的，但列族中的列可以动态地添加和删除。HBase 表中的行数据可以存储在 HBase 集群中的不同机器上，可以根据行键对这些数据进行高效的检索。本文提出一个用单表存储本体数据的方案，如表 1 所列。

表 1　HBase-OnoSM 存储模型

Row-id	Column-family							
	Data		Index					
	graph	graphids	Is	Ip	Io	Iss	Ioo	Iso
G_1. id	G_1	id_1	Is_1	Ip_1	Io_1	Iss_1	Ioo_1	Iso_1
G_2. id	G_2	id_2	Is_2	Ip_2	Io_2	Iss_2	Ioo_2	Iso_2
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
G_n. id	G_n	id_n	Is_n	Ip_n	Io_n	Iss_n	Ioo_n	Iso_n

对表 1 分析如下：

1)表的行键是一个本体标识符，可以是该本体的文件名，若该本体过大而被分割为多个 graph 进行存储，可以在本体名后加编号；

2)第一个列族是数据列族，它分为 3 列，graph 列存储着这个本体图的完整三元组数据，这些数据以固定的顺序排列，三元组位置的变动会影响该行数据索引的变动，当一个本体过大而被分割为多个 graph 时，graphids 列用于存储该本体其他相关行的 id；

3)第二个列族是索引列族，其中存储着前文计算的索引数据。

在存储时，需要将一个本体划分成多个子图。关于本体子图划分，目前也有比较多的研究方案，如哈希分割、边界分割以及半径分割等。为了避免将一个图划分在不同的机器上时可能发生的不必要的数据传输，本文将每个 RDF 图存储为一个值，而不是将其分割成子图甚至是三元组。

2　基于 HBase 的数据存储过程

数据从 ref 本体文件到 Hadoop 平台上，HBase 数据表的整个存储过程，如图 1 所示。

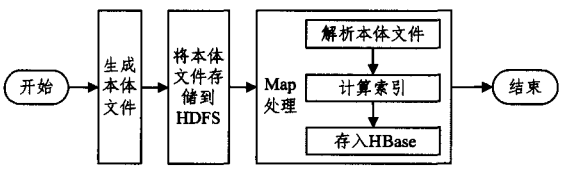

图 1　数据存储过程

本节以西藏文化本体中的一个片段为例，来解释本体存储过程。

1)生成 RDF/XML 格式的本体文件

本文中所介绍的本体源文件格式，可以使用斯坦福大学的本体编辑工具 Protégé 生成[7]。它的格式为：

```
〈rdf:RDF xmlns="http://www.semanticweb.org/ontologies/2014/3/culturedance.owl#"〉
  〈owl:NamedIndividual rdf:about="&culturedance;昌都锅庄"〉
    〈rdf:type rdf:resource="&culturedance;锅庄舞"/〉
    〈clothing rdf:resource="&culturedance;cloth2"/〉
    〈bgMusic rdf:resource="&culturedance;建立桑耶寺"/〉
    〈hasvideos rdf:resource="&culturedance;银光闪烁的王宫"/〉
  〈/owl:NamedIndividual〉
    〈owl:NamedIndividual rdf:about="&culturedance;玉树依舞"〉
    〈rdf:type rdf:resource="&culturedance;锅庄舞"/〉
    〈clothing rdf:resource="&culturedance;cloth1"/〉
    〈prop rdf:resource="&culturedance;prop1"/〉
    〈hasvideos rdf:resource="&culturedance;丰收啊丰收"/〉
    〈bgMusic rdf:resource="&culturedance;北方大草原"/〉
  〈/owl:NamedIndividual〉
  ...
〈/rdf:RDF〉
```

这个本体片段表示该本体名为“http://www.semanticweb.org/ontologies/2014/3/culturedance.owl#”，其中定义了两个实例：昌都锅庄和玉树依舞。昌都锅庄是锅庄舞类的一个实例，它的 clothing 属性值为 cloth2，bgMusic 属性值为建立桑耶寺，hasvideos 属性值为银光闪烁的王宫；玉树依舞是锅庄舞类的一个实例，它的 clothing 属性值为 1，bgMusic 属性值为北方大草原，hasvideos 属性值为丰收啊丰收。其本体图如图 2 所示。

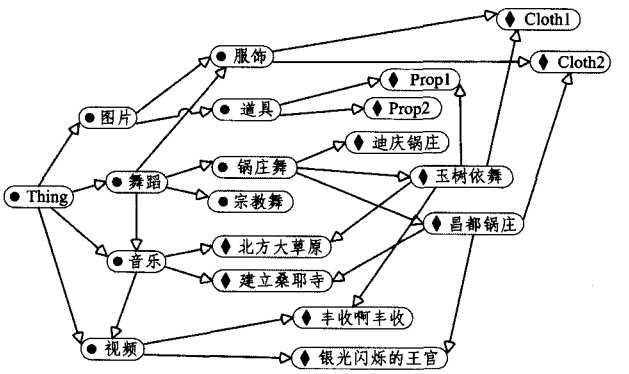

图 2　两个实例的本体图

2)将本体文件分别存储在 HDFS 上

可以利用 org.apache.hadoop.fs.FileSystem 类中的 copyFromLocalFile 方法将数据存储到分布式文件系统(HDFS)中。

3)Jena 解析处理

Jena[8]是由惠普实验室开发的一个 Java 的 RDF API。通过它可以将 XML 格式的本体数据转换成三元组。利用 ontology 包中的 ontology API 将整个 RDF 资源看作一个本体模型(ontology model)，通过 Model 访问 RDF/XML 中的 Statement。

MapReduce 调用 Jena 的类和接口，将本体文件解析为下列三元组结构：

(舞蹈，subClassOf，Thing)，(锅庄舞，subClassOf，舞蹈)，(服装，subClassOf，图片)，(昌都锅庄，clothing，cloth2)，(昌都锅庄，bgMusic，建立桑耶寺)，…，(玉树依舞，hasvideos，丰收啊丰收)，…。

4)索引计算

Map 调用相应的索引计算函数对自己分配到的三元组图集进行计算。计算出图集对应的 Is，Io，Ip，Iss，Ioo，Iso。以上述的三元组为例，计算 Ip，Io，Is，Iss，Ioo。

Ip＝{(subClassOf，11100…0…)，(clothing，00010…0…)，(bgMusic，00001…0…)，(hasvideos，00000…1…)，…}。从选择索引 Ip 中可以看出的信息是：以 subClassOf 为谓词的有第 1，2，3 个三元组，以 clothing 为谓词的有第 4 个三元组，以 bgMusic 为谓词的有第 5 个三元组，等等。

Io＝{(Thing，10000…0…)，(舞蹈，01000…0…)，(图片，00100…0…) … }。从 Io 中可以看出的信息是：在图集中以 Thing 为宾语的有第 1 个三元组，以舞蹈为宾语的有第 2 个三元组，以图片为宾语的有第 3 个三元组，等等。

Is＝{(舞蹈，10000…0…)，(锅庄舞，01000…0…)，(服装，00100…0…)，(昌都锅庄，00011…0…)…}，表示以舞蹈为

主语的有第1个三元组，以锅庄舞为主语的有第2个三元组，以服装为主语的有第3个三元组，以昌都锅庄为主语的有第4,5个三元组。

Iss={(1,10000…0…),(2,01000…0…),(3,00100…0…),(4,00011…0…),(5,00011…0…)…}。从 Iss 可以得到信息：第4个三元组的主语与第5个三元组的主语相同，其他的各不相同。

Iso={(1,01000…0…),(2,10000…0…)…}。从 Iso 可以得到信息：第1个三元组的主语与第2个三元组的宾语相同，其他的各不相同。

其他根据定义以此类推。

5)写入 HBase

Hadoop 下各台机器根据自己分配到的图组以及上一步计算出来的索引值，向 HBase 表中插入数据。若本体过大，可以将本体分为多个图存储。Data:graphids 中存储其他相关图的 id。

至此，本体数据存储过程执行完毕。

3 实验结果及分析

为了验证本文提出的基于 HBase 的本体存储模型 HBase-OntSM 的优势，以及本文定义的存储格式和索引结构所带来的本体数据存储性能的提升，设计如下实验。

3.1 实验环境

本文所做实验使用了7台机器。其配置如下：硬件为2.5GHz Intel 酷睿 i5 处理器，4GB DDR3L 内存，1TB 5400 转硬盘驱动器。软件为 Window7 操作系统，其中5台安装有 VMware 虚拟机和 Ubuntu 11.1 32 位操作系统，另外2台安装 Cygwin 在 Windows 操作系统下模拟 Linux 环境。Hadoop 用1.0.1版本，HBase 用0.96版，安装 Oracle JDK 1.7。

3.2 数据生成与解析存库

UBA 是 Lehigh University Benchmark (LUBM)[9] 项目的数据生成器，它可以生成以一所以大学为单位，满足 LUBM 标准的本体数据，可生成 RDF 或 DAML+OIL 的数据。这些数据是可重复和可定制的，以允许用户为随机数指定种子数据。

为更好地测试提出的存储模型，依次用不同大小的数据集来进行实验。在 UBA 中，通过控制参数可以生成不同大小的数据集。本文涉及到的数据集大小如表2所列。

表2 UBA 产生的数据集

数据集	学校数	RDF 三元组数	大小
D1	1	151405	10.7M
D2	50	10108912	537.6M
D3	100	20347317	1.05G
D4	300	60550872	3.1G
D5	500	101304241	5.24G

得到足够的本体文件数据后，利用 Jena 解析这些本体数据，并将其存入 HBase 数据库中。

3.3 实验结果及分析

本文根据 LUBM 提供的14组查询，设计了5组查询对存储在 HBase 中的数据进行查询，并与使用 Jena 查询进行效率上的对比实验，结果如图3所示。图3中用 HOS 表示 HBase-OntSM。5组查询条件如表3所列。

表3 5组查询条件

查询序号	查询条件
S1	SELECT * WHERE {?A?B?C}
S2	SELECT ?A WHERE {?X rdf:type ub:UndergraduateStudent}
S3	SELECT ?A WHERE {?A rdf:type ub:GraduateStudent. ?A ub:takesCourse〈http://www.Department1.University1.edu/GraduateCourse1〉}
S4	SELECT ?A WHERE {?A rdf:type ub:Publication. ?A ub:publicationAuthor〈http://www.Department1.University1.edu/AssistantProfessor1〉}
S5	SELECT ?A,?B,?C WHERE {?A rdf:type ub:GraduateStudent. ?B rdf:type ub:University. ?C rdf:type ub:Department. ?A ub:memberOf?C. ?C ub:subOrganizationOf?B. ?A ub:undergraduateDegreeFrom?B}

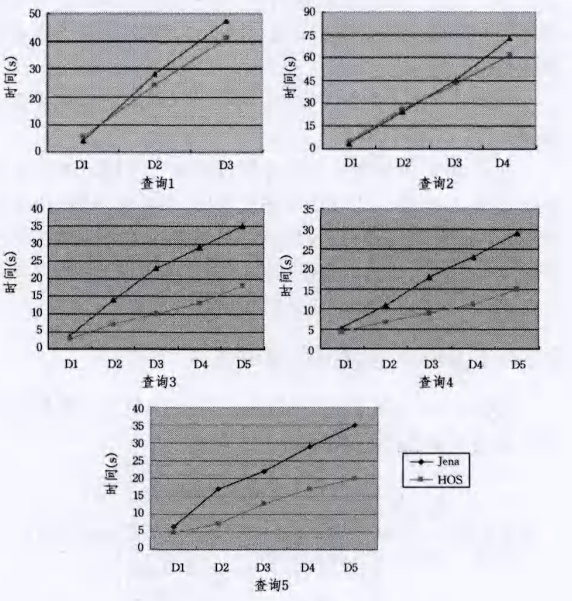

图3 查询处理结果

结果分析如下：

1)查询1和查询2是基于一个三元组模型的查询，查询返回的结果集比其他查询产生的结果集大得多，查询1对D4和D5、查询2对D5的查询因返回数据量过大而失败。针对这两个查询，Jena 和本文提出的 HOS 存储查询方法在效率上相差无几。

2)查询3和查询4是包含了两个三元组查询模型的类似查询，在本组查询中，查询3返回的数据是查询4返回的数据的两倍多，但可以看到它们所用时间相差不多。由图可知，HOS 的查询效率高于 Jena 的查询效率，而且随着数据集的增大，HOS 查询的时间并没有显著增加，查询曲线较 Jena 平缓，说明 HOS 查询具有良好的纵向扩展性。

3)查询5的 BGP 中包含了多个三元组查询模型，相比前4个查询复杂了很多，但从结果上看，复杂的查询并没有使 HOS 查询消耗更多的时间，而对 Jena 来说，查询时间增加速度较快。

由此可见，本文提出的查询算法要优于 Jena 查询，而且无论是数据量的增长还是三元组查询模式的增长，该查询算法都有很强的扩展性。

结束语 本文对目前本体的存储方式进行了分析研究，

特别是对利用非关系数据库 HBase 的存储方式进行了深入研究，提出了基于 HBase 的本体存储模型 HBase-OntSM，其用一种稀疏多维排序的方式将数据存储到 HBase 数据库中，以少量的存储空间换取高效的查询效率；以西藏文化本体为例解释该存储模型及其存储过程；最后通过设计实验与其他模型进行对比来突出本文所提模型的优势，特别是对于大规模的本体，其可以将它分为多个图进行存储，这更能体现出性能优势。

参 考 文 献

[1] Lv Yan-hui. Storage of Fuzzy Ontologies Based on Relational Databases[J]. Computer Science, 2011, 38(6): 217-222, 245(in Chinese)
吕艳辉. 基于关系数据库的模糊本体的存储方法[J]. 计算机科学, 2011, 38(6): 217-222, 245

[2] Abraham J, Brazier P, Chebotko A, et al. Distributed storage and querying techniques for a Semantic Web of scientific workflow provenance[C]// 2010 IEEE International Conference on Services Computing (SCC). IEEE, 2010: 178-185

[3] Chebotko A, Abraham J, Brazier P. Storing, Indexing and Querying Large Provenance Data Sets as RDF Graphs in Apache Hbase[C]// 2013 IEEE Ninth World Congress on Services. 2013: 1-8

[4] Sun J, Jin Q. Scalable rdf store based on hbase and mapreduce [C]// 2010 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE). IEEE, 2010: V1-633-V1-636

[5] Papailiou N, Konstantinou I, Tsoumakos D, et al. H2RDF: adaptive query processing on RDF data in the cloud[C]// Proceedings of the 21st International Conference Companion on World Wide Web. ACM, 2012: 397-400

[6] Qiang Yan, Lu Jun-zuo, Liu Tao, et al. HBase Based Parallel BFS Method[J]. Computer Science, 2013, 40(3): 228-231(in Chinese)
强彦, 卢军佐, 刘涛, 等. 基于 HBase 的并行 BFS 方法[J]. 计算机科学, 2013, 40(3): 228-231

[7] protege[EB/OL]. http://protege.stanford.edu

[8] Khadilkar V, Kantarcioglu M, Thuraisingham B. Jena-HBase: A Distributed, Scalable and Efficient RDF Triple Store[C]// CEUR Workshop Proceedings. 2012, 914: 85-88

[9] Guo Y, Pan Z, Heflin J. LUBM: A benchmark for OWL knowledge base systems[J]. Web Semantics: Science, Services and Agents on the World Wide Web, 2005, 3(2): 158-182

(上接第 16 页)

[37] Zhang Z, Deriche R, Faugeras O, et al. A robust technique for matching two uncalibrated images throuth the recovery of the unknown epipolar geometry[J]. International Journal of Artificial Intelligence, 1995, 78(1/2): 87-119

[38] Yu Q X. Research on mobile localization techniques for wheeled restaurant service robots[D]. Shanghai: Shanghai Jiaotong University, 2013(in Chinese)
于清晓. 轮式餐厅服务机器人移动定位技术研究[D]. 上海: 上海交通大学, 2013

[39] Kanade T, Kano H, Kimura S, et al. Development of a video-rate stereo machine[C]// 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems 95. IEEE, 1995, 3: 95-100

[40] Jiang H Y, Peng Y S, Ying Y B. Binocular stereo vision applied to harvesting robots[J]. Journal of Jiangsu University (Natural Science Edition), 2008, 29(5): 377-380(in Chinese)
蒋焕煜, 彭永石, 应义斌. 双目立体视觉技术在果蔬采摘机器人中的应用[J]. 江苏大学学报(自然科学版), 2008, 29(5): 377-380

[41] Li H, Chen Y L, Chang T, et al. Binocular vision positioning for robot grasping[C]// 2011 IEEE International Conference on Robotics and Biomimetics (ROBIO). IEEE, 2011: 1522-1527

[42] Guo B, Sun J, Wei Y, et al. Kinect identity: Technology and experience[J]. Computer, 2011, 44(4): 94-96

[43] Dong J M, Chen W H, Yue H S, et al. Automatic recognition and location of tomatoes based on Kinect vision system[J]. Chinese Journal of Agricultural Mechanization, 2014, 35(4): 169-173 (in Chinese)
董建民, 陈伟海, 岳昊嵩, 等. 基于 Kinect 视觉系统的西红柿自动识别与定位[J]. 中国农机化学报, 2014, 35(4): 169-173

[44] Figueroa J, Contreras L, Pacheco A, et al. Development of an Object Recognition and Location System Using the Microsoft Kinect™ Sensor[M]// RoboCup 2011: Robot Soccer World Cup XV. 2012: 440-449

[45] Zhang H, Yan R J, Zhou W S, et al. Binocular Vision Sensor (Kinect)-Based Pedestrian Following Mobile Robot[J]. Applied Mechanics and Materials, 2014, 670: 1326-1329

[46] Nakano Y, Izutsu K, Tajitsu K, et al. Kinect Positioning System (KPS) and its potential applications[C]// International Conference on Indoor Positioning and Indoor Navigation. 2012

[47] Wang S, Pan H, Zhang C, et al. RGB-D image-based detection of stairs, pedestrian crosswalks and traffic signs[J]. Journal of Visual Communication and Image Representation, 2014, 25(2): 263-272

[48] Premebida C, Carreira J, Batista J, et al. Pedestrian detection combining RGB and dense LIDAR data[C]// 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014). IEEE, 2014: 4112-4117

[49] Yagi Y, Yachida M. Real-time generation of environmental map and obstacle avoidance using omnidirectional image sensor with conic mirror[C]// IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. IEEE, 1991: 160-165

[50] Yamazawa K, Yagi Y, Yachida M. Obstacle detection with omnidirectional image sensor hyperomni vision[C]// Proceedings of 1995 IEEE International Conference on Robotics and Automation, 1995. IEEE, 1995, 1: 1062-1067

[51] Huo Z H. Omnidirectional vision based human detection and localization in complex environments[D]. Beijing: Peking University, 2010(in Chinese)
霍振华. 基于全方位视觉的复杂环境下人体目标检测与定位[D]. 北京: 北京大学, 2010