

文章编号: 1005-8451 (2013) 03-0018-03

# 大型管理信息系统数据库更新策略研究

易明中<sup>1</sup>, 李樊<sup>1</sup>, 刘春明<sup>2</sup>

(1.中国铁道科学研究院 电子计算技术研究所, 北京 100081;

2.中国信息安全测评中心 科技处, 北京 100085)

**摘要:** 大型管理信息系统由于研发、提供服务和管理、维护等原因, 经常需要对其各数据库进行修改、改进和更新等操作。本文设计并实现了一个独立性、通用性较高、基于Oracle分布式数据库的数据库自动更新策略。实际使用表明: 该策略大幅减少了人工乏味、重复性工作。大型管理信息系统的数据库更新方式实现了从手工作坊式更新到工厂化大规模更新的变革。

**关键词:** 分布式数据库; 更新; 脚本

**中图分类号:** TP392      **文献标识码:** A

## Update strategy research on databases of large-scale Management Information System

YI Mingzhong<sup>1</sup>, LI Fan<sup>1</sup>, LIU Chunming<sup>2</sup>

(1. Institute of Computing Technologies, China Academy of Railway Sciences, Beijing 100081, China;

2. Science and Technology Department, China Information Technology Security Evaluation Center, Beijing 100085, China )

**Abstract:** Usually, modification, amendment, update and so on were necessary for multiple databases of a large-scale Management Information System for reasons like research, development, offering service and administration, maintenance. In this paper, an independent and universal Database Automatic Update Strategy based on Oracle distributed database was originally designed and achieved. Practical running demonstrate that laborious, tedious, repeating human labor could be greatly attenuated. The renovation that database update for large-scale MIS could be evolved from manual workshop to manufactory assembly came into reality.

**Key words:** distributed database; update; script

由于研发、提供服务和管理、维护等原因, 管理信息系统可能常需对其数据库进行修改、改进和更新等各种操作。且要保证这些操作在各数据库间的一致性, 对于大型管理信息系统来说这较难实现。

有多种修改、改进和更新数据库的方式<sup>[1-3]</sup>。可快捷、灵活地批量操作, 提高数据访问的效率, 便于相关的数据处理<sup>[4-5]</sup>, 降低命令输入量, 统一管理, 事先和事后审核, 可追溯, 重用和回退。

本文所依据的大型分布式管理信息系统的数据库脚本按日期编号。比如, date1.sql 代表 date1 的第 1 个脚本, date1\_1.sql 代表 date1 的第 2 个脚本, …。脚本号既唯一, 又有具体含义。为叙述方便, 以下对各数据库的修改、改进和更新等

各种操作都统称为更新, 运行脚本简称为打脚本。为保证一致性, 一旦某脚本在任一数据库被运行, 则它就不能再被更改了 (immutable), 但可将所需的更改做成一个新脚本。需在本文所依据的大型管理信息系统的各数据库运行的脚本大致可分为 3 类: (1) 打往所有或多个数据库的脚本; (2) 随前台补丁包只打往一个或多个数据库的脚本; (3) 单独打往一个或少数几个数据库的脚本。

### 1 手动数据库更新方式及弊端

最初主要采用不一致的方法手动直接更新数据库, 难以共享数据库变更, 前台开发人员难以与数据变更保持同步, 用户也可能遇到数据不一致和数据损坏等问题。后来, 虽然主要通过数据库脚本来更新数据库, 但还是手动方式。如图 1

收稿日期: 2012-07-12

作者简介: 易明中, 助理研究员; 李樊, 助理研究员。

所示,通过第三方软件“Radmin”或Windows自带的“远程桌面连接”功能手动远程登录到某服务器,用安装在该服务器上的第三方软件“PL/SQL Developer”手动登录连接到某数据库。在其“命令窗口”中手动运行patch.sql(可取为其它文件名)来调用需运行的脚本(比如date.sql、date\_1.sql等),“命令窗口”会显示脚本的运行过程、出现的问题等。利用“PL/SQL Developer”的“编译无效对象”功能来检查并编译可能的无效对象。最后还需手动编制一些文档以记录脚本的运行过程、出现的问题、无效对象及其解决过程。

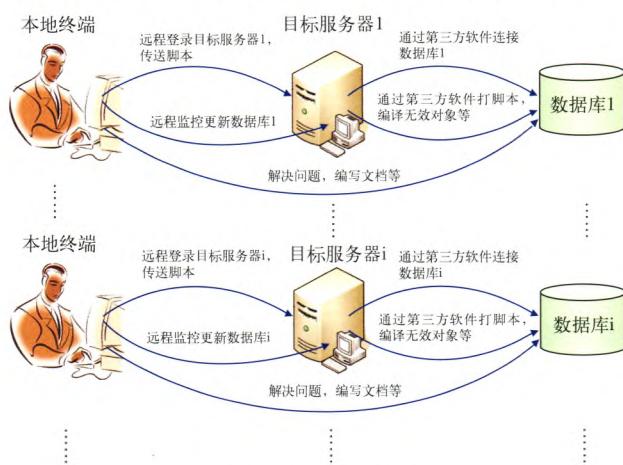


图1 纯手动方式数据库更新

这种纯手动的数据库更新方式的弊端是：(1)耗时长；(2)效率低；(3)功能差；(4)集中性差；(5)计划性差；(6)可追溯性差；(7)扩展性差；(8)成本高。

## 2 改进策略

通过改进,选定现有的任一服务器作为集中点服务器,运行all\_patch.bat就能为各数据库依次打开SQL\*Plus窗口并自动远程连接,人所耗时大幅减少,改善了上述的第1个弊端。且只利用了一个现有的服务器,其它服务器基本不受影响,改善了第2个弊端。此时,人就不必记住并键入登录到各服务器的用户名、密码和连接到各数据库的用户名、密码,改善了第3个弊端。对各数据库的更新就可能集中完成,改善了第4个弊端。能将该all\_patch.bat添加进一个任务计划,在任意的计划时段定期自动运行,改善了第5个弊端。

### 2.1 针对第1类脚本的改进策略

(1)对于打往所有或多个数据库的共性的第1类脚本,可制作一个共性的放在系统最上层的patch\_ALL.sql。在all\_patch.bat为各数据库依次打开的SQL\*Plus窗口中,自动运行原来需在“PL/SQL Developer”的“命令窗口”中手动运行的这个patch\_ALL.sql,以调用需要打往各数据库的那些脚本。

(2)记录打脚本时的过程、出现的问题、无效对象等。共性的脚本打到每个数据库时的运行过程、出现的问题、无效对象等是个性化的东西。比如,patch.sql打到第1个数据库时可能没有出现问题和无效对象,但打到第i个数据库时就可能出现了。所以只由一个共性的patch\_ALL.sql很难完成所有这些个性化的工作。所以,在将patch.sql这类共性的脚本打往所有或多个数据库时,为每个数据库调用date.sql等脚本是最共性的工作,可集中到一个文件patch\_ALL.sql中来完成,放在系统的最上层。而为每个数据库生成文档以记录脚本的运行过程、出现的问题、无效对象等个性化的工作,应分布到下层的多个文件,比如patch\_1\_ALL.sql、…、patch\_i\_ALL.sql、…。由各patch\_i\_ALL.sql调用patch\_ALL.sql,并由各patch\_i\_ALL.sql生成日志文件Destination\_i.LOG以记录脚本打往数据库Destination\_i时的运行过程、出现的问题等。还需制作一个reCompile\_i.sql以统计和编译无效对象,并生成tmp\_i.sql以记录无效对象等信息,reCompile\_i.sql也由各patch\_i\_ALL.sql来调用。由于上述工作都是由计算机在一个集中点服务器上自动完成,因而大幅改善了上述的第1个~第4个、第6个弊端,尤其是改善了第6个弊端。

打第1类脚本最终归结为运行批处理文件all\_patch.bat。因而大幅改善了第7个弊端。

### 2.2 针对第2类脚本的改进策略

第2类随前台补丁包只打往一个或多个数据库的个性化脚本被打包在对应的后台补丁包中,所以对应的大多是个性化的工作。比如,给数据库Destination\_1打个性化脚本时调用脚本的文件是patch\_1.sql,给数据库Destination\_2打个性化脚本时调用脚本的文件是patch\_2.sql。patch\_1.sql实际上是将打第1类脚本时的patch\_ALL.sql

和 patch\_1\_ALL.sql 的功能合并, patch\_2.sql 实际上是将 patch\_ALL.sql 和 patch\_2\_ALL.sql 的功能合并。

### 2.3 针对第3类脚本的改进策略

第3类脚本和第2类脚本本质上可归为一类, 即都是只打往一个或多个数据库的个性化的脚本, 只不过第3类脚本不随前台补丁包打往一个或多个数据库。所以, 对于第3类脚本, 处理方法跟第2类脚本基本上类似。对于 Destination\_1 来说, patch\_1.sql 既可用于第2类脚本, 也可用于第3类脚本; 对于 Destination\_2 来说, patch\_2.sql 既可用于第2类脚本, 也可用于第3类脚本; 对于 Destination\_i, 亦然。

与打第1类脚本一样, 在打第2、第3类脚本时也需为各数据库依次打开 SQL\*Plus 窗口, 并在该窗口中自动运行原来需在“PL/SQL Developer”的“命令窗口”中手动运行的 patch\_i.sql 等以调用需打往各数据库的脚本。也有一个类似于前述的 all\_patch.bat 的批处理文件, rescue.bat。所以, 更新第2、第3类脚本时, 最终也归结为运行一个批处理文件 rescue.bat。

为各数据库自动打各类脚本最终都归结为运行一个批处理文件, 如图2所示。

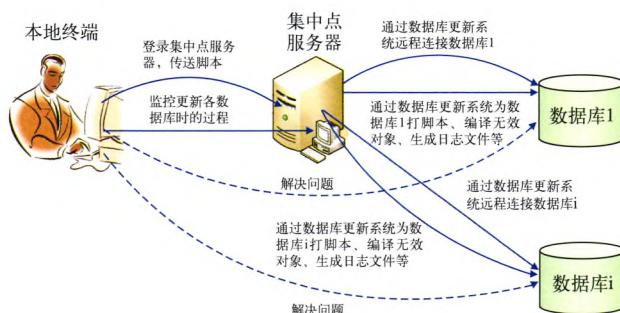


图2 数据库自动更新系统

## 3 数据库更新改进策略的优、缺点

### 3.1 优点

采用上述大型管理信息系统数据库更新的改进策略, 耗时短、效率高、功能强、集中性高、计划性强、可审核性高、可追溯性好、通用性强、依赖性低、扩展性好、成本低。本策略利用了 Oracle 数据库的一些技术, 对操作系统和第三方工具的依赖性不高。对该策略稍加改动, 也可用

于其它类型软件的多个数据库的更新。

### 3.2 缺点及进一步改进思路

尽管采用改进的策略, 偶尔还是需要人工解决数据库更新时的突发、异常等问题。另外, 本策略占用了一些通信资源。在网络状况不好的时候, 可能导致数据库更新较慢。若集中点服务器和某目标服务器 i 之间的网络时断时续, 还可能造成对数据库 i 的更新被中断。另外, 对于一些特别大的脚本文件, 从集中点服务器对数据库 i 远程打脚本则也可能较慢。对于以上这几种情况, 可先登录目标服务器 i, 把需更新的脚本传送到服务器 i, 然后启动服务器 i 上的对应于某数据库 i 的分系统 i。

## 4 结束语

改进策略将共性的、机械的、可重复的工作分解出来, 让计算机来完成。而把个性化的、不能重复的工作留给人来做。因而, 人、机能互相取长补短, 大幅提高了数据库更新的技术含量和效率, 大幅减少了人工和耗时。且通过数据库脚本来持久化对数据库的各种操作, 有利于对数据库的管理、维护和更新, 在某种程度上实现数据库的版本管理。这是一个独立性、通用性较高的基于分布式数据库的数据库批量自动更新策略。实际使用表明, 该策略较好地适应了集中批量更新大型管理信息系统的各数据库的要求, 实现了数据库的手工作坊式更新方式向工厂化大规模更新方式的转变。

### 参考文献:

- [1] 刘洋, 杨志. 一种数据库系统及其升级的方法 [Z]. 深圳: 中兴通讯股份有限公司, 2010, 02.
- [2] 崔羽, 韦光亮. 一种数据升级方法及系统 [Z]. 北京: 金蝶软件 (中国) 有限公司, 2008, 12.
- [3] 向稳新, 唐璐. 一种基于 SQL 的数据库管理方法和装置 [Z]. 南京: 南京中兴力维软件有限公司, 2012, 01.
- [4] 常伟, 常红平. 脚本加速 DB2 存储过程的开发 [Z]. 北京: IBM 中国软件开发中心, 2007, 06.
- [5] 常伟. 利用 Shell 脚本实现 DB2 数据库的敏捷开发 [Z]. 北京: IBM 中国软件开发中心, 2010, 03.

责任编辑 杨利明