

Java 存储过程实现 Oracle 远程逻辑备份与恢复^①

苟 盛 王茂芝 郭 科 (成都理工大学 信息管理学院 四川 成都 610059;

摘要: 在介绍 Oracle 数据库中 Java 存储过程及其开发步骤的基础上, 通过实例详细论述了如何开发 Java 存储过程实现 Oracle 数据库远程逻辑备份与恢复。实践表明, Java 存储过程具有提高系统执行效率和可维护性、增强系统安全性等优点, 同时它还为客户端在网络环境下实现复杂应用提供了良好解决方案。

关键词: 存储过程; 远程逻辑备份与恢复

Java Stored Procedures in Oracle to Achieve Remote Logical Backup and Restoration

GOU Sheng, WANG Mao-Zhi, GUO Ke

(College of Information Management, Chengdu University of Technology, Chengdu 610059, China)

Abstract: After introducing Java stored procedures in Oracle database and its development steps, this paper discusses how to develop Java stored procedures to achieve Oracle remote logical backup and restoration through an example. Practice shows that Java stored procedure dose not only have the advantages of improving the efficiency and maintainability of system, and enhancing system security, etc., but also provides a good solution to the client to achieve complex application in the network environment.

Keywords: stored procedures; remote logical backup and restore

1 引言

Oracle 是基于网络计算的数据库, 通常都集成了 Java 虚拟机(Oracle JVM), 并内置了对 Java 的支持。为了降低客户端的负荷, 在执行数据增、删、改、查的时候, 可将复杂的数据计算等操作放在 Oracle 数据库服务器端, 以充分利用资源, 提高数据计算速度。Oracle JVM 不仅实现了将很多原先需要使用 Java 完成的任务放到 Oracle 数据库中完成, 还提供了一种设计 Oracle 函数、存储过程的方便方法, 即允许在数据库中运行被称为 Java 存储过程的 Java 程序。

通常情况下, Oracle 开发人员使用 PL/SQL 来开发和管理系统的存储过程, 但 PL/SQL 是一种只针对 Oracle 数据库, 经过高度优化的、优秀的过程语言, 它编写的存储过程只能在 Oracle 数据库运行。相比之下, Java 语言具有跨平台性和高灵活性。Java 存储过程可以很容易地转化成运行在其它

数据库上的存储过程^[1]。更为重要的是, 它允许重用已经存在的代码, 从而大幅度地提高开发效率。除此之外, 通过 Oracle JVM 来执行 Java 存储过程, 可以充分利用 Java 语言高效的内存回收技术和线程管理能力。

本文着重探讨了客户端在网络环境下开发应用系统实现 Oracle 数据库远程逻辑备份与恢复的关键技术, 并给出了相应的解决方案。

2 存储过程

存储过程是存储在数据库中, 为执行一定任务而组合的一段 SQL 语句集, 通常由声明部分, 执行部分和异常处理部分组成。当存储过程创建时, 通过语法分析, 系统会对 SQL 语句进行编译, 并将执行代码作为数据库对象存储到数据库的高速缓冲存储器中。以后使用时只需从高速缓冲区中调用已编译的代码, 这样便达

^① 基金项目: 国家高技术研究发展计划(863)(2009AA01Z405); 国家自然科学基金(40873035)

收稿时间: 2009-09-18; 收到修改稿时间: 2009-10-30

到了加速 SQL 语句执行,提高系统性能的目的。存储过程具有以下优点^[2]:

① 提高系统执行效率。存储过程可以将多条经过优化编译后 SQL 语句包含在一起执行,客户端和数据库交互的只是输入、输出参数的一次来回传递,传递信息少,因此其执行效率比起执行同样功能的 SQL 语句要高很多。

② 提高系统维护性。在开发大型应用程序过程中可以将一些标准功能逻辑模块作为存储过程放在数据库中,使得模块化程序开发的结构更加清晰,而且易于维护。

③ 增强系统安全性。数据库中的存储过程可分别得到保护。比如,可以授予某用户执行某个特定存储过程的权限(如更新表),但无权访问其它模式对象(底层表)。

④ 保持数据透明性。因为存储过程增加了一个从底层的数据库对象结构中提取出的层。存储过程的实现细节及底层对象结构与存储过程的调用者(客户端)被隔离开了。

⑤ 降低网络负荷流量。这是由于 SQL 语句可以在服务器端以批处理的方式执行,因此当客户端发出执行存储过程的请求时,只有执行存储过程的命令在内部网络上传送而无需从客户端发送多个 SQL 请求。

存储过程在保持以上优点的同时,也有一些缺点,主要体现在:①可能加重服务器方负载;②可能造成并发冲突;③可移植性差。因为各种数据库的存储过程语法差别较大,这会给将来的数据库移植带来困难。

基于以上分析,在实际的数据库应用程序开发过程(如:Java 应用程序)中,可充分发挥存储过程的优点,利用数据访问对象(Data Access Object DAO)设计模式建立同数据库的连接,然后通过对存储过程的调用实现数据库的操作。下面分别展开并通过实例予以说明。

3 Java存储过程开发步骤

Oracle 存储过程是保存在服务器端的数据库中,客户端应用程序是通过 JDBC 建立数据库连接,通过 CallableStatement 接口调用存储过程的。由于 Java 是面向对象程序设计语言,而 Oracle 的 PL/SQL 是结构化程序设计语言,因此,Java 存储过程对 Java 语言添加了限制,如能够被 Oracle 调用的 Java 函数只能

是 static 类型。另外 Java 类文件必须被加载到 Oracle 数据库中,并映射为普通的 Oracle 函数,Java 存储过程才能被 Oracle 调用。服务器端 Java 存储过程加载及客户端调用过程如图 1 所示。

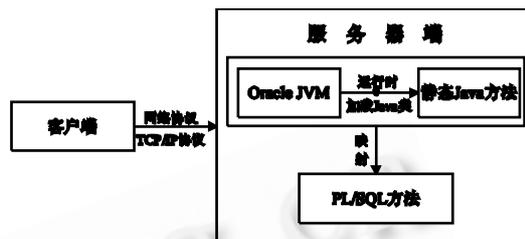


图 1 服务器端 Java 存储过程

加载及客户端调用示意图。

总的说来,创建 Java 存储过程主要由以下几个步骤构成:

① 创建和编译 Java 类文件。这里需要注意:Java 存储过程的 Java 类中没有 main() 方法;采用的编译器必须和 Oracle 数据库中的 Java 虚拟机兼容。

② 利用 Oracle 提供的 loadjava 命令行工具或其它方式将 Java 类加载到数据库中;

③ 将 Java 类文件中的方法、参数类型和返回类型映射为普通的 Oracle 存储过程(其 SQL 对应部分);

④ 赋予用户权限。这里主要指为能够调用该存储过程的用户赋予相关权限以及存储过程本身涉及到的调用系统资源的权限(如文件读写权限)。

4 Oracle数据库备份

数据库备份恢复工作是管理任务中非常常规和重要的内容,以便将来出现数据丢失时把数据库恢复到理想状态。在 Oracle 中,数据丢失可分为物理丢失和逻辑丢失。物理丢失指操作系统级的数据库组件丢失;逻辑丢失指诸如表、索引、或表记录等数据库级组件的丢失。

Oracle 数据库有两类备份方法。第一类为物理备份,该方法实现数据库的完整恢复,但数据库必须运行在归档模式下(业务数据库在非归档模式下运行),且需要极大的外部存储设备;第二类为逻辑备份,业务数据库采用此种方式,此方法不需要数据库运行在归档模式下,不但备份简单,而且可以不需要外部存储设备。Oracle 数据库的逻辑备份分为三种模式:表备份、用户备份和完全备份^[3]。逻辑备份可以备份整个

数据库,或仅备份部分重要数据。但一旦数据库被物理损坏,导致不能启动,逻辑备份的数据不能帮助恢复数据库。Oracle 数据库恢复又包括:完全恢复、不完全恢复、RMAN 备份和恢复等方式。

逻辑备份是创建数据库对象的逻辑拷贝并存入一个二进制转储文件。它需要在数据库启动的情况下使用。EXP 和 IMP 是 Oracle 提供的一种逻辑备份工具。EXP 是把数据库中的数据导出,IMP 是把 EXP 导出的数据导入到数据库中。导出实质就是读取数据库记录集并将该记录集写入文件;导入实质就是读取转储文件并执行其中的命令。

5 远程逻辑备份与恢复应用实例

在涉及到以 Oracle 为后台数据库平台开发网络应用系统时,系统往往需要具备数据库备份和恢复的功能。对于 Oracle 这样的大型数据库,数据库备份工作尤其显得重要。但是系统管理人员不可能实时管理数据库,他们通常需要通过网络实现远程备份和恢复数据库。下面结合攀枝花市矿产资源数据管理系统开发实际,按照第三部分介绍的开发步骤详细讨论如何通过 Java 存储过程实现 Oracle 数据库的远程逻辑备份与恢复。

5.1 创建和编译 Java 类文件

设计开发 Java 存储过程时,应当使其完成相对单一、集中的任务。首先需要编写 Java 源代码,且只有 static 类型的方法才可以作为 Java 存储过程被 Oracle 调用。为了实现 Oracle 远程逻辑备份和恢复功能,Java 类文件设计思路为首先打开操作系统本身的控制台,然后调用 EXP/IMP 命令实现数据库逻辑备份与恢复。

由于 Java 类中调用了 Runtime.getRuntime().exec()方法,因此执行数据库 EXP/IMP 命令的过程中可能出现错误而导致系统主线程死锁,因此,在 Java 程序中,应该启用两个线程,分别调用 getInputStream 方法获取控制台的正常输入流,调用 getErrorStream 方法获取控制台的错误输入流,以避免 Process 的 waitFor 方法导致死锁而造成备份(恢复)失败。具体的 Java 文件编写方式可参考文献[4]。

将 Java 类文件编写完毕后,应当使用和 Oracle 数据库中自带的 Java 虚拟机兼容的编译器将其编译为 class 文件。这里我们在控制台调用 javac 命令进行编译。

5.2 加载 Java 类文件

接下来我们就应当将 Java 类加载到数据库中。这可以通过两种方式实现:

① 利用 Oracle 提供的 loadjava 命令行工具加载;

② 登陆 SQL*PLUS 后使用 CREATE OR REPLACE AND COMPILE JAVA SOURCE NAMED "RunCommand" AS ... 命令。

5.3 建立存储过程

当 Java 类文件加载完毕后,应当将 Java 类映射成 Oracle 存储过程,因此在登陆 SQL*PLUS 后可以使用如下命令完成存储过程的创建。

```
CREATE OR REPLACE PROCEDURE RunHost  
Command (command IN VARCHAR2)  
AS LANGUAGE JAVA NAME 'RunCommand.  
executeCommand (java.lang.String);
```

其中,RunCommand 是 Java 类文件名,executeCommand 是类文件中的一个 static 方法,java.lang.String 表明了该方法需要传递一个类型为 String 的参数。

5.4 赋予用户权限

当存储过程创建完毕后,用户就可以调用该存储过程。但是一般情况下,存储过程只能由 DBA 和创建者本人调用,其它用户如果要具备执行存储过程的权限,只能由 DBA 赋予权限。一旦用户具有了调用存储过程的权限,即便他没有访问该存储过程所依附的底层数据库对象(如表)的权限,用户仍然可以运行该存储过程。

由于数据库逻辑备份和恢复涉及到文件的读写操作,因此,这里要赋予调用者读写文件和执行命令的权限。赋予用户权限的命令如下:

```
EXEC DBMS_JAVA.grant_permission('USER',  
'java.io.FilePermission', '<<ALL FILES>>',  
'read,write,execute,delete');
```

```
EXEC Dbms_Java.Grant_Permission('USER',  
'SYS:java.lang.RuntimePermission', 'writeFile-  
Descriptor,');
```

```
EXEC Dbms_Java.Grant_Permission('USER',  
'SYS:java.lang.RuntimePermission', 'readFile-  
Descriptor,');
```

5.5 远程调用实现备份与恢复

服务器端存储过程 RunHostCommand 建立完

毕后，客户端应用程序中便可以通过调用存储过程实现向服务器端发送 EXP/IMP 命令完成 Oracle 数据库的逻辑备份与恢复功能。在客户端实际开发中，主要经历以下几个步骤：

① 加载 JDBC 驱动程序。要通过 JDBC 连接数据库，应用程序必须加载合适的 JDBC 驱动程序，可以使用 Class.forName 方法注册驱动程序。

② 建立数据库连接，获得连接对象。建立数据库连接可使用 DriverManager.getConnection 方法建立一个 Connection 对象。

③ 通过 CallableStatement 接口调用存储过程。如果存储过程有输入参数，可以调用 setxxx 方法设置(IN 参数)；当存储过程有输出参数时，必须用 registerOutPuter 方法注册输出参数，存储过程执行以后，再使用 getxxx 方法获得输出参数的值。

5.6 应用案例



图 2 Oracle 数据库中探矿权信息记录



图 3 单条探矿权信息记录的详细信息

在攀枝花矿产资源管理系统的实际开发过程中，

Oracle 数据库中存在大量的有关探矿权、采矿权、储量等的信息记录，如图 2 中所示的探矿权信息，同时每条信息记录中还包括的各种数据类型的字段，如图 3 中的勘察面积为数字型，有效期起止时间为日期型。

对于这样大规模的数据，为了实现远程逻辑备份与恢复功能，应用程序客户端需要向存储过程传入相应的 EXP/IMP 命令。作为示例，这里使用 EXP 命令调用 Oracle 数据库的 RunHostCommand 存储过程完成备份，其中通过文件选择器实现了向 setString 方法传入了内容为 “exp admin/cdut@PZH file=E:\Backup\back.dmp” 的 String 字段。当服务器端备份成功完成后，会传送给客户端相应的指令。实践证明，客户端调用该存储过程能够完成 Oracle 数据库的远程逻辑备份与恢复。

6 结语

在以 Oracle 为后台数据库平台开发应用系统时，可以充分利用 Java 存储过程功能来实现某些复杂功能。本文建立的存储过程不仅可以完成远程逻辑备份与恢复的功能，还调用服务器端的本地命令或执行其它命令，这为客户端实现在网络环境下复杂应用提供了很好的解决方案。

参考文献

- 张凡,田斌,周晶晶.Oracle9i 中 Java 存储过程的应用.微机发展, 2005,6:45 - 47.
- 张玉新,滕桂法,韩俊玲,王芳,赵洋,李阅历.存储过程及其在基于 J2EE 的选课系统中的实现.河北农业大学学报, 2007,(1):109 - 112.
- 王健. Oracle 数据库的备份与恢复策略研究.现代情报, 2007,(4):158 - 160.
- Tim. Shell Commands From PL/SQL[2009-07-29]. [http:// www.oracle-base.com/articles/8i/Shell CommandsFromPLSQL.php](http://www.oracle-base.com/articles/8i/Shell_CommandsFromPLSQL.php).