文章编号：2096-1472(2021)-01-06-04
DOI:10.19644/j.cnki.issn2096-1472.2021.01.002

异构数据库同步技术的研究与实现

张记强，王 仁，蒋欣欣，李明磊

(中国航天科工集团第二研究院706所，北京 100854)
✉zhangjiqiang0517@163.com; thank816you@163.com;
jiangxinxin958@163.com; liminglei_casic@hotmail.com

摘 要：随着信息化应用的不断深入，企业内部不同应用系统、企业与外部信息系统数据同步的需求日益强烈。由于各信息系统使用的数据库管理软件各不相同，导致数据之间无法实现同步与共享，因此，需要设计一种异构数据库之间同步的方法，解决不同信息系统数据间的同步问题。本论文在现有成熟数据库产品的基础上，分析了数据同步过程中的关键问题，如数据类型差异、数据捕获策略、同步振荡等。针对这些问题，设计采用建立映射模式的方法、触发器与控制表相结合的方法以及在同步任务中对控制表进行干预的方法予以解决。最后，在QT(跨平台软件开发框架)下，设计实现异构数据库同步系统，实验验证表明，本系统达到了较好的数据同步效果。

关键词：异构数据库；数据同步；数据捕获；触发器；QT
中图分类号：TP311.13 **文献标识码**：A

Research and Implementation of Synchronization Technology of Heterogeneous Database

ZHANG Jiqiang, WANG Ren, JIANG Xinxin, LI Minglei

(*Institute 706*, *Second Academy of China Aerospace Science and Industry Corporation*, *Beijing* 100854, *China*)
✉zhangjiqiang0517@163.com; thank816you@163.com;
jiangxinxin958@163.com; liminglei_casic@hotmail.com

Abstract: With the development of information applications, there is a growing demand for data synchronization between different application systems within enterprises, between enterprises, and amongst external information systems. However, due to different database management software used by different information systems, data synchronization and sharing cannot be achieved. Therefore, it is necessary to design a method of synchronization between heterogeneous databases to realize data synchronization of different information systems. Based on existing mature database products, this paper analyzes key issues in the process of data synchronization, such as data type difference, data capture strategy, synchronous oscillation and so on. In order to solve these problems, methods of establishing mapping pattern, combining trigger with control table, and intervening control table in synchronization task are designed. At last, under the framework of QT (the cross-platform software development framework), a heterogeneous database synchronization system is designed and implemented. Experimental results show that the system achieves better data synchronization effect.

Keywords: heterogeneous database; data synchronization; data capture; trigger; QT

1 引言(Introduction)

在数据库管理系统领域，已经存在多个成熟的数据库产品，而在企业搭建的众多应用系统中，由于应用开发部门不同或者应用系统本身特点等因素，使得不同的系统使用的数据库系统也不同。随着业务与用户规模的不断增加，传统的单一数据中心已经不能满足企业的发展需求，跨地域的互联数据中心建设方案由此产生，例如两地三中心、异地多活数据库等方案[1]，这种场景就涉及异构数据库之间数据同步的问

基金项目：“十三五”全军共用信息系统装备专用技术预先研究项目(31511090104).

题。同时以神通、武汉达梦、人大金仓为代表的一大批优秀的国产数据库软件逐渐被市场认可，并被企业应用到内部系统中，用于替换国外数据库产品。

现有技术中，异构数据库数据同步方式大多是基于自身的同步复制技术实现的，各数据库厂商在各自数据库复制的基础上提出了数据库的同步方案。例如SQL Server提出的出版者及订阅者同步方案，用于多台数据库服务器之间的数据同步[2,3]；Oracle采用多主复制和物化视图的方案实现数据同步功能。

本文通过对异构数据库数据同步问题进行分析与研究，提出了解决异构数据库数据同步关键问题的方法。在该方法的基础上，设计并实现了基于QT框架的异构数据库同步系统，实现异构数据库间数据的同步与共享。

2 相关工作(Related work)

2.1 天熠嵌入式数据库简介

天熠嵌入式数据库(TYCHE)是中国航天科工集团第二研究院706所面向综合化、网络化、信息化应用需求自主研发的一款嵌入式实时数据库产品，具备强实时、高可靠的产品特性，目前在项目中应用广泛。其特点是占用资源少，并支持主流的操作系统，尤其是它的处理速度非常快，单条数据读取速率能达到40微秒。本项目使用天熠数据库作为数据存储的软件。

2.2 QT开发工具

QT是由Qt Company开发的跨平台C++图形用户界面应用程序开发框架。本文选择QT作为开发工具主要是由于在其上开发的软件具有很好的跨平台性，即一次开发可以在微小改动的情况下，实现在Windows和中标麒麟下的编译运行[4]。

2.3 相关研究

目前已经出现了一些异构数据库数据同步技术，但总的来说，这些技术与各数据库厂商产品紧密绑定，扩展性较差。

文献[3]研究的是分布式异构数据库环境下数据同步技术，由系统管理员启动同步程序，从而将Linux服务器下的Oracle数据库中的数据单向同步到Windows服务器下的Oracle数据库中。该技术实现了Oracle到Oracle之间的数据同步，并且只能单向同步，底层借助的是Oracle的SQL Plus工具。该文献涉及的方法与Oracle数据库紧密绑定，不能解决其他异构数据库之间数据同步的问题。

3 数据同步常见问题(Data synchronization FAQ)

3.1 数据类型差异

对于异构数据库的数据同步，必须解决异构数据库数据类型差异的问题[5]。数据类型差异表现在以下四个方面。

(1)异构数据库支持的数据类型种类不完全相同。例如A数据库有时间类型，B数据库不一定有时间类型。

(2)异构数据库相同数据类型的名称可能不同。例如存储整型数据，A数据库类型名称为integer，B数据库类型名称为int。

(3)在不同场景下，一个数据库的同一数据类型对应其他数据库的数据类型不同。例如A数据库的varchar类型根据实际数据长度，可能对应B数据库的varchar或blob数据类型。

(4)类型映射往往为单向映射。例如A数据库支持money类型，B数据库不支持money类型但是支持double类型。此时A数据库的money类型对应B数据库的double类型，但是B数据库的double类型一般不能直接转换为A数据库的money类型。

本系统解决异构数据库数据类型差异的方案是通过建立数据类型映射，用户可根据需要自定义或者修改数据类型映射规则。

3.2 数据捕获策略

不同数据库产品支持不同的数据捕获策略，常见的有快照法、触发器法、日志法、API法、影子表法、控制表法[6]。这些方法各有优缺点，若本系统针对每个数据库采用其独特的数据捕获方式，会对系统的通用性产生一定的影响。

考虑到当今主流数据库都已经实现触发器机制，并且用户对触发器的接受程度较高，本系统提出一种基于触发器的数据捕获方法，并在触发器机制上进行了改进，增加了控制表机制。

3.3 同步振荡问题

同步振荡是由于在异构数据库两端同时建立触发器导致的一种循环更新的问题。比较典型的场景是用户操作A数据库，在A数据库端通过触发器产生一条数据变更记录写入控制表，系统将数据同步到B数据库，B数据库也会通过触发器产生一条数据变更写入控制表。若系统将B数据库变化回写到A数据库，此时就会形成一种循环更新的现象。

本系统解决同步振荡问题的方案是在同步过程对控制表进行干预。

4 异构数据库同步系统的设计与实现(The design and implementation of Heterogeneous database synchronization system)

4.1 系统总体架构

图1为异构数据库同步系统架构图。异构数据库数据同步过程由建立触发器、创建控制表、数据源注册、建立模式映射、建立同步任务五部分功能组成。其中建立触发器与创建控制表功能位于异构数据库管理系统端；数据源注册、建立模式映射、建立同步任务功能位于数据同步服务器端。

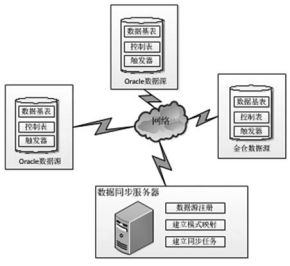

图1 异构数据库同步系统架构图

Fig.1 The architecture of heterogeneous database synchronization system

异构数据库端完成以下工作：

(1)创建控制表。为完整记录数据变化过程，为每一个数据库中的数据基表创建一个对应的控制表[7](CTRLTABLE)。控制表包含数据基表的主键，同时还包括其他字段记录数据源的变化过程。控制表必须在同步的两端均存在。控制表包含字段如表1所示。

表1 控制表结构

Tab.1 The structure of CTRLTABLE

字段说明	字段名	类型
控制表主键	Id	整型
变更数据所在的表名	TableName	字符串
变更数据的主键字段名称	FiledName	字符串
基表主键值	FiledKey	整型
变更类型	OperateType	字符串
同步标识(0为未同步，1为已同步)	IsSync	字符串
变化时间戳	UpdateTime	时间类型

(2)建立触发器。触发器的作用是在数据基表发生变化时，捕获变化的数据。在本系统中建立的触发事件包括INSERT、UPDATE与DELETE[8,9]。

如图2所示为数据基表的一个示例，该表名称为testTable。结合表1中控制表结构，本文设计INSERT触发器结构如图3所示，UPDATE触发器结构如图4所示，DELETE触发器结构如图5所示。

```
CREATE TABLE testTable (
        smallmoney smallmoney    NULL,
        RoleFlags int NULL primary key)
```

图2 数据基表testTable示例

Fig.2 The example of testTable

```
CREATE TRIGGER tigger-CtrlTableinsert
   ON   testTable   FOR   INSERT   AS
   BEGIN
   INSERT   INTO   CTRLTABLE
   (TableName,FiledName,FiledKey,OperateTyp
e,IsSync,UpdateTime)
   select
   'testTable','RoleFlags',inserted.[ROLEFLAGS
   ],'insert','0',GETDATE()     from inserted
   END
```

图3 INSERT触发器示例

Fig.3 The example of insert trigger

```
CREATE TRIGGER tigger_update
   ON   testTable   FOR   UPDATE   AS
   BEGIN
   INSERT          INTO          CTRLTABLE
   (TableName,FiledName,FiledKey,OperateTyp
   e,IsSync,UpdateTime)
   select
   'testTable','RoleFlags',inserted.[ROLEFLAGS
   ],'update','0',GETDATE()   from inserted
   END
```

图4 UPDATE触发器示例

Fig.4 The example of update trigger

```
CREATE TRIGGER tigger_ctrlTabledelete
   ON testTable     FOR DELETE     AS
   BEGIN
   INSERT   INTO   CTRLTABLE
   (TableName,FiledName,FiledKey,OperateTyp
   e,IsSync,UpdateTime)
   select
   'testTable','RoleFlags',deleted.[ROLEFLAGS]
   ,'delete','0',GETDATE()   from deleted
   END
```

图5 DELETE触发器示例

Fig.5 The example of delete trigger

数据同步服务器端完成以下工作：

(1)数据源注册。将异构数据库作为数据源接入系统中，其他功能模块均依赖本功能，未注册到系统的数据源不会作为数据同步的源数据库或者目标数据库。

(2)建立模式映射。为屏蔽各数据源的差异，为异构数据源建立模式映射。如图6所示，系统提前将Oracle与达梦的元数据进行获取并保存在本地TYCHE中，在数据同步时使用建立的模式映射关系屏蔽异构数据库的差异。

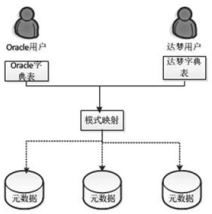

图6 模式映射图

Fig.6 The structure of pattern map

(3)创建同步任务。用于管理整个数据同步的过程，并且解决同步振荡问题。

4.2 关键模块的实现

4.2.1 数据源注册

注册数据源分为两步，第一步通过加载数据库提供的ODBC驱动连接数据库；第二步将注册信息写入TYCHE中。数据源注册需要的信息包括数据源类型、注册名称、IP地址、端口号、DNS信息、默认连接数据库名称、用户名与密码。图7为保存数据源注册信息的表结构图，表名称为“数据源表”。

图7 数据源表存储结构

Fig.7 The storage structure of data source table

通过以上两步，即可将数据库接入系统中，并且与数据源建立连接。

4.2.2 建立模式映射

建立模式映射由两部分组成：建立数据字典信息和建立数据类型映射关系。

(1)建立数据字典信息。建立数据库信息、数据库内表信息、表内字段信息三类数据字典信息，并保存到本地TYCHE中，数据字典存储结构如图8所示。

图8 数据字典存储表结构

Fig.8 The structure of dictionary storage table

①获取数据源数据库信息，并且写入TYCHE的“数据库表”(T_DATABASEINFO)。

在数据源注册后，执行SQL语句获取数据源数据库信息。以Oracle数据库为例，在QT下获取数据源数据库信息的方式为：QsqlQuery::exec(“SELECT * FROM USER_USERS”)。

通过执行SQL语句INSERT INTO T_DATABASEINFO VALUES，将获取到的数据库信息写入T_DATABASEINFO表。

②获取表的信息，并且写入TYCHE的“数据表表”(T_TABLEINFO)。

通过执行获取表信息的SQL语句，获取数据源内表的信息。以Oracle数据库为例，在QT下获取数据源内表的信息的方式为：QsqlQuery::exec(“SELECT * FROM USER_TABLES”)。

通过执行SQL语句INSERT INTO T_TABLEINFO VALUES，将获取到的表信息写入T_TABLEINFO表。

③获取字段信息，并且写入TYCHE的“数据字段表”(T_FIELDINFO)。

通过执行获取字段信息的SQL语句，获取数据源内字段信息。以Oracle数据库为例，在QT下获取数据源内字段信息的方式为：QsqlQuery::exec(“SELECT COLUMN_NAME,DATA_TYPE,DATA_LENGTH FROM USER_TAB_COLUMNS WHERE TABLE_NAME = '%1'").arg(strtable) ”)，其中strtable为表的名称。

通过执行SQL语句INSERT INTO T_FIELDINFO VALUES，将获取到的字段信息写入T_FIELDINFO表。

(2)建立数据类型映射关系。针对接入系统中的各异构数据库数据源，提取支持的所有数据类型，保存到本地TYCHE数据库进行统一管理，数据库中对数据类型管理的存储结构如图9所示。

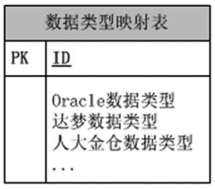

图9 数据类型存储结构

Fig.9 The structure of data types

本系统为各数据源的各类数据类型设置一个默认数据类型，以降低用户配置的难度，同时在数据迁移时运行用户手动配置实际的数据类型。考虑到数据类型映射往往为单向映射，本系统中在存储数据类型映射关系时添加映射源字段，以指明当前映射关系中可以作为源的数据源类型。

4.2.3 创建同步任务

同步任务控制整个数据同步的流程。如图10所示为同步任务进行一次同步的流程图。

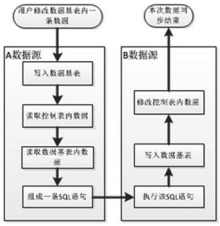

图10 同步任务流程图

Fig.10 The flowchart of synchronization task

用户修改A数据源的数据基表内一条数据，在同步任务内就会开始一次数据同步的过程，最终完成将该条数据变更到B数据源内。

下面以用户在A数据源内写入一条主键值为X的数据为例，说明同步任务的整个流程。其中A数据源与B数据源内数据基表名称为testTable，控制表为CTRLTABLE。

(1)A数据源：在A数据源上主要完成数据基表的更新与形成操作SQL语句，由以下四步完成。

①写入数据基表。将本次要写入的数据首先写入数据基表，只有写入成功才能触发INSERT事件触发器，触发器触发后，将在控制表CTRLTABLE写入一条数据，TableName为“testTable”，FiledKey为X，OperateType为“INSERT”。

②读取控制表内数据。使用select语句在控制表CTRLTABLE内读取数据，条件为IsSync=0，IsSync为0表示本条数据未被同步。

③读取数据基表内数据。使用select语句在数据基表testTable内读取一条数据，条件是关键字为X。

④组成一条SQL语句。根据控制表内数据与数据基表内数据形成一条操作语句，由于控制表的OperateType为“INSERT”，因此本条SQL语句为一条insert语句。

(2)B数据源：在B数据源上主要完成基础表的更新与处理数据振荡问题。

①执行该SQL语句。在B数据源上执行一条SQL语句。

②修改控制表内数据。本步解决数据振荡问题，在B数据源的控制表CTRLTABLE中使用select语句查找关键字为X的一条数据，并且将该数据的IsSync设置为1，即该条数据变更不需要进行同步。

经过以上过程即完成一次数据同步，由于A数据源与B数据源上均配置触发器与控制表，因此数据从B数据源到A数据源是完全相同的过程。

4.3 应用与验证

该系统已经成功应用在某舰船系统中，该舰船系统的软件环境如表6所示。

表6 某舰船系统软件环境

Tab.6 Ship system software environment

数据库	操作系统	IP地址	数据库
Oracle11g	Linux	172.16.200.11	设备故障数据库
达梦7	中标麒麟	172.16.200.10	设备故障数据库

系统验证分为两个阶段。第一个阶段是船体离岸各设备加电运行阶段，此时数据故障信息首先保存在达梦数据库中，通过本文所设计的系统同步到Oracle数据库中。第二个阶段是船体靠岸后，用户分别从Oracle与达梦数据库中查找数据，经过对比发现两个数据库的数据完全一致。

实际应用结果表明，采用本文所设计的系统，对Oracle、达梦数据同步效果较好，达到预期效果。

5 结论(Conclusion)

当前异构数据库之间的数据同步已经成为数据库领域研究的重要方向。本文在现有成熟数据库产品的基础上，分析解决了数据类型差异、数据捕获策略、同步振荡等数据同步

(下转第5页)