

文章编号: 2096-1472(2016)-02-10-04

基于隔离网闸的异构数据库同步技术研究是实现

崔恒香¹, 冯 径¹, 马玮骏¹, 张 庆², 庞 琨³, 余 堃⁴

(1.解放军理工大学气象海洋学院, 江苏 南京 211101;

2.南京军区空军气象中心, 江苏 南京 210018;

3.96151部队, 安徽 黄山 245000;

4.61741部队, 北京 100000)

摘 要: 数据同步是分布式环境中维护各节点数据库数据一致性的一项关键技术。本文针对存在物理隔离网闸的应用环境, 提出一种基于触发器的SQL文件级异构数据库同步方法, 该方法适用于企业异构数据库同步。文中将阐述隔离网闸环境下异构数据库同步的工作原理, 提出同步系统的总体框架, 给出变化捕获、SQL文件生成、文件传输、数据更新和故障处理这五个环节的实现细节。最后通过在洪水预报系统中的实际应用, 验证本文方法的高效性及可靠性。

关键词: 数据同步; 网闸; 异构数据库; 触发器; SQL

中图分类号: TP302.7

文献标识码: A

Research and Implementation of Heterogeneous Database Synchronization Based on Netgap

CUI Hengxiang¹, FENG Jing¹, MA Weijun¹, ZHANG Qing², PANG Kun³, YU Kun⁴

(1. Institute of Meteorology and Oceanography, PLA University of Science and Technology, Nanjing 211101, China;

2. Air Force Meteorological Center of Nanjing Military Area Command, Nanjing 210018, China;

3. 96151 Army, Huangshan 245000, China;

4. 61741 Army, Beijing 100000, China)

Abstract: Data synchronization is a key technology to maintain the consistency of the database in distributed environment. In view of the application environment with physical isolation gateway, this paper proposes a new method of SQL file-level heterogeneous database synchronization based on trigger, which is suitable for enterprise. Then, expounds the principle of the gap under the environment of heterogeneous database synchronization, proposes overall framework for the synchronization system, and the implementation details of the five parts of the steps are given, such as the change capture, file generation, file transfer, data update and fault handling. The efficiency and reliability of this method is verified by the practical application of the flood forecasting system.

Keywords: data synchronization; netgap; heterogeneous database; trigger; SQL

1 引言(Introduction)

网络技术的飞速发展和分布式计算的广泛应用, 大大方便了企业的跨地域发展, 许多企业通过不断重组和分散经营来提高效率, 形成了分散、异构的环境特点。这些机构在地域上虽然分散, 但在管理上相对集中, 往往既要有各机构的局部控制和分散管理, 又要求各机构之间的信息能灵活交流和共享, 能统一管理和使用^[1]。所以对于异构数据库同步技术的研究是十分有必要的。

绝大多数企业并不是把这些地域上分散的应用数据库通过直接的传输通道进行连接, 考虑到网络传输的安全性, 往往在传输通道上添加物理隔离网闸, 将两者隔离, 建立单向连接, 来保证信息传递的安全。网闸对应用的支持主要是通过应用协议的剥离和重建来完成的, 能够在保证安全的情况下尽可能支持数据交换, 如果不安全则断开, 可以解决数据库保护所需要的完整性, 可用性, 保密性, 可鉴别和防抵赖等特性, 因此被广泛应用在数据库保护上^[2]。目前, 在物理

隔离网闸的特定网络环境下, 如何有效地解决异构数据库间的数据同步问题已成为企业数据库系统应用的重要环节。

本文在物理隔离网闸的环境下, 设计出一种效率高、可靠性强的异构数据库同步方法, 实现企业内外网间的数据同步。

2 相关工作(Related work)

2.1 异构数据库同步的基本方法

数据同步大致可分为存储系统层同步、操作系统层同步和数据库系统层同步^[3], 本文所涉及的是数据库系统层同步技术的研究。变化捕获作为数据库系统层面数据同步的基础, 直接决定了数据库的更新方式, 其方法包括以下几种: 基于触发器法、基于快照法、基于日志法、基于API法等。

基于触发器法是在源数据库为同步对象创建触发器, 当对同步对象进行增删改等DML命令时, 触发器被唤醒将变化传播到目标数据库; 基于日志法是通过分析数据库日志信息来捕获同步对象的变化序列; 基于快照是指数据库中存储对象在某一时刻的即时映像, 通过为同步对象定义一个快照

或采用类似方法将它的当前映像作为更新副本；基于API法是指在应用程序和数据库之间引入中间件，由它提供一系列API(例如JDBC驱动程序)这些中间件，在完成应用程序对数据库修改的同时，也把同步对象的变化序列记录下来从而达到捕获的目的^[4]。

对于不同数据库而言，数据库的体系结构、数据类型、所支持的标准都不尽相同，所以上述几种方法在实际应用中都有或多或少的缺陷。

2.2 数据库厂商解决异构数据库同步的方法

目前学术界对于异构数据库同步的研究并不多。相比之下，产业界对异构数据库同步研究取得了不少技术成果，其中一些已经得到了广泛应用，详见表1。

表1 厂商数据库同步方案对比表

Tab.1 Vendor database synchronization scheme comparison table

厂商	方案	缺点
Oracle	通过数据转换器及快照方式实现同步	复杂性高，成本较大
Sybase	利用LTM(日志传输器)实现同步	对网络要求较高
IBM	采用触发器机制实现同步	占用较多系统资源
SQL Server	出版者/预订者方案实现同步	实现较为复杂

四大数据库厂商提出的方案均重点关注异构数据库向自身的数据同步，在实现上也采用与DBMS(DataBase Management System)密切结合的技术，并且具有成本高、实现复杂、缺少灵活性等缺点。设计并实现一种新型的灵活性强、成本小且复杂性较低的异构数据库同步方法十分有必要。

3 基于触发器的SQL文件级异构数据库同步方法 (SQL file level heterogeneous database synchronization method based on trigger)

基于触发器的SQL文件级异构数据库同步方法的基本思想就是为源数据库创建触发器，当源库数据发生变化时，触发器在变更表中通过SQL语句的重组记录目标库能够直接执行的SQL语句，然后将SQL语句定时导出到文件，并通过网闸将文件传输到目标端，目标库定时执行SQL文件将源库中的变更数据更新到目标库，同时在生成SQL文件、文件传输以及数据更新等步骤执行完后将结果状态向源数据库端反馈，针对同步过程中出现的不同失败状态采取不同措施，以实现整个数据同步过程的故障定位与恢复。

3.1 异构数据库同步设计方案

整个系统由变化捕获、SQL文件生成、文件传输、数据更新和故障处理五个功能模块组成。变化捕获模块负责以SQL语句记录变化，SQL文件生成模块把SQL语句定时导出到数据文件，文件传输模块负责将SQL文件传输到目标端，数据更新模块负责把传送过来的数据更新到目标数据库，故障处理模块将根据各个模块的状态反馈，对同步流程中出现的故障点

实施针对性的补救措施。下面将对系统主要模块的实现方法进行详细阐述，同步系统流程设计如图1所示。

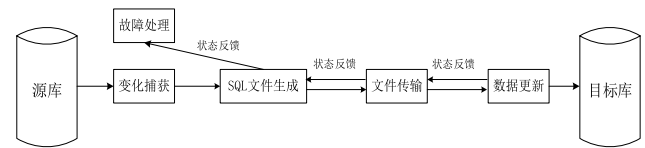


图1 同步系统流程图

Fig.1 Synchronization system flow chart

3.2 各模块具体设计

3.2.1 基于触发器的变化捕获

变化捕获是捕获源表的变化序列的过程。触发器法和日志法是最基本的变化捕获的方法，目前流行的商用数据库都提供触发器机制或日志分析工具。而本文需要在捕获变化的同时重写目标库能执行的SQL语句并记录下来，因此本方法将采用触发器法，能够在触发器程序块中编写重组SQL语句的程序。

基于Oracle触发器重组SQL语句的基本原理：

Oracle触发器机制中使用了两个特殊的临时驻留内存的关键词：New和Old。New用于存储INSERT和UPDATE操作所影响的行的副本；Old用于存储DELETE和UPDATE操作所影响的行的副本。当执行INSERT操作时，新建记录被同时添加到触发器表和New中；当执行DELETE操作时，记录从触发器所在表中删除，并传输到Old中；当执行UPDATE操作时，旧记录被移到Old中，新记录被添加到New和触发器表中。因此通过获得Old和New中行的副本再进行重组便能够得到目标库可直接执行的SQL语句。

3.2.2 SQL文件生成的方法

上文通过触发器捕获变化并将重组后的SQL语句记录在变更表中，那么只要通过建立操作系统层的定时任务程序对数据库建立连接按规则获得变更表中的SQL语句输出到数据文件，即完成了SQL文件的生成。

基于Oracle的sql*plus工具生成SQL文件的基本原理：

sql*plus是与Oracle进行交互的客户端工具，通过Spool命令将数据导出到SQL文件是它的主要功能之一，可以通过编写批处理程序(Windows平台)登录sql*plus并执行带有输出控制设置的Spool命令的ctl文件，再通过操作系统自带的定时任务功能定时执行bat文件就可以实现SQL文件的定时导出。

3.2.3 文件传输及数据更新

文件传输模块主要包括源库端、网闸和目标库端，源库端定时将生成的SQL数据文件从导出文件目录移动到网闸的内网存储单元，同时复制一份存放到数据备份单元，并且在任务执行失败的情况下向故障处理模块发送类型为failure的消息文件(用于故障处理模块进行故障分析)；网闸定时将内网存储单元的SQL文件转移到外网存储单元；目标库端定时将外网存储单元的SQL文件移到目标库文件系统，并且在任务执行失败

的情况下向故障处理单元发送类型为failure的消息文件，文件传输模块如图2所示。

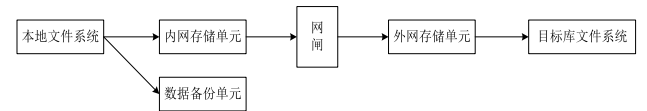


图2 文件传输模块构成图

Fig.2 File transfer module

目标库端同样创建定时任务，由于目标库文件系统所存放的就是目标库可直接执行的SQL文件，通过定时执行SQL文件即能完成数据的同步更新。并且在任务执行失败时向故障处理单元发送类型为failure的消息文件；同步成功则发送类型为success的消息文件。

3.2.4 基于预先配置策略的故障处理

源库端根据总结的历史经验配置不同failure状态故障类型的应对措施，故障处理模块通过读取同步系统各模块实时反馈的同步状态消息文件，根据不同的failure状态故障类型实施相应的应对的措施，使得故障点恢复运行保证同步系统的正常运行，从而提高数据同步系统的稳定性和可靠性。

4 实际实现与应用(Practical implementation and application)

实际应用中考虑到现有的软硬件环境，实现工作主要放在从源端Oracle到目标端Sql Server数据库业务数据表的数据同步。并根据同步系统各模块以及主要工作创新点阐述实现与应用。

4.1 变化捕获的实现

一般的触发器内部不能够对INSERT、DELETE、UPDATE分别触发。由于使用触发器是为了变化捕获的高效和灵活，如果为每张表分别建立INSERT、DELETE、UPDATE三个触发器，这无疑会对数据库系统的查询速度、性能带来影响。因此必须减少触发器的数量。

4.1.1 优化后的触发器编写

解决方法是在源表上只创建一个包含INSERT、DELETE、UPDATE操作的触发器，本文将触发器开头设计为：

```
CREATE TRIGGER trigger_name
BEFORE INSERT, UPDATE, DELETE
ON table_name
FOR EACH ROW
AS Sql_statement
```

触发器程序内部通过获取关键字Old和New中的行数据副本来判断具体是INSERT、DELETE还是UPDATE操作。

4.1.2 基于触发器重组SQL语句解决数据库异构的问题

因为UPDATE操作原理上就等于一个DELETE操作加上INSERT操作顺序执行的，因此通过UPDATE操作来说明会更全面，本文SQL语句重组流程如下：

Step1：Old.colname字符获取触发器表UPDATE之前(将

被update影响)行的列值；New.colname字符获取触发器表UPDATE之后行的列值。

Step2：通过CONCAT函数拼接Delete语句。

Step3：接Insert语句，由于Insert操作涉及表每列值的插入，因此我们必须考虑源库表中每列与目标异构库表的数据类型差异以及存在的表结构差异(表结构差异与具体企业业务相关联)。

本文列举了数字、字符串、时间等常用数据类型，在Oracle中存储类型分别为number、varchar2、date；对应SQL Server中为int、varchar、datetime；并且两表的表结构存在差异：目标端用单个id列存储源端id、num两列的连接值。详见表2和表3。

表2 源端Oracle表

Tab.2 Source Oracle table

列名	属性	值	
id	number	101	102
num	number	10	11
str	varchar2	test1	test2
time	date	2015/10/15	2015/10/16

表3 目标端SQL Server表

Tab.3 Target SQL Server table

列名	属性	值	
id	int	10101	11102
str	varchar	test1	test2
time	datetime	2015/10/15	2015/10/16

Part1：解决数据类型的差异，Oracle到SQL Server需把date类型时间转换成datetime类型：通过将date类型数据转换为字符串，再将字符串拼接到Cast函数中，其中Cast函数是SQL Server的类型转换函数。

Part2：解决数据库表结构的差异，本文主要解决表字段合并操作，因此需要将源端的id和num字段按照顺序拼接在一起进行存储来完成字段值的连接。

Step4：将拼接好的Delete、Insert语句记录到变更表中，详见表4。

表4 变更表

Tab.4 Change table

id	sql	type	insert time
1	DELETE FROM table1 WHERE id=10101;	delete	2015/10/25
2	INSERT INTO table1 VALUES(10101,' A Test', CAST(' 2015-10-25 13:42:14' ,AS DATETIME));	insert	2015/10/25
3	DELETE FROM table1 WHERE id=11102;	delete	2015/10/25
4	INSERT INTO table1 VALUES(11102,' A Test', CAST(' 2015-10-25 13:42:26' ,AS DATETIME));	insert	2015/10/25

变更表temptable中sql字段存储的便是重组后的SQL语句，并且记录了操作类型和变更时间便于根据不同的策略取得相应的数据。

4.2 SQL文件生成的实现

SQL文件生成主要步骤：配置操作系统定时任务，任务执行对象为bat文件；编写包含登录sql*plus且执行Spool命

令的bat文件：Path1\sqlplus username/password Path2\filename.ctl，其中，Path1为sql*plus的路径；Path2为ctl文件的路径；ctl文件分为输出控制设置和Spool导出语句两部分：

Part1：输出控制设置部分：例如设置每行的长度、脚本中命令执行结果是否显示、标准输出每行的拖尾空格是否去除等等设置。

Part2：Spool导出语句部分如下

```
Spool filepath/filename.sql
//导出SQL文件名称及位置
Select colname From tablename Where_statement
//导出的内容为查询结果
```

根据条件查询结果按设置导出到文件。4.1中将重组后的SQL语句记录在表temptable的sql字段，因此此处直接对sql字段内容查询便实现了源表数据变更对应SQL的导出。

Spool命令的优化：一方面，在导出SQL语句的同时添加删除已导出数据的delete语句，实时删除临时变更表冗余的数据，减少对系统资源的过多占用，达到优化性能的效果；另一方面，Spool导出命令中通过获取临时变更表最新记录时间MaxInsertTime，在取数据的select语句和删除数据的delete语句where条件中均添加变更时间小于等于MaxInsertTime的条件，在导出MaxInsertTime时间点之前的数据同时删除的也保持一致，这样做的好处相比直接select、delete表的所有数据，避免了在取完数据后删除数据之前刚好有数据插入表中所带来的数据误删。

4.3 SQL文件的传输与执行以及相关的故障处理

SQL文件的传输模块主要靠物理隔离网闸进行源端到目标端的搬运，以及移动文件和生成反馈状态文件的程序。目标端SQL文件执行模块同样是配置定时任务来完成，本文目标端为SQL Server库，那么通过定时任务打开cmd.exe执行语句：osql-S127.0.0.1-Usa-Psa-ipath\filename.sql便可完成数据的更新，其中osql为SQL Server的命令行工具，-i表示要执行SQL文件路径。

当同步过程中某模块出现故障时会向故障处理模块发送消息文件，故障处理模块通过读取文件中故障类型，根据故障类型从策略表获取应对策略，并向发生故障的模块发送应对策略指令使其实施相应的措施恢复运行，从而提高数据同步系统的稳定性和可靠性。策略表预先配置，详见表5。

表5 策略表

Tab.5 Policy table

故障类型	状态反馈模块	故障原因	应对策略
Type_1	源端文件传输模块	本地文件系统未生成SQL文件	重启源库
Type_2	目标库文件传输模块	外网存储单元无SQL文件	重启网闸
Type_3	数据更新模块	目标库出现故障无法同步	重启目标库
Type_4	数据更新模块	SQL文件损坏或更新过程中出现故障	获取备份单元SQL文件重传

4.4 实际应用

本文的“基于触发器的SQL文件级同步法”已经成功地应用于某地区洪水预报系统中。该系统包括内、外两个网络，内网是一个水流量信息采集系统；外网是一个B/S架构洪水预报信息系统，提供洪水预报的计算处理与查询。由于内、外网两个应用系统的数据分别放置于不同的数据库上，因此为了保持内、外网两个数据库的一致性，须将内网的数据及时地同步到外网，并且需要考虑数据传输的安全性，这就是一个典型的基于网闸的异构数据库间数据同步的问题。

运行结果表明，随着记录数的增长，“基于触发器的SQL文件级同步法”的时间复杂度呈线性增长，空间复杂度主要为内存消耗，基本上保持常量，因而总体性能比较理想，并且相对于原先系统采用的“基于API的传统轮询同步法”，效率提高明显，可靠性、稳定性都得到了保障。

5 结论(Conclusion)

本文针对物理网络隔离的应用环境，提出了一种“基于触发器的SQL文件级同步法”适合于企业数据同步。该方法运用触发器对源数据库变化进行捕获，并转换成目标库可直接执行的SQL语句记录下来且定时导出到数据文件，通过执行数据文件更新目标数据库，从而达到数据同步。该方法具有适应异构系统、效率高、可靠性强、稳定性佳的特点。

参考文献(References)

[1] 郑海明.基于SQL还原法的异构数据库同步技术的研究与实践[J].计算机时代,2008(10):15-18.

[2] 张晖,欧阳慎.基于oracle的跨网闸数据同步方案研究[R].郑州铁路局“十百千”人才培养助推工程论文集,会议时间:2011.

[3] 王新伟.基于Oracle数据库的逻辑数据同步技术在实践中的应用研究[J].电子技术与软件工程,2015(08):195-198.

[4] 张大朋,陈驰,徐震.异构数据库复制技术的研究与实践[J].中国科学院研究生院学报,2012(01):101-108.

作者简介:

崔恒香(1987-),男,硕士,学生.研究领域:信息与系统集成.

冯 径(1962-),女,博士,教授.研究领域:系统集成与应用.

马玮骏(1980-),男,博士,讲师.研究领域:系统集成与应用.

张 庆(1978-),男,本科,工程师.研究领域:信息工程.

庞 琨(1980-),男,硕士,工程师.研究领域:信息与系统集成.

余 堃(1983-),男,硕士,工程师.研究领域:信息与系统集成.