文章编号: 1000-5862(2016)05-0476-05

基于 BSON 文档树的 NoSQL 数据库与关系数据库双向映射算法研究

周　莉
(华东交通大学软件学院,江西 南昌　330013)

摘要: 分析了 BSON 文档的结构,通过比较类似结构的映射方法,给出了 BSON 文档树的概念和结构,并提出 NoSQL 数据库文档到关系数据库的映射策略,在此基础上建立了 BSON 文档模式和关系模式之间的双向映射模型,并给出了双向映射算法.

关键词: BSON 文档树; 关系数据库; 双向映射
中图分类号: TP 311　　文献标志码: A　　DOI: 10.16357/j.cnki.issn1000-5862.2016.05.06

0　引言

关系型数据库是目前使用最为广泛的数据库系统,数据库中存储的是支持关系模型的结构化数据. 然而随着互联网技术的发展,传统的关系数据库在存储和处理非结构化数据时已经暴露了效率低下、维护代价高等问题. 因此,非关系型的数据库 NoSQL 得到了非常迅速的发展.

NoSQL(Not Only SQL),是对非关系型数据库的统称. 近年来,由于其高性能、高并发和低成本等特点,在互联网行业得到了广泛的应用. NoSQL 从存储结构上可以分为 4 大类: (i) 键-值存储数据库,(ii) 列存储数据库,(iii) 文档型数据库,(iv) 图形数据库[1]. 其中文档型数据库由于其结构灵活、表示简单和高效查询等优点成为应用最为广泛的 NoSQL 数据库.

关系数据库目前仍是数据存储的主体,在构建 NoSQL 数据库时,与关系数据库之间进行数据交换和数据传输是当前数据库设计者亟待解决的问题.

BSON(Binary Serialized Document Format)是一种二进制形式的存储格式,采用了类似于 C 语言结构体的名称/对表示方法,支持内嵌的文档对象和数组对象,具有轻量性、可遍历性、高效性的特点[2],可以有效描述非结构化数据和结构化数据. 这就使得 BSON 成为理想的数据交换语言,可用于解决 NoSQL 数据库中文档结构与关系数据库的记录结构之间的双向映射问题.

1　映射方法研究

目前尚未有基于 BSON 的 NoSQL 数据库和关系数据库的映射方法. 然而,由于 BSON 与 JSON(JavaScript Object Notation)[3] 在语法上的相似性,以及 BSON 文档与 XML 文档在结构上的相似性[4],可以参考上述文档与关系数据库的映射方法. 文献[5]提出了一种 JSON 数据与关系数据库的映射方法,一种称为关系 JSON(RelationalJSON)的模式用于 JSON 文档与关系数据库的转换. 文献[6-8]提出了 XML 文档和关系数据库的映射方法,文献[6]中对一种高效的新型 XML 映射技术 s-XML 的性能展开深入探讨; 文献[7]使用 XSL 将描述数据库抽象结构的 XML 转换到各数据库支持的 SQL,进而生成具体的物理表结构; 文献[8]则提出了一种基于关系型数据的半结构化 XML 及结构.

上述方法中,JSON 数据主要用于定义通用数据交换格式,目前尚未有 NoSQL 数据库采用此种格式的数据. 另外,在 XML 文档与关系数据库的映射方法中,XML 文档的结构必须符合模型所规定的结构,例如表格模型和数据专用对象模型[9]. 而 NoSQL 数据库存储的数据多为非结构化数据,XML 文档结构无法涵盖所有数据结构.

收稿日期: 2016-01-09
基金项目: 国家自然科学基金(F030408,F020106)和华东交通大学校立科研基金(13RJ02)资助项目.
作者简介: 周　莉(1977-),女,江西南昌人,讲师,主要从事云计算搜索引擎方面的研究.

所以,针对 NoSQL 数据库,需要一个更加灵活的映射模式,使得对 NoSQL 数据结构的限制尽可能的少. 此外,映射模式还必须能较好地表达关系数据库的各种约束.

本文在文献[10]的基础上提出了基于 BSON 元素树的 NoSQL 数据库与关系数据库的双向映射算法. 首先给出 BSON 元素树的定义,并创建元素树与 NoSQL 中文档及关系数据库中关系模型之间的映射关系,在此基础之上定义相互之间转换的算法.

2　BSON 文档树

BSON 数据由文档组成,文档支持嵌套关系,可将嵌套关系看成树状结构中的父子关系. 文档由元素组成,每个元素都是一个键/值对,键不能再包含元素,值可以是基本类型数据或文档的单值或数组. 因此,BSON 文档就可以表示为树状结构.

定义 1　BSON 文档树是 BSON 文档的关系树,用于表示文档-元素结构及其文档之间的关系,各部分定义如下:(i) 最外层的文档定义为文档树的根结点;(ii) BSON 中的每个文档定义为文档树的一个结点;(iii) 每个结点包括元素列表;(iv) 元素列表中的每个元素看成一个二元组(键,值),值可以是单值或者数组,单值或数组元素有 2 种类型: 基本类型数据和文档;(v) 元素值是文档单值或数组的话,文档作为一个新的结点;(vi) 元素值是文档单值或数组的话,该元素所在结点指向新的结点;(vii) 每个结点包括双亲结点指针.

如下例 BSON 文档的关系树如图 1 所示.

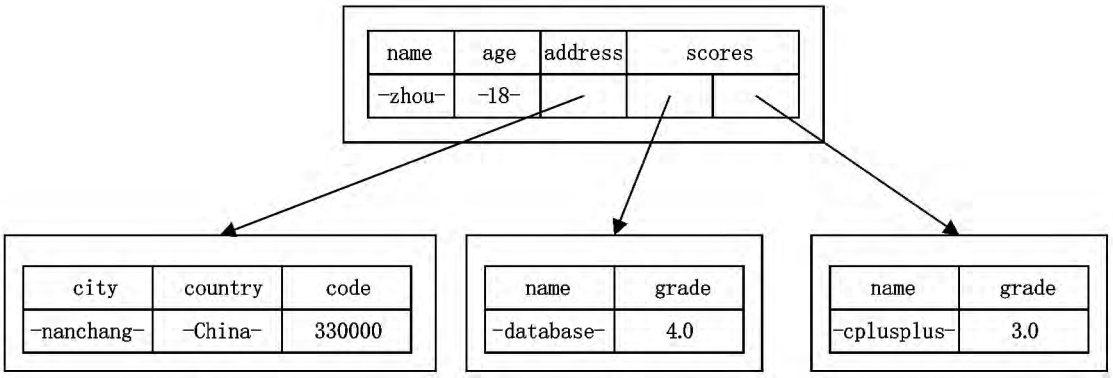

图 1　BSON 文档树

```
{
    name: "zhou",
    age: "18",
    address: {
        city: "nanchang",
        country: "China",
        code: 330000
    },
    scores: [
    {
      name: "database",
      grade: 4.0
    },
    {
      name: "cplusplus",
      grade: 3.0
    }

  ]
}
```

BSON 文档树可以表示为一个五元组 $T=(V, E, elem, parent, root)$,其中 V 是文档结点的有限集合,E 是元素的有限集合,$elem$ 是集合 V 到 E 的部分映射,满足对于任意的 $v \in V$,都有 $elem(v)=[e_1, e_2, \cdots, e_n]$,且 $e_i \in E$; $parent$ 是从 V 到 V 的部分映射; $root$ 是 V 中唯一结点,$parent(root)=\varnothing$,称为 T 的根.

为了构造 BSON,定义文档结点的逻辑结构为 (LN, EL, PT),其中 LN 为结点标识名; EL 为元素列表; PT 为双亲结点指针.

算法 1　BSON 文档树的生成算法:

输入: BSON 文档;

输出: BSON 文档树;

具体步骤:

(i) 选取 BSON 文档,表示为 D;

(ii) 创建结点 S, $S.LN=$"$root$", $S.EL=elem(D)$, $S.PT=\varnothing$, $root=S$. 将 S 加入 V 中,即 $V=V\cup\{S\}$;

(iii) 如果 $V\neq\varnothing$,则选取 V 中的一个结点 R,且 $V=V-\{R\}$; 否则算法结束;

(iv) 按照下列步骤读取并分析 $R.EL$:

(a) 如果 $R.EL=\varnothing$,则 R 为叶子结点;

(b) 如果 $R.EL$ 中所有元素的值为基本数据类型,则 R 为叶子结点;

(c) 对于 $R.EL$ 中的元素的值为文档 d,则创建一个结点 S,$S.LN$ = 元素的键,$S.EL=elem(d)$,$S.PT=R$,并将 S 加入 V 中,即 $V=V\cup\{S\}$;

(d) 如果 $R.EL$ 的元素值为数组,则依次进行判断. 若数组中包含文档值,则执行步骤(c),否则 R 为叶子结点.

(v) 转到步骤(iii).

3 BSON 文档树到关系数据库模式的映射算法

将 BSON 文档树映射成为关系数据库模式,是 NoSQL 数据库中的文档映射为关系数据库模式的关键.

算法 2 BSON 文档树到关系数据库模式的映射算法:

输入: BSON 文档树;

输出: 关系数据库模式;

具体步骤:

(i) 先序遍历 BSON 文档树中的每个结点;

(ii) BSON 文档树的根结点对应关系数据库中的一张表,称为根表;

(iii) 令树中当前访问的文档结点为 P,$P.EL\neq\varnothing$,则按照以下步骤将 P 映射为数据库中的一张表 R;

(a) $P.LN$ 作为 R 的表名;

(b) $P.EL$ 中的元素值为基本数据类型,则将元素的键作为 R 的一个字段;

(c) $P.EL$ 中的元素值为数组,则数组作为 R 的多值属性,按照弱实体集建模和设计方法[11],映射为一张表 Q;

(d) 如果 $P.PT\neq\varnothing$,则 $P.PT.LN$ 作为 R 的一个外键.

4 双向映射策略

按照算法 1 和算法 2,可以将 BSON 文档分为 2 步映射成为关系数据库模式. BSON 文档是树形结构,对象模型也是树形结构[11],所以这种映射是一种基于对象模型驱动的映射,或者是基于文档树驱动的映射. 此种映射是可逆的,即 NoSQL 数据库的 BSON 文档结构与关系数据库模式之间的映射是双向映射.

映射策略首先把 BSON 文档映射成一个对象模式,接着把对象模式映射成关系数据库模式. 或者将 2 步映射结合在一起直接从 BSON 文档到关系数据库模式的映射规则: (i) 从 BSON 映射到对象模式: 把基本数据类型映射为标量数据类型[12]; 把文档类型映射为类,文档中的每个元素的键映射为类的属性; 把嵌套文档映射为对类的对象的指针或引用. (ii) 从对象模式映射到关系数据库模式: 把指针或者引用映射为主键/外键,把标量数据类型属性映射成列; 把类映射成表.

按照上述规则,可以将关系数据库模式映射为对象模式,然后从对象模式映射为 BSON 文档. 转换架构如图 2 所示.

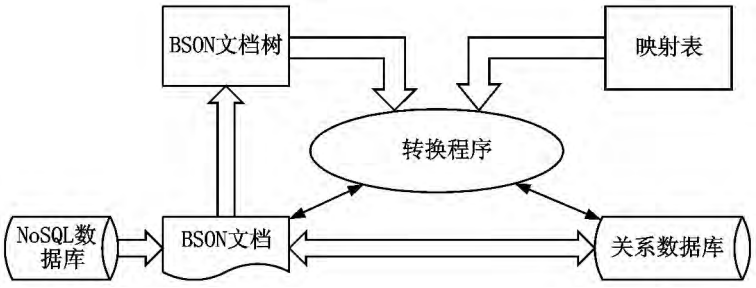

图 2 转换架构图

5 双向映射算法

双向映射算法对于 NoSQL 中的每个文档以及关系数据库中的每一个实体,在映射前后通过语义使得元素和属性得以继承. 同时,实现文档间的嵌套关系和实体的一对一、一对多和多对多关联的保持.

算法 3 从 BSON 文档生成关系数据库模式的映射算法:

输入: BSON 文档;

输出: 关系数据库模式;

具体步骤:

(i) 扫描和分析 BSON 文档,对于文档中每个元素,如果元素值为基本数据类型,则将元素的键映射成为对应类表的属性列;

(ii) 对于文档中值为基本数据类型数组的元素,生成一个带有一个外键的属性表,外键为元素的键;

(iii) 文档中元素值为嵌套文档,生成一个带有主键的类表;

(iv) 对于嵌套文档的引用生成指向上级文档类表的外键;

(v) 文档中元素值为文档数组,为数组中的子文档生成一个带有主键的类表;

(vi) 文档中元素值为文档数组,生成一个同时带有两个相关类表外键的联系表,2 个外键分别为文档元素的键和数组中子文档对应类表的主键.

算法 3 中,不仅能够将文档和元素映射为关系数据库中的实体和属性,而且将文档和元素的包含和嵌套关系映射为属性表和联系表.

算法 4　从关系数据库模式映射生成 BSON 文档模式的算法:

输入: 关系数据库模式;

输出: BSON 文档模式;

具体步骤:

(i) 为每一个表示实体的表,生成一个文档类型;

(ii) 为表中不为外键的列生成一个〈键,值〉对中的键,并添加到对应的表文档类型中;

(iii) 如果表含有一个外键,将含有外键的表称为子表,外键对应的表为父表,其转换原则是:

(a) 若父表和子表是一对一的对应关系,则可以在父表对应的文档类型中增加一项元素,键为子表的外键,值类型为与子表对应的文档类型;

(b) 若父表和子表是一对多的对应关系,则在父表对应的文档类型中增加一项元素,键为子表的外键,值类型为数组,数组类型为基本类型.

(iv) 若表含有 2 个外键,将含有外键的表称为子表,外键 1 对应的表为父表 1,外键 2 对应的表为父表 2,父表 1 与子表为一对多的对应关系,父表 2 与子表为一对一的关系,则在父表 1 对应的文档类型中增加一项元素,键为子表的外键 1,值类型为数组,数组类型为父表 2 对应的文档类型.

BSON 文档模式指的是 BSON 文档的类型,元素仅包含键,值指定类型.

算法 4 中对于含有外键的表的映射是一对多关系的映射,由于在 BSON 文档模式中不存在多对多的联系,所以在算法中无须加以考虑.

6　实验及结果分析

根据上述算法用 C++ 语言编写一个转换程序,在计算机配置为 CPU: Intel Core i5-4570 @ 3.2 GHz,内存 16 GB,操作系统为 CentOS 7.1 64 位的环境下,对 NoSQL 数据库 MongoDB [13] (版本 3.2) 到关系型数据库 MySQL(版本 5.6) 和 Oracle 11g 之间的转换时间进行测量,以关系型数据库中的记录条数为实验数据的基本单位,时间以秒为单位,取 5 次实验的平均值.

从 NoSQL 文档转换为关系数据库记录的测试结果如表 1 所示.

表 1　NoSQL 到关系数据库转换性能测试

记录条数/条	MySQL/s	Oracle/s
3 000	0.066	0.082
9 000	0.108	0.127
30 000	0.362	0.401
200 000	1.533	2.539
500 000	3.472	4.233

从关系数据库转换为 NoSQL 文档的测试结果如表 2 所示.

表 2　关系数据库到 NoSQL 转换性能测试

记录条数/条	MySQL/s	Oracle/s
3 000	0.041	0.057
9 000	0.087	0.100
30 000	0.298	0.354
200 000	1.315	2.250
500 000	3.068	4.102

测试结果显示,系统能够实现基于 BSON 文档树进行 NoSQL 和关系数据库的双向转换,且算法时间复杂度与数据规模成正比关系.

为了比较该算法与基于 XML 文档的转换算法之间的优劣,在相同的实验环境下用 C++ 基于文献 [13] 的算法编写了基于 XML 文档的转换程序,并在 MySQL 上进行实验,将结果与表 1 的结果进行比较,结果如图 3 所示.

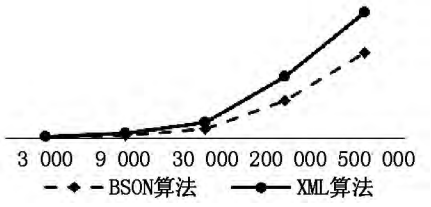

图 3　算法测试结果比较图

从图 3 可以看出,与基于 XML 的算法相比,本文算法当转换数据量逐渐增大时,由于 BSON 文档相比于 XML 文档的高效性以及 BSON 文档树在算法上的优势,在转换时间上的优势愈加明显.

7 结束语

根据 NoSQL 数据库的文档和关系数据库模式与对象模型之间的关联性,本文提出了基于对象模式的 BSON 文档树的一种 NoSQL 文档与关系数据库之间的双向映射策略,并在此策略基础上给出了双向映射模型算法. 有效地从理论层面上解决了 NoSQL 数据库文档与关系数据库之间的转换问题. 同时,由于 BSON 与 JSON 在结构上的相似性,该算法还可推广至基于 JSON 格式的异构数据库通信领域[12]. 由于 NoSQL 数据库在文档结构上的多样性,该算法还有进一步改进的必要.

8 参考文献

[1] 申德荣,于戈,王习特 等. 支持大数据管理的 NoSQL 系统研究综述 [J]. 软件学报,2013(8):1786-1803.

[2] Whittaker G. Improving performance of schemaless document storage in PostgreSQL using BSON [J]. 2013.

[3] Nolan D,Lang Duncan Temple. JavaScript,Object Notation [M]. New York: Springer,2014: 227-253.

[4] Simon M,Christian R,Peter M,et al. XML and JSON [M]. Hoboken: John Wiley & Sons,Ltd,2014: 41-78.

[5] Bharthan A,Bharathan D. Relational JSON,an enriched method to store and query JSON records [J]. International Journal of Computer Applications,2014,98(7):3-6.

[6] 尹晓奇. 关系数据库中 XML 数据存储的有效映射方案 [J]. 微型电脑应用,2014(12):61-64.

[7] 罗正伟. 一种 XML 与关系数据库的安全转换方法 [P]. CN103778147A,2014.

[8] 曹叡,吴玲达,崔亮. 基于关系数据库的半结构化 XML 数据组织技术 [J]. 计算机工程与设计,2014(10):3472-3479.

[9] 史涛,沈艳霞. XML 文档到关系型数据库的模型映射方法 [J]. 江南大学学报: 自然科学版,2015,14(5):590-595.

[10] 周莉. 基于 DTD 元素树的 XML 与 RDB 的双向映射算法研究 [J]. 江西师范大学学报: 自然科学版,2011,35(5):503-506.

[11] 周莉,王珏,周勇. 函数依赖集在属性子集上投影的新方法 [J]. 江西师范大学学报: 自然科学版,2013,37(4):387-391.

[12] 胡甜甜,曹旻. 基于本体理论的关系数据库存储模式 [J]. 计算机工程与设计,2014,35(9):3075-3079.

[13] Chodorow K. Mongo DB: The definitive guide [M]. O'Reilly Media,Inc,2013.

[14] 朱建红,陆保国. 基于对象序列化技术的数据分发系统 [J]. 网络安全技术与应用,2014(1):49-50.

[15] 耿飙,宋余庆,梁成全,等. XML 文档到关系数据库映射方法的研究 [J]. 计算机应用研究,2010,27(3):951-954.

The Research on the Bi-Directional Mapping Algorithm between NoSQL Database and RDB Based on BSON Document-Tree

ZHOU Li

(School of Software,East China Jiaotong University,Nanchang Jiangxi 330013,China)

Abstract: Analyzed BSON document's structure,the concept and structure of BSON Document-Tree is given by comparing methods of analogy,and the mapping strategy from NoSQL document database to relational database has been presented. Based on this,the bi-directional mapping model between the documents in NoSQL database and the RDB schema is built and a bi-directional algorithm is given.

Key words: BSON Document-Tree; relational database; bi-directional mapping

(责任编辑: 冉小晓)