(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104504008 B
(45)授权公告日 2018.10.02

(21)申请号 201410757171.9

(22)申请日 2014.12.10

(65)同一申请的已公布的文献号
申请公布号 CN 104504008 A

(43)申请公布日 2015.04.08

(73)专利权人 华南师范大学
地址 510631 广东省广州市天河区中山大道西55号
专利权人 广州杰赛科技股份有限公司

(72)发明人 赵淦森 李立波 林巧英 王翔 程庆年 周冠宇 高晓杰 周尚勤 王欣明

(74)专利代理机构 广州嘉权专利商标事务所有限公司 44205
代理人 谭英强

(51)Int.Cl.
G06F 17/30(2006.01)

(56)对比文件
CN 104123392 A,2014.10.29,
CN 103631907 A,2014.03.12,
CN 103810275 A,2014.05.21,
CN 102308297 A,2012.01.04,
EP 1896995 A1,2008.03.12,
胡晓鹏 等.一种基于XML映射规则的数据迁移方法设计和实现.《计算机应用》.2005,第25卷(第8期),

审查员 陈竹心

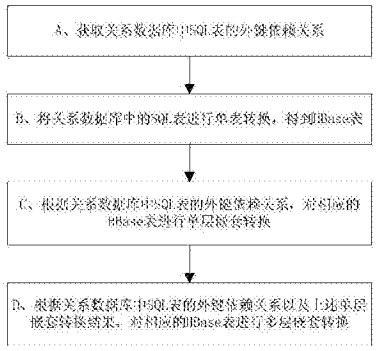
权利要求书1页 说明书3页 附图3页

(54)发明名称

一种基于嵌套的SQL到HBase的数据迁移算法

(57)摘要

本发明公开了一种基于嵌套的SQL到HBase的数据迁移算法，包括有以下步骤：A、获取关系数据库中SQL表的外键依赖关系；B、将关系数据库中的SQL表进行单表转换，得到HBase表；C、根据关系数据库中SQL表的外键依赖关系，对相应的HBase表进行单层嵌套转换；D、根据关系数据库中SQL表的外键依赖关系以及上述单层嵌套转换结果，对相应的HBase表进行多层嵌套转换。本发明方法对SQL表进行单表转换，进而根据外键依赖关系对转换成的HBase表进行单、多层嵌套转换，从而成功地在不丢失外建依赖信息的情况下实现数据迁移，数据迁移后的HBase中查询一个表即可得到结果，无需连接多个表，加快了查询效率。本发明作为一种基于嵌套的SQL到HBase的数据迁移算法可广泛应用于大数据处理领域。



CN 104504008 B

1.一种基于嵌套的SQL到HBase的数据迁移算法，其特征在于：包括有以下步骤：

A、获取关系数据库中SQL表的外键依赖关系；

B、将关系数据库中的SQL表进行单表转换，得到HBase表；

C、根据关系数据库中SQL表的外键依赖关系，对相应的HBase表进行单层嵌套转换；

D、根据关系数据库中SQL表的外键依赖关系以及上述单层嵌套转换结果，对相应的HBase表进行多层嵌套转换。

2.根据权利要求1所述的一种基于嵌套的SQL到HBase的数据迁移算法，其特征在于：所述步骤B中，优先对不存在外键依赖关系数据库中其他SQL表的SQL表进行单表转换。

3.根据权利要求1所述的一种基于嵌套的SQL到HBase的数据迁移算法，其特征在于：所述步骤C中单层嵌套转换的结果用增加HBase表中的列族的方式来表示。

4.根据权利要求1所述的一种基于嵌套的SQL到HBase的数据迁移算法，其特征在于：所述步骤D中多层嵌套转换的结果用增加HBase表中的列名前缀的方式来表示。

5.根据权利要求4所述的一种基于嵌套的SQL到HBase的数据迁移算法，其特征在于：所述HBase表中可同时包括列族和列名前缀。

一种基于嵌套的SQL到HBase的数据迁移算法

技术领域

[0001] 本发明涉及大数据处理领域，尤其是一种基于嵌套的SQL到HBase的数据迁移算法。

背景技术

[0002] 术语解释：

[0003] 1关系数据库(Relational database)：创建在关系模型基础上的数据库，借助于集合代数等数学概念和方法来处理数据库中的数据。现在主流的关系数据库有Oracle、Sql Server、MySQL等。

[0004] 2HBase：一个开源的非关系型面向列存储分布式数据库，它参考了谷歌的BigTable建模，实现的编程语言为 Java。它是Apache软件基金会的Hadoop项目的一部分，运行于HDFS(分布式文件系统)上，为 Hadoop 提供类似于BigTable规模的服务。因此，它可以容错地存储海量稀疏的数据。

[0005] 3依赖关系：表示关系数据库中表与表之间存在外键依赖关系。

[0006] 随着大数据时代的来临，关系型数据库在面临海量数据存储、查询及分析的挑战时，暴露出扩展性差、查询效率低以及难以应对高并发请求的不足。而NoSQL数据库因其不再遵从规范化设计的数据模型及有不同的底层架构设计，能很好地满足某些系统对海量数据处理的需求。目前，越来越多系统需要把数据从现有关系数据库迁移至NoSQL数据库。

[0007] HBase是目前最热门的NoSQL数据库之一，具备高扩展性、高性能、强一致性的特点。现在受到了越来越多企业青睐，并且出现了越来越多数据从关系数据库迁移到HBase的需求。而因为HBase的数据模式(即数据组织形式)与关系数据库的数据模式有巨大差异。HBase表模式并没有外键的设置，因此迁移后不能保留原关系数据库的外键依赖信息。

[0008] 现有技术中，关于关系数据库到HBase的数据迁移系统的相关工具和研究方案数量上都不多，同时，每一种工具或者方案都有其较大的不足之处，特别是每一种方案都没有很好地考虑原关系数据库中外键依赖信息的保留问题。例如Chung W C等人提出了一种利用MapReduce实现对HBase上使用SQL语句进行查询的方案。方案具体方法是将关系数据库中的表映射为HBase中同一张表的不同列族，表中的属性映射为HBase中对应的列族的列，即同一张表的数据会放置在HBase表的同一个列族。串行地排列不同表的数据，同时设置了一个额外的列族用于存放用于表示关系数据库中表之间的外键依赖关系的信息。基于这种存储，系统可以将一个SQL查询转换成一个对HBase的查询请求序列组成的MapReduce任务，在HBase上实现查询并返回结果。这种方案的确是可以完成从关系数据库到HBase的模式迁移，并对之进行查询，但是这种方案存在着两个重大的弊端。1、以串行的方式排列表与表之间的数据，导致数据矩阵稀疏。2、当同一个SQL查询涉及多个join操作，MapReudce任务将需要频繁访问HBase表查询存放外键信息的列族，效率低下。

发明内容

[0009] 为了解决上述技术问题，本发明的目的是：提供一种不丢失外键依赖信息的基于嵌套的SQL到HBase的数据迁移算法。

[0010] 本发明所采用的技术方案是：一种基于嵌套的SQL到HBase的数据迁移算法，包括有以下步骤：

[0011] A、获取关系数据库中SQL表的外键依赖关系；

[0012] B、将关系数据库中的SQL表进行单表转换，得到HBase表；

[0013] C、根据关系数据库中SQL表的外键依赖关系，对相应的HBase表进行单层嵌套转换；

[0014] D、根据关系数据库中SQL表的外键依赖关系以及上述单层嵌套转换结果，对相应的HBase表进行多层嵌套转换。

[0015] 进一步，所述步骤B中，优先对不存在外键依赖关系数据库中其他SQL表的SQL表进行单表转换。

[0016] 进一步，所述步骤C中单层嵌套转换的结果用增加HBase表中的列族的方式来表示。

[0017] 进一步，所述步骤D中多层嵌套转换的结果用增加HBase表中的列名前缀的方式来表示。

[0018] 进一步，所述HBase表中可同时包括列族和列名前缀。

[0019] 本发明的有益效果是：本发明方法对SQL表进行单表转换，进而根据外键依赖关系对转换成的HBase表进行单层嵌套转换和多层嵌套转换，从而成功地在不丢失外建依赖信息的情况下将数据从关系数据库迁移到HBase，尤其是是原本在关系数据库中存在外键依赖关系的表都整合至HBase的一张表中，数据迁移后的HBase中查询一个表即可得到结果，而无需连接多个表，加快了查询效率。

附图说明

[0020] 图1为本发明方法的步骤流程图；

[0021] 图2为数据库内各表关系示意图；

[0022] 图3为关系数据库到HBase迁移前后表状态示意图；

[0023] 图4为student 数据库图模型示意图；

[0024] 图5为单表转换示意图；

[0025] 图6为单层嵌套转换示意图；

[0026] 图7为多层嵌套转换示意图。

[0027] 具体实施方式；

[0028] 下面结合附图对本发明的具体实施方式作进一步说明：

[0029] 参照图1，一种基于嵌套的SQL到HBase的数据迁移算法，包括有以下步骤：

[0030] A、获取关系数据库中SQL表的外键依赖关系；

[0031] B、将关系数据库中的SQL表进行单表转换，得到HBase表；

[0032] C、根据关系数据库中SQL表的外键依赖关系，对相应的HBase表进行单层嵌套转换；

[0033] D、根据关系数据库中SQL表的外键依赖关系以及上述单层嵌套转换结果，对相应

的HBase表进行多层嵌套转换。

[0034] 首先，结合附图说明关系数据库中的外键依赖关系：

[0035] 关系数据库可用一个有向无环图给予描述。V表示图G中的点集，E表示图G中的边集。每一个在关系数据库中的表即为点集V中的一个点，而边表示表A有外键引用表B，这样图G可称为关系图。参照图2，点A,B,C代表表A,B,C，因为表A有外键spno引用表B，即表A依赖表B，因此有一条有向边指向表B，另外两条边同理。

[0036] 为了加快在数据迁移后在HBase中的查询效率，类似于反规范化中的增加冗余列的方式，本发明中将外键依赖关系转换为嵌套关系，在存在外键依赖关系的两个表中，让被依赖表的数据添加进依赖表中。

[0037] 参照图3，表A依赖表B，则可以称表A嵌套表B，同理，表B 嵌套表C，那么表A与表B的关系我们称为“单层嵌套”，表A与表C的关系我们称为“多层嵌套”。而因为表A同时也直接依赖表C，所以表A与表C的关系既有单层嵌套也有多层嵌套。因此，在转换后的HBase数据库中，会有相对应的三个表，HTable-A、HTable-B、HTable-C，即在原关系数据库每一个表都有相应的一个迁移后的HTable相对应。而根据它们之间的嵌套关系，HTable-A会包含HTable-B及HTable-C的信息。

[0038] 对于单层嵌套，利用增加HBase中的列族(family)的方式来表示；对于多层嵌套，利用增加HBase中的列名(qualifier)前缀的方式来表示。上述方式是的HBase表可以很好地代替SQL表，存储器本身的数据并对于每一个SQL语句，无论它包括有多少个连接条件，都只需要在钱以后的HBase中查询一个表即可得到结果，从而加快了查询效率。

[0039] 在转换完成之后，对应于迁移后的表的列族数只是比原关系数据库的表的外键数多1，这将保证迁移后的表的列族数不会过多而导致HBase的性能下降。

[0040] 参照图4的数据库模型，将其中数据库迁移到HBase，经过单表转换、单层嵌套转换和多层嵌套转换的步骤：

[0041] 参照图5单表转换示意图，将SQL表“speciality”转换成HBase表“speciality”。参照图6单层嵌套转换示意图，利用增加HBase中的列族的方式来表示。参照图7多层嵌套转换示意图，利用增加HBase中的列名前缀的方式来表示，例如：

[0042] speciality:department.dno=“5001” 。

[0043] 以上是对本发明的较佳实施进行了具体说明，但本发明创造并不限于所述实施例，熟悉本领域的技术人员在不违背本发明精神的前提下还可以作出种种的等同变换或替换，这些等同的变形或替换均包含在本申请权利要求所限定的范围内。

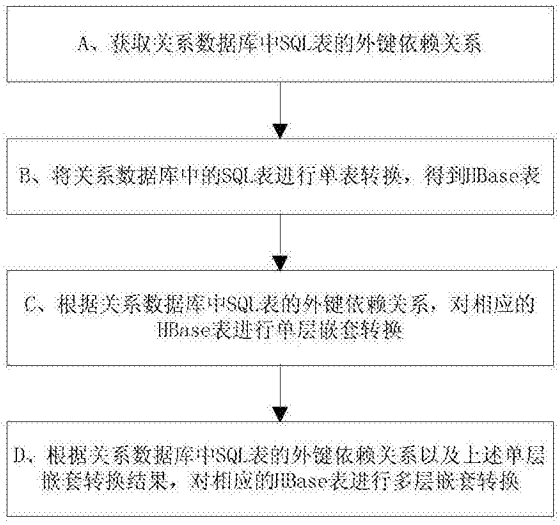

A、获取关系数据库中SQL表的外键依赖关系
B、将关系数据库中的SQL表进行单表转换，得到HBase表
C、根据关系数据库中SQL表的外键依赖关系，对相应的HBase表进行单层嵌套转换
D、根据关系数据库中SQL表的外键依赖关系以及上述单层嵌套转换结果，对相应的HBase表进行多层嵌套转换

图1

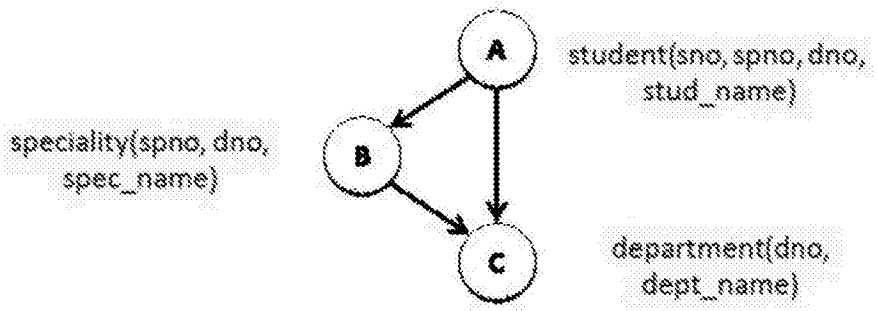

A
B
C
student(sno, spno, dno, stud_name)
speciality(spno, dno, spec_name)
department(dno, dept_name)

图2

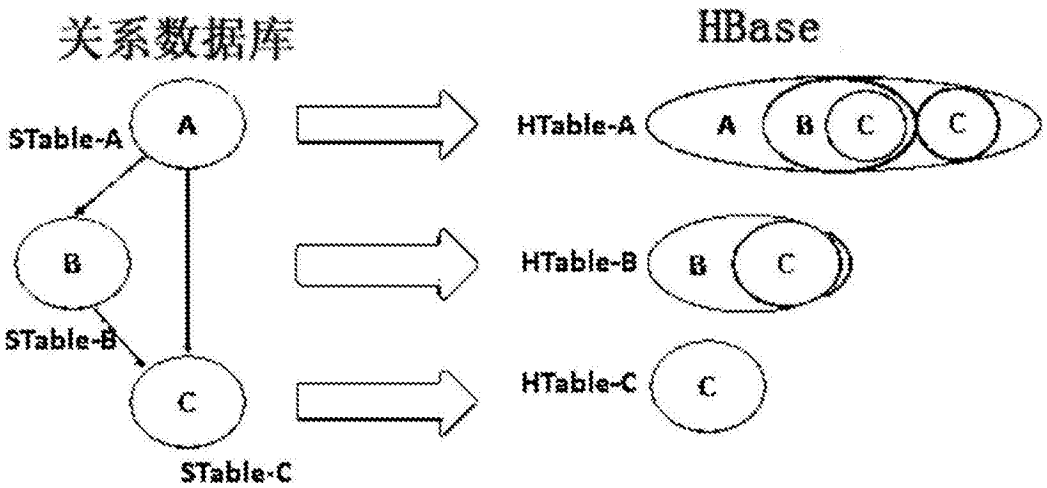

关系数据库
HBase
STable-A
A
B
STable-B
C
STable-C
HTable-A
A
B
C
C
HTable-B
B
C
HTable-C
C

图3

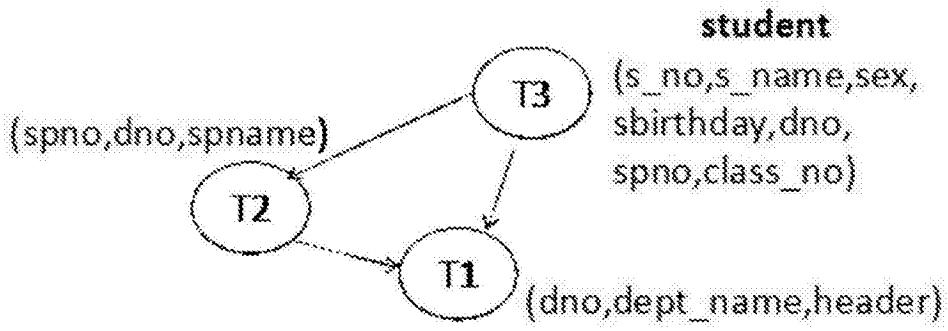


图4

SQL数据库中的表speciality

spno	dno	spname
101	5001	Software
102	5002	Math

数据迁移

HBase中的表speciality

Rowkey	Timestamp	speciality
101	t1	speciality:dno="5001",speciality:spname="Software"
102	t2	speciality.:dno="5002",speciality:spname="Math"

图5

SQL数据库中的表speciality

spno	dno	spname
101	5001	Math
102	5002	Math

HBase中的表department

Rowkey	Time stamp	department
5001	t1	department:dept_name="Computer",department:header="Alice"
5001	t2	department:dept_name="Mathematics",department:header="David"

数据迁移

HBase中的表speciality

Rowkey	Timestamp	speciality	department
101	t1	speciality:spname="Software",speciality:dno="5001"	department:dno="5001",department:dept_name="Computer",department:header="Alice"
102	t2	speciality:spname="Math",speciality:dno="5002"	department:dno="5002",department:dept_name="Mathematics",department:header="David"

图6

SQL数据库中的表student

s_no	s_name	sex	sbirthday	dno	spno	class_no
001	libo	M	19891111	5001	101	1
002	zijing	F	19910888	5001	101	2

HBase中的表speciality

Rowkey	Timestamp	speciality	department
101	t1	speciality:spname="Software",speciality:dno="5001"	department:dno="5001",department:dept_name="Computer",department:header="Alice"
102	t2	speciality:spname="Math",speciality:dno="5002"	department:dno="5002",department:dept_name="Mathematics",department:header="David"

HBase中的表department

Rowkey	Timestamp	department
5001	t1	department:dept_name="Computer",department:header="Alice"
5001	t2	department:dept_name="Mathematics",department:header="David"

数据迁移

HBase中的表student

Rowkey	Timestamp	student	department	speciality
001	t1	student:s_name="libo",student:sex="M",student:sbirthday="19891111",student:dno="5001",student:spno="101",student:class_no="1"	department:dno="5001",department:dept_name="Computer",department:header="Alice"	speciality:spno="101",speciality:dno="5001",speciality:spname="Software",speciality:department.dno="5001",speciality:department.dept_name="Computer",speciality:department.header="Alice"
002	t2	...	...	...

图7