

# 基于 DM4 的 Java 存储过程的研究

冯玉才 胡 泉 班鹏新  
(华中科技大学计算机学院 武汉430074)

**摘 要** 本文对拥有自主知识产权的国产分布式数据库管理系统 DM4 进行研究,探讨了如何实现其对 Java 存储过程的支持。

**关键词** Java 存储过程,DM4,SQL/JRT 标准,虚拟机,代理服务器

## Research on the Stored Procedures of Java Based on DM4

FENG Yu-Cai HU Quan BAN Peng-Xin

(School of Computer, Huazhong University of Science and Technology, Wuhan 430073)

**Abstract** This paper researches about the DM4 which has independent copyright and domestic distribute DMS. And it finds out how to use the DM4 to realize the stored procedure of Java.

**Keywords** Stored procedures of Java, DM4, SQL/JRT standard

## 1 引言

今天的企业级应用程序通常运行在多数数据库平台上,在这种环境下,代码的可移植性和可维护性正变得越来越重要。由于数据库供应厂商都使用他们自己专有的过程语言,从一个数据库平台到另一个数据库平台的代码移植是非常复杂的,而且还要花费大量时间和人力物力。为了解决这些问题,居于领导地位的数据库供应厂商,例如 Oracle 和 IBM,都在他们的数据库和其他的应用程序平台上提供了 Java 存储过程机制。

Java 存储过程和普通存储过程一样,是由用户编写的保存在数据库中的程序,可以被客户应用程序或者另一个存储过程/函数通过引用它的名字而调用。它拥有普通存储过程所不具备的可移植性,它利用了 Java 所承诺的统一的、可移植的应用软件开发解决办法和 Java 语言强大的数学计算能力,为用户提供了一种更高效率的编程手段。本文在拥有自主知识产权的国产分布式数据库管理系统 DM4 的基础上,讨论了如何设计和实现 Java 存储过程。

## 2 Java 存储过程机制的优点

Java 存储过程机制允许用户使用 Java 语言来编写存储过程,它除了拥有普通存储过程的众多优点之外,还具有下列的优点。

### 2.1 更强大的数学计算和程序设计能力

SQL 语言可以完成简单的数学计算,而由于 Java 语言本身强大的数学计算能力,能够提供大量的数学操作符和数据类型,因此在大量的数学计算中,Java 存储过程有明显的性能优势。

### 2.2 适用范围广

Java 程序可以在一个复杂的应用程序的任何一层上配置:在客户端上,在中间层的应用程序服务器上或者在数据库服务器本身中,Java 存储过程也拥有这样广泛的适用范围。

### 2.3 兼容性、可移植性强

由于 Java 存储过程运行在服务器的内置 JVM 上,因此可以兼容各种支持 Java 存储过程的数据库,例如:Oracle、DB2、Sybase、Informix 等。应用程序和 Java 存储过程也不需要作任何的修改,就可以移植到以上这些数据库上,因此拥有很高的兼容性和可移植性。

## 3 Java 存储过程模块的功能和实现

Oracle 的存储过程使用的 PL/SQL 语言,它包括一整套的数据类型、数据结构、循环结构和异常处理结构。PL/SQL 可以创建过程和函数,以加速模块代码的开发,PL/SQL 和 SQL 紧密联系为一体,PL/SQL 程序中执行 SQL 语句,SQL 语句中可以调用 PL/SQL 函数。

DM4 系统的 Java 存储过程模块不同于 Oracle 的 PL/SQL 设计,在 DM4 服务器存储过程机制的上进行了修改和扩充,引入了虚拟机和代理服务器的设计,如图 1 所示。它针对 Java 存储过程所需的 SQL 语法和存储过程字典表进行了扩展,新增了服务器内置的 JVM、JDBC 和代理服务器保证 Java 存储过程运行时 DBMS 的安全。Java 存储过程机制主要包括:存储过程的使用、存储包、权限问题和预编译技术。以下着重对使用 Java 存储过程代码、虚拟机和代理服务器等模块进行讨论。

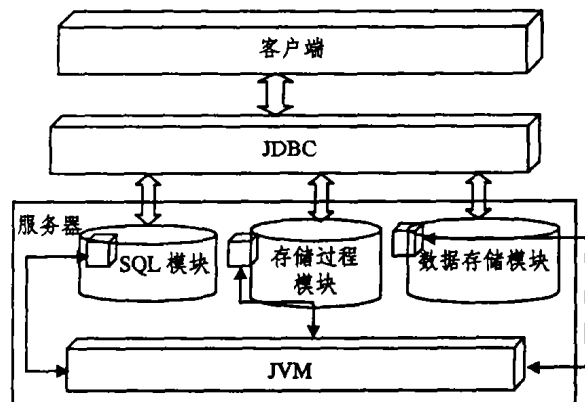


图1 Java 存储过程系统总体框架

### 3.1 注册 Java 存储过程代码

注册 Java 存储过程代码的过程实际上是将用户编写的 Java 及其相应的类存放至服务器,有两种方法可以注册 Java 存储过程的代码到 DBMS:一种是使用 LoadJava 工具,另一种则是使用扩展的语法“create java…”和“alter java…”进行注册。如图2所示,处理注册的具体过程如下:

- (1)对于 Java 文件,调用服务器内置的 JVM 编译成 class 文件或 jar 文件;
- (2)检查用户注册的 Java-class 名,排他锁封锁 Java-class 字典块,检查并修改全局字典,注册代码名、创建者、权限等信息;
- (3)将 class 文件和 jar 文件存放至服务器;
- (4)解锁,返回消息。

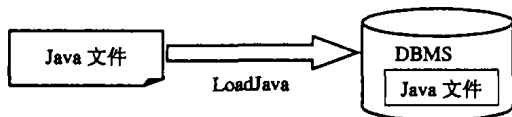


图2 LoadJava 装载 Java 类

### 3.2 创建 Java 存储过程

参考 SQL/JRT 规范,引入创建 Java 存储过程的语法“create procedure…external name…language java parameter style java”,并扩展全局字典表 SYSTEM.SYSDBA.SYSPROCS,创建的过程如图3所示,概括为如下的步骤:

- (1)接收创建 Java 存储过程的消息;
- (2)调用词法分析程序得到存储过程名、注册的 java 存储过程代码名等信息;
- (3)排他锁封锁 SYSTEM.SYSDBA.SYSPROCS 等全局字典,修改字典信息,注册存储过程名、存储过程代码和创建用户,使用权限等;
- (4)解锁返回消息。

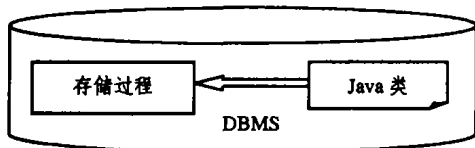


图3 创建 Java 存储过程调用说明

### 3.3 调用 Java 存储过程

DM4的服务器处理 Java 存储过程的调用可以分成3个部分,分别由服务器、代理程序和虚拟机组成,具体结构如图4所示。

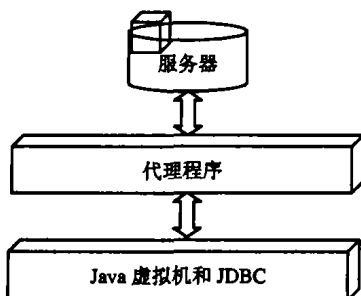


图4 Java 存储过程调用模块的结构

- (1)服务器接收调用 Java 存储过程的消息,进行语法词法分析;

- (2)查询字典表,找到指定的 Java 存储过程代码;

- (3)通知代理程序,调用指定的 Java 存储过程代码;

- (4)代理程序启动内置虚拟机 JVM 和内置 JDBC,执行 Java 存储过程,处理执行结果,返回消息。

3.3.1 虚拟机 服务器内置的 Java 虚拟机和 JDBC 构造了一套 Java 存储过程的运行环境。在 DM4中,Java 存储过程的代码以字节码的方式存放在服务器中,由于 JVM 的操作系统无关性,保证了不同操作系统的 DM4服务器的相互移植。每一个 Java 存储过程都申请一个单独的 JVM 来保证程序的安全执行,不会因为某一个存储过程意外地对虚拟机造成破坏性的影响后,导致所有的 Java 存储过程都无法正常执行。内置的 JDBC 则使更多的网络通讯本地化,节省 Java 存储过程的执行时间。一个需要执行  $N$  条 SQL 语句的 Java 程序,不使用内置的 JDBC 时需要和服务器进行  $2N \sim 4N$  次网络通讯,在使用内置的 JDBC 和 Java 存储过程后,仅需要4次网络通讯就可以得出结果,其余的网络通讯全部在服务器本地完成。

3.3.2 代理服务器 由于 Java 程序的功能强大,直接在服务器上调用 JVM 执行 Java 存储过程,可能会对服务器造成破坏性的影响,因此需要设计一个代理程序限制 Java 存储过程的执行。当 Java 存储过程中执行对 DBMS 有破坏性的操作,或者 Java 存储过程本身有错误产生时,代理服务器会处理这些操作使其不影响服务器的稳定性。

另外,当执行远程存储过程调用时,需要将远程存储过程代码加载到本地服务器上,然后在本地服务器上执行。如图5所示,在服务器 A 远程调用服务器 B 上的存储过程,先将存储过程的代码发送至服务器 A,再在 A 执行存储过程。由于 Java 存储过程的代码通常比较大,并且存储过程所需要的数据仍然在服务器 B 上,这样会产生大量的中间数据。

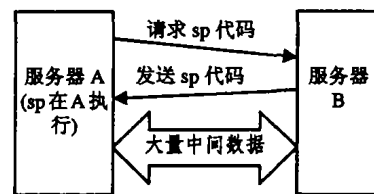


图5 远程调用存储过程(一)

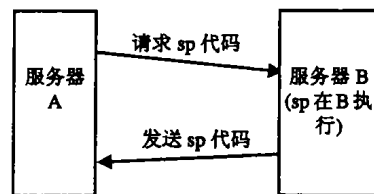


图6 远程调用存储过程(二)

在设计了代理程序后,Java 存储过程的执行和服务器的执行分开,因此可以将执行方式改成如图6所示的方式,服务器 A 发送远程执行存储过程的请求,存储过程在服务器 B 上完成,再将结果发送给服务器 A,节省了网络通讯的时间,网络数据流量也将大大减少。

结束语 随着企业信息系统应用的升级及电子商务的发展,Java 以其无比的优势将得到越来越广泛的应用。而 Java 存储过程因其简单、高效、易于维护及众多厂家支持且与平台无关的众多特性,其应用前景无比广阔。本文首次在拥有自主

产权的国产分布式数据库管理系统 DM4 上探讨了如何实现 Java 存储过程,并首次实现了 Java 虚拟机和 JDBC 驱动程序的内置,使数据库管理系统更好地从内部支持 Java 语言。但该系统尚存在不足之处,如何实现 Java 存储过程的分布式处理和虚拟机的安全共享内存执行,尚有一段路要走。

## 参考文献

- 1 Louden K C. Compiler Construction Principles and Practice. PWS Publishing Company, 1997
- 2 Aho A, Jeffrey V, Ullman D. Principles of Compiler Design. Addison-Wesley, 1977
- 3 Ertl M A. Stack Caching for Interpreters. In: Proc. of the ACM SIGPLAN Conf. on Programming Language Design and Implementation (PLDI'95), ACM Press, June 1995. 315~327
- 4 Romer T H, Lee D, Voelker G M, Wolman A, Wong W, Baer J-L, Bershad B N, Levy H M. The Structure and Performance of Inter-

preters. ASPLOS VII, 1996

- 5 Proebsting T A. Optimizing an ANSI C Interpreter with Superoperators. In: Proc. Symp. on Principles of Programming Languages, 1995. 322~332
- 6 Miranda E. BrouHaHa---a portable Smalltalk interpreter. In: Proc. of the Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'87), ACM Press, Oct. 1987. 354~365
- 7 SQLJ reference information site. <http://www.sqlj.com>
- 8 Oracle JDeveloper information. <http://www.oracle.com/tools/jdeveloper/suiteintro2.html>
- 9 Information on IBM's SQLJ implementation. <http://www.ibm.com/java>
- 10 Sun Microsystems Inc. The Java virtual machine specification. 2<sup>nd</sup> ed. [EB/OL]. <http://java.sun.com/docs/books>, 1999-09-01/2004-04-25

(上接第10页)

区。节点的状态信息量与覆盖网络控制算法相关,通常,基于集中式控制和泛洪方式的组管理的状态信息量较大,维护网络所需的通信过载也较大;而层次型结构和基于局部知识的组管理节点状态信息量较少,控制也相对简单。2)组播会话个数,该参数主要与组播树的构建方法相关。对共享树而言,所有数据源都通过该组播树发送数据,不需为每个会话维护单独的状态信息;而单独树由于要为每个数据源建立独立的数据分发树,节点必须为不同会话维护相应的状态信息,因此关于会话的状态信息与会话个数成正比。但从传输效率来看,由于单独树可以为每个发送源定制树,可以创建具有较小数据传输延迟的优化树。

从协议构造方式来看,网格优先协议由于创建的是单独树,且有较高的状态维护要求,故较适合小规模组播应用。而隐式协议一般都基于结构化设计,因而有较强的可伸缩能力,非常适合大规模应用。

**总结** 应用层组播技术将组播功能实现于应用层,避免了 IP 组播对网络基础设施的要求,解决了 IP 组播应用所面临的缺陷(如路由技术、组播地址空间),为组通信应用提供了一种新的解决途径。协议设计借鉴了 IP 组播的数据通信模型,节点间根据网络条件和组成员关系自组织成覆盖网络,并根据系统的动态变化作出自适应调整,为大规模动态环境(如 P2P 应用)提供了有效的数据传输服务。应用层组播的协议实现可集成现有网络设施对 IP 组播的支持,但大规模应用不要求网络层提供这些额外支撑,易于在 Internet 环境下部署。

应用层组播适应了新型分布式计算环境特别是 P2P 网络的需要。近年来,相关研究主要针对系统的动态性,结合协议设计的性能要求提出了一些解决方案,为相关技术的广泛应用提供了基础。进一步工作还应针对应用层的实现方式从软件部署方法、服务质量、系统安全等方面来增强和完善现有工作。

## 参考文献

- 1 Deering S. Multicast Routing in Internetworks and extended LANs. In: Proc. of ACM SIGCOMM. Aug. 1988
- 2 Saltzer J, Reed D, Clark D. End-to-end Arguments in System Design. ACM Transactions on Computer Systems, 1984, 2(4): 195~206

- 3 Francis P. Yoid: Extending the Internet Multicast Architecture. April 2000, White Paper. <http://www.aciri.org/yoid>.
- 4 Bannerjee S, Bhattacharjee B, Kommareddy C. Scalable Application Layer Multicast. ACM SIGCOMM, Aug. 2002
- 5 Pendarakis D, Shi S, Verma D, Waldvogel M. ALMI: An Application Level Multicast Infrastructure. In: Proc. of 3<sup>rd</sup> Usenix Symposium on Internet Technologies & Systems, March 2001
- 6 Ratnasamy S, Handley M, Karp R. Application-level Multicast using Content-Addressable Networks. In: Proc. of 3<sup>rd</sup> Intl. Workshop on Networked Group Communication, Nov. 2001
- 7 Zhuang S Q, Zhao B Y, Joseph A D, Katz R H, Kubiawicz J D. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination. In: Eleventh Intl. Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV 2001), 2001
- 8 Chu Y, Rao S G, Seshan S, Zhang H. A Case for End System Multicast. In: the Proc. of ACM SIGMETRICS, June 2000
- 9 Castro M, Druschel P, Kermarrec A-M, Rowstron A. SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications (JSAC), 2002
- 10 Chawathe Y. Scattercast: An Architecture for Internet Broadcast Distribution as an Infrastructure Service. [Ph. D. Thesis]. University of California, Berkeley, Dec. 2000
- 11 Leibeherr J, Nahas M. Application-layer Multicast with Delaunay Triangulations. In: Global Internet Symposium, Globecom, Nov. 2001
- 12 Zhang B, Jamin S, Zhang L. Host multicast: A framework for delivering multicast to end users. In: Proc. of IEEE INFOCOM 2002, New York, NJ, USA, June 2002
- 13 Finlayson R. The UDP Multicast Tunneling Protocol. work in progress, Draft-finlayson-umtp-07. txt, Sept. 2002
- 14 Roca V, El-sayed A. A Host-based Multicast (hbm) Solution for Group Communications. In: 1<sup>st</sup> IEEE Intl. Conf. on Networking, Calmar, France, July 2001
- 15 El-Sayed A, Roca V, Mathy L. A Survey of Proposals for an Alternative Group Communication Service. IEEE Network, Jan. / Feb. 2003
- 16 Bannerjee S, Bhattacharjee B. A Comparative Study of Application Layer Multicast Protocols. <http://www.cs.umd.edu/users/suman/publications.html>
- 17 Tran D A, Hua K a, do T T. A Peer-to-Peer Architecture for Media Streaming. To appear in IEEE JSAC Special Issue on Advances in Overlay Networks