文章编号：1005-8451（2013）10-0037-04

基于XML的分布式异构数据库变化捕捉及动态同步系统实现

赵金铃，谭献海，王亚兰，何　磊
（西南交通大学　信息科学与技术学院，成都　610031）

摘　要：针对分布式异构数据库在网络环境下变化数据的捕捉与同步更新存在的问题，本文采用基于触发器的捕捉方法实现异构数据库之间同步策略，研究并解决了一对多映射环境下的同步问题。理论分析和实验结果表明，该方法在实际应用中具有重要意义。

关键词：分布式异构数据库；同步更新；变化捕捉；一对多映射
中图分类号：U285：TP39　**文献标识码：**A

Implementation of change capture and Dynamic Synchronization System of distributed heterogeneous database based on XML

ZHAO Jinling, TAN Xianhai, WANG Yalan, HE Lei
(School of Information Science and Technology, Southwest Jiaotong University, Chengdu　610031, China)

Abstract: Aimed at the problems of capturing the change data and synchronization update for distributed heterogeneous database under the environment of network, a capturing strategy based on trigger was proposed to implement the synchronization between heterogeneous databases. The paper focused on the synchronization problem under the one-to-many mapping. Theoretical analysis and experiments showed that the method and strategy whether in theory or in practical application were with important significance.

Key words: distributed heterogeneous database; synchronization update; change capture; one-to-many mapping

随着社会信息化建设和企业的跨地域发展，网络技术和分布式技术应用越来越广泛，众多领域的应用中都涉及到数据同步问题，尤其对于分层管理模式下的企业和国家单位，各级机构之间需要及时、可靠地同步大量数据信息，从而整体提高生产效率和管理水平，增强市场竞争力和应变能力。

然而，在数据同步的过程中存在多种问题：企业和各级机关单位系统不同，数据库种类较多，存在较多的外界因素等。字段名称、字段类型、字段个数和字段顺序的冲突等。数据同步要安全、实时、高效、可靠，并且能大数据量的使用，还有一对多映射等。

本文提出了一种基于XML的“SQL还原技术”。该方法运用触发器捕获变化数据，并对触发器算法进行优化，然后将其转化为Data XML，再根据源表元数据和目标数据库表元数据建立映射文件（Mapping File），最后两者关联还原为相应的SQL语句后进行插入、删除、更新操作。

收稿日期：2013-02-27
基金项目：国家科技计划项目（2009BAG12A06）。
作者简介：赵金铃，在读硕士研究生；谭献海，副教授。

1　分布式异构数据库与XML

分布式数据库是一种既分散又集中的数据库系统。其分散体现在具体应用的环境通常是跨地域的，在管理上又是集中统一的整体。分布式数据库具有可靠性、自治性、模块性与系统升级能力、效率及可用性等优点。

XML是文档标记的一种标准，是标准通用标识语言SGML的一个优化子集，它具有可扩展、结构性、平台独立性等特点。

一对一映射是指源数据库中一张表只对应目标库中一张表，两者在结构上基本相同。

一对多映射是指源数据库中一张表对应多张

同步表。可能存在源表对应一个目标库中多张表或源表对应多个目标库中多张表。它与一对一映射的区别在于应用性更广、灵活性更强、隐藏的冲突问题更多、同步性能要求更高。

文中采用 Oracle 作为共享数据库，专业数据库有 DB2、SQL Server2000、MySql 等，专业数据库 1 与共享数据库、专业数据库 2 之间数据表有同步关系，与专业数据库 3 之间没有。若专业数据库 3 要使用专业数据库 1 中的数据，只能从共享数据库读取（本文以专业数据库 1 为例），如图 1 所示。

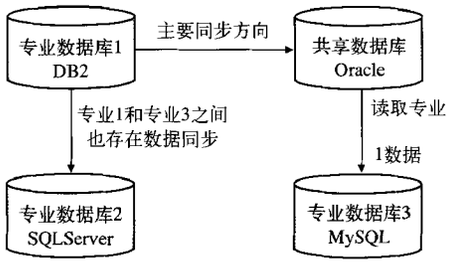

图1 分布式异构数据库同步关系

2 系统架构和同步策略

基于 XML 的“SQL 还原”技术同步更新系统架构如图 2 所示。

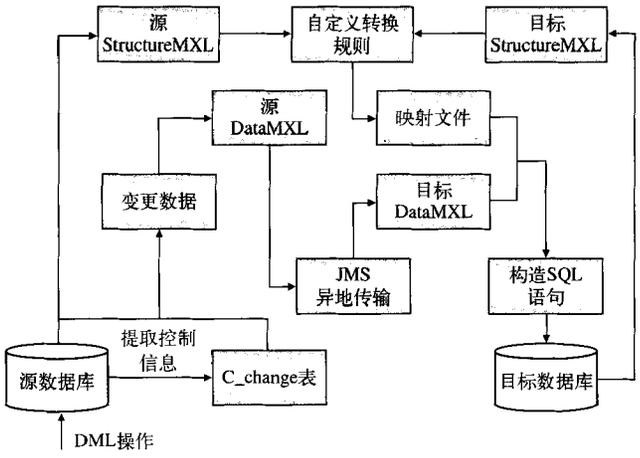

图2 同步更新系统架构

其同步策略为：

（1）在源数据库建立一张变化表（c_change），当对源表进行 DML 操作时，利用触发器提取源表控制信息插入到 c_change 中，根据变化表中控制信息和目标表关联将变化数据转化为 DataXML；

（2）提取源表数据和目标表数据建立映射文件（Mapping File），映射文件需要有层次和规律性，并尽可能整理出存在的冲突情况；

（3）利用 Dom4j 解析 DataXML，根据源表信息在映射文件中找到目标表信息，并判断映射是否存在一对多情况，消除存在的冲突问题，最终还原 SQL 语句；

（4）数据同步成功后，删掉源 c_change 表中的对应信息记录，若由于网络或者其他未知原因造成的中断，则停止当前变更记录的所有操作，数据库回滚，等待下一次轮回同步操作。

3 关键技术实现

3.1 变化数据提取

不同数据库产品支持不同的变化捕捉方法。如快照法、触发器法、日志法、API 法、影子法、时间戳法、控制表法。这些方法虽各有优点，但同时也存在一定的局限性。本文选用触发器提取源表中的数据，是因为通过触发器可以直接获取净变化数据且操作简单，而触发器在大多数数据库中都存在，使用效率较高。

为了便于提取数据，在源数据库中创建 c_change 表，用来存放源表的控制信息。以 SQL Server 为例，c_change 表的字段结构如表 1 所示。

其中字段 UPDATETYPE 的值为：I、D、(U/D、U/I)。“I”代表对源表在进行 Insert 操作，“D”为 Delete 操作，U/D 代表 Update 前的值，U/I 代表 Update 后的值。例如：现有一张坡度表，其字段信息如表 2 所示。

当对坡度表进行 Insert 操作后，通过触发器

表1 c_change表结构及说明

字段名称	字段类型	字段说明	备注
NUMBER	INT(4)	主键，记录同步个数	4 个字节的整形
S_PRIMARYKEY	VARCHAR(50)	源表主键名称	
S_PRIMARYKEYVALUE	VARCHAR(50)	源表主键值	
UPDATETYPE	VARCHAR(3)	变更类型	I、D、(U/D、U/I)
S_TABLE	VARCHAR(20)	源表名称	
S_DBNAME	VARCHAR(15)	源数据库类型	如：DB2 下的一个库 TEST
S_DBTYPE	VARCHAR(20)	源数据库名称	如：DB2、SQL Server
OPERATIONTIME	DATETIME	当前操作时间	

表2　坡度表

字段名称	行别	起点里程	终点里程	起轨顶标高	坡度	坡长	线路编号
备注	主键		主键		主键		主键

提取控制信息放入到 c_change 中，观察其数据如表 3 所示。

表3　c_change表数据

NUMBER	PRIMARYKEY	PRIMARYKEYVALUE	UPDATETYPE	S_TABLE	S_DBNAME	S_DBTYPE	OPERATIONTIME
3125	行别 / 终点里程 / 坡度 / 线路编号	上 /2345/3/2	I	坡度表	SQL Server	Test	2013-1-12 10:45:12

联立表 2、表 3，通过如下 SQL 语句：

```
Select a.* from 坡度表 a left join c_change b on 行别 = '上'
and 终点里程 =2345 and 坡度 =3 and 线路编号 = 2
```

就可以从坡度表中提取刚插入的数据，再通过 Dom4j 转化为 DataXML 如图 3 所示。

```
<?xml version="1.0" encoding="UTF-8"?>
<DataXML>
    //编号是在c_change中的位置，即当前Number的值
    <Row 编号="3125">
        <行别>上</行别>
        <起点里程>1234.0</起点里程>
        <终点里程>2345.0</终点里程>
        <起轨顶标高>3.0</起轨顶标高>
        <坡度>3.0</坡度>
        <坡长>2.5</坡长>
        <线路编号>2</线路编号>
        <UpdateType>I</UpdateType>
        <SourceTable>坡度表</SourceTable>
        <SourceDbName>Test</SourceDbName>
        <SourceDbType>SQLServer2000</SourceDbType>
        <primarykeyvalues>上 /2345/3/2</primarykeyvalues>
    </Row>
</DataXML>
```

图3　DataXML示意图

3.2　触发器算法

数据同步时，在源表中创建触发器，以便用户将更新的信息记录到变化表中。若在书写的同时对其性能进行优化，可以提高同步效率。其算法分析如下：

A1：对源表进行 Insert 操作

B1：提取控制信息插入到 c_change 中，UPDATETYPE 的值设置为“I”；

A2：对源表进行 Delete 操作

B1：若 c_change 中已经存在该条信息的 Insert 操作，删除该条记录，提取的控制信息不再插入；

B2：若 c_change 中已经存在该条记录的 Update 操作，提取的控制信息插入到 c_change 表中，UPDATETYPE 设置为“D”，S_PRIMARYKEYVALUE 的值为两条更新记录里面第 1 条（UPDATETYPE 为 U/D）S_PRIMARYKEYVALUE 的值，并删除已存在的两条更新记录；

B3：若 c_change 中不存在该条记录的信息，则提取控制信息插入到 c_change 中，UPDATETYPE 的值为“D”；

A3：对源表进行 Update 操作

B1：若 c_change 中已经存在该信息的 Insert 操作，则将该条记录的 S_PRIMARYKEYVALUE 更改为当前值，UPDATETYPE 更改为“I”，提取的控制信息不再插入；

B2：若 c_change 中已经存在该条记录的 Update 操作，则把第 1 条记录（UPDATETYPE 为 U/D）中 S_PRIMARYKEYVALUE 的值更改为第 2 天记录（UPDATETYPE 为 U/I）中 S_PRIMARYKEYVALUE 的值，更改第 2 条记录中 S_PRIMARYKEYVALUE 的值为当前值，提取的控制信息不再插入；

B3: 若 c_change 中不存在该条记录的信息，则插入两条 UPDATE 记录，第 1 条 UPDATETYPE 的值为“U/D”，第 2 条 UPDATETYPE 的值为“U/I”。

通过对触发器的优化处理，保证数据的变化更改在源数据库端解决，以提高数据同步的高效性。

3.3　一对多映射处理

一对多映射是指源数据库表对应多张同步表。在解析 DataXML 时，通过源表信息在映射文件中查找目标表的信息，若源表和目标表之间存在字段个数、字段顺序、字段类型、字段名称等冲突问题，这时一般通过映射文件的匹配和主键一致性来解决。映射文件如图 4 所示。

由图 4 可知，坡度表存在一对多映射，第 1 映射为专业 1 与共享数据库之间的字段映射，即主要映射；第 2 映射是专业 1 与专业 3 之间的映射，为次要映射。映射文件主要包含有目标表信息和字段属性。

4　试验验证

本文以 Java 作为开发工具实现同步系统，通

```
<MappingFileXml>
    <SourceDB 源数据库类型="SQLServer2000">
        <SourceDBName 源数据库名="Test">
            <SourceTable 源数据库表="坡度表"> //该表存在一对多映射
                <SourceFieldName 字段名称="行别" 字段类型="char">
                    <MappingRelation> //第一映射，也是主要映射
                    <DestinationDB>Oracle</DestinationDB>
                        <DestinationDBName>nlabdc1</DestinationDBName>
                        <DestinationTable>DS_ROUTE_TESTTABLE</DestinationTable>
                        <DestinationField>行别</DestinationField>
                        <DestinationFieldType>CHAR</DestinationFieldType>
                        <IsPrimaryKey>Yes</IsPrimaryKey>
                        <Isnull>0</Isnull>
                        <Fieldlong>2</Fieldlong>
                        <Fieldscale>0</Fieldscale>
                    </MappingRelation>
                    <MappingRelation> //第二映射，次要映射
                        <DestinationDB>DB2</DestinationDB>
                        ..........
                    </MappingRelation>
                </SourceFieldName>
                <SourceFieldName 字段名称="起点里程" 字段类型="float">
                    <MappingRelation>
                        ..........
                    </MappingRelation>
                        ..........
```

图4 映射文件

过 JDBC 访问数据库。系统分为 2 个独立的部分进行：

（1）以 Oracle 为共享数据库，SQL Server、DB2 分别为专业数据库进行一对一同步测试；

（2）再以 DB2 为专业数据库 1，SQL Server 为专业数据库 2，Oracle 为共享数据库，进行 DB2 到 SQL Server 和 Oracle 的一对多同步测试。在一对多测试中模拟字段名称、字段顺序、字段个数、字段类型等冲突问题。同步用时和同步数据项数目之间的关系如图 5 所示。

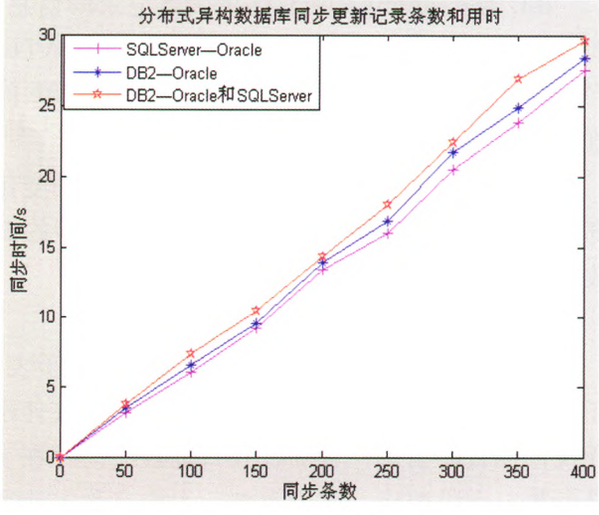

图5 同步数据项关系

从图 5 可以看出两者基本呈线性关系，这说明实现方案是可行的。因为数据库的不同，异构之间同步有一定的时间差异。其次，方案在设计方面存在灵活性、可靠性，同步之前对每一条数据都需要进行位置判断，确保其同步的安全性。如果数据同步成功，变化表 c_change 中对应的记录也同时删除，否则反之。因为每条数据同步都需要重新建立数据库连接，判断字段的数据类型，删除 c_change 表中对应的记录，频繁地做上述操作是比较耗时的，对系统的运行效率有一定的影响。

5 结束语

本文针对分布式异构数据库同步中的数据变化捕捉方法和同步策略关键技术进行了研究，在保证用户自治性前提下，设计了触发器捕捉方法，在解决一对一同步的基础上，考虑实际应用并完成了一对多映射和冲突问题等。本文在方案设计上具有如下优点：

（1）变化表捕捉方法不更改源表结构，能适用于所有支持触发器的异构数据库；

（2）变化表中只记录变更控制信息，节省了数据库存储空间；

（3）对触发器的算法进行了优化，确保数据同步前的唯一性；

（4）以 XML 作为中间件，根据 DataXML 在映射文件中查找目标表信息，保证了平台的可扩展性；

（5）若数据同步成功则清除变化表中的记录，否则反之；

（6）可以实现同库中一表对多表，异构中一表对多表的数据同步等。

参考文献：

[1] 杨 鹏，杨海涛，王正华．异构数据库变化捕捉及同步策略 [J]. 计算机工程，2008（16）：53-55.

[2] 刘永毅．异构分布式数据库远程数据同步的研究和设计 [D]. 长春：吉林大学，2010.

[3] 李 佩. 基于XML的异构数据库同步技术研究 [D]. 山东：曲阜师范大学，2010.

[4] 时俊苓，叶 丹．一个数据同步系统的设计及实现 [J]. 计算机系统应用，2008（9）：12-14.

[5] 沈 敏，许华虎，季永华．基于 XML 的分布式异构数据库数据同步系统的研究 [J]. 计算机工程与运用，2005 (5)：184-186.

[6] 熊 现，邱卫东，陈克非．基于 JAVA/XML 的分布式异构数据库同步系统的研究 [J]. 计算机应用与软件，2008，25（2）：121-123.

[7] 张月琴，袁新坤．一个数据同步系统的设计与实现 [J]. 微计算机信息，2008（24）：182-184.

[8] 王玉标，饶锡如，何 盼．异构环境下数据库增量同步更新机制 [J]. 计算机工程与设计，2011（11）：948-951.

责任编辑 陈 蓉