

基于J2EE的数据库连接服务

黄文* 谢寄石

(电子科技大学机械电子工程学院 成都 610054)

【摘要】讨论了数据库连接服务体系结构,并详细地阐述了既支持连接池,又支持分布式交易的连接管理模块的规范设计以及功能实现。该设计不仅使应用代码具有较好的可移植性,而且能够在很大程度上改善应用服务器的整体性能,可为商务应用高效、透明地访问各种关系数据库管理系统提供一种更优的选择方案,特别适合分布式计算环境下的多用户并发访问。

关键词 连接池; 分布式交易; 本地交易; 可交易连接; 可交易资源

中图分类号 TP311.13; TP399

随着电子商务的迅猛发展,应用服务器的使用越来越广泛,所有基于Internet的WEB应用和EJB应用均可采用它来开发,此类应用通常需要访问数据库,而且可能需要同时访问多个不同的数据库,为了支持数据库访问,应用服务器通常集成有数据库连接服务。

作为应用服务器一个必不可少的系统级服务组件,数据库连接服务一般由服务管理模块和连接管理模块构成。服务管理模块构建于连接管理模块之上,应用服务器可用它来管理各种数据源,包括数据源的配置、初始化以及到名字服务中的注册,不过其设计在很大程度上取决于应用服务器的服务组件体系结构。连接管理模块是数据库连接服务的核心,可分为应用接口实现、连接池和分布式交易三个功能块^[1~3],应用服务器通过连接管理模块来透明地访问各种关系数据库管理系统,其设计必须遵守一定版本的JDBC规范。本文主要讨论分布式计算环境下连接管理模块的规范设计以及功能实现。

1 应用接口

为了访问数据库,JDBC应用组件JSP、Servlet或EJB可使用驱动程序管理器DriverManager来建立物理连接^[4]。首先调用Class.forName方法装载特定厂商的Driver实现,以便将其一个实例注册到DriverManager中,然后再调用DriverManager.getConnection方法来创建物理连接。但这种方式弊端较多,首先Driver实现标识一个特定的驱动程序供应商,其次数据库服务器信息,例如用户名、口令以及URL完全硬编码到程序中。这两种情形均不利于程序代码的维护和数据库服务器的安全管理,因此JDBC应用组件使用这种方法来访问数据库并不合适。而连接管理模块为数据库访问提供了一个统一的、高度抽象的数据源接口DataSource,通过该接口,JDBC应用组件只需调用getConnection方法就可获得连接。在启动应用服务器时,由于数据库连接服务管理模块读取数据源配置信息并通过JNDI名字服务将底层的关系数据库管理系统映射到一个逻辑名上,因此JDBC应用组件可直接使用逻辑名来访问数据库,另外数据源配置信息,例如用户名、口令或者URL的改变并不影响应用组件的正确运行,也无须修改任何应用代码,故使用DataSource接口不仅可避免程序中的硬编码,而且能够确保应用代码具有较好的可移植性。

2 数据库连接池

数据库服务器在为新连接分配内存资源的同时还要进行一系列操作,例如校验网络协议、验证

2001年9月24日收稿

* 男 28岁 硕士生

用户身份、为新连接设置安全上下文和交易上下文。这些操作不仅耗时,而且很浪费系统资源。为了提高应用服务器多用户并发访问效率和缓解数据库服务器压力,在设计连接管理模块时,通常要实现连接池,使用其来缓存已创建的物理连接以便在适当的时候重用,从而缩短连接建立时间并提高数据库访问效率。

为了支持数据库连接池,JDBC2.0标准扩展API定义了如下接口:

```
javax.sql.ConnectionPoolDataSource  
javax.sql.PooledConnection
```

其中连接池数据源ConnectionPoolDataSource是池化连接PooledConnection对象工厂,其对象引用通常注册在JNDI名字服务中,而池化连接代表真正的物理连接,调用连接池数据源的getPooledConnection方法可获得池化连接。

3 分布式交易

和数据库连接服务一样,交易服务也是应用服务器的一个系统级服务组件,应用服务器可使用它来划分交易边界、传播交易上下文、同步交易和管理可交易资源。由于多个关系数据库管理系统可能同时参与分布式计算,因此应用服务器需要交易服务来管理、协调分布式交易下的多个数据库管理系统之间的关联,并对所有已登记资源执行两阶段提交以确保交易的原子性、完整性、一致性和持久性。

为了支持分布式交易,JDBC2.0标准扩展API和JTA1.0规范定义了如下接口:

```
javax.sql.XADataSource  
javax.sql.XAConnection  
javax.transaction.xa.XAResource
```

其中XA数据源XADataSource是可交易连接XAConnection对象工厂,其对象引用通常注册在JNDI名字服务中,调用getXAConnection方法可获得可交易连接,可交易连接是参与分布式交易的池化连接,扩展了PooledConnection接口,并添加了一个新的方法getXAResource以获取可交易资源XAResource。XAResource是X/Open XA接口的JAVA映射,交易服务使用它来登记参与分布式交易的可交易连接,并将可交易连接和分布式交易关联起来,从而把数据库服务器所管理的本地交易委派给交易服务管理,由交易服务负责分布式交易的提交和回滚。交易服务始终通过可交易资源来操作可交易连接,而不直接与可交易连接交互。

3.1 分布式交易和可交易连接的关联

分布式交易通过XAResource.start方法与可交易连接关联,并通过end方法与之分离。由于可交易资源XAResource不支持交易嵌套,因此在任何时候可交易连接都只能与一个交易关联,仅当XAResource.start和XAResource.end方法能够在多个交易上下文之间正确切换时,可交易连接才可能为多个不同的交易共享。下面通过举例来说明分布式交易是如何实现可交易连接共享的,其代码如下:

```
// 假设已获得一个可交易连接对象xac  
XAResource xares = xac.getXAResource();  
xares.start(xid1); // 关联分布式交易xid1和可交易连接xac  
..  
xares.end(xid1); // 分离分布式交易xid1和可交易连接xac  
..  
xares.start(xid2); // 关联分布式交易xid2和可交易连接xac  
..
```

```

// 当分布式交易xid2和可交易连接xac关联时, 交易服务对已分离的分布式交易xid1执行两阶段提交
//
Status status = xares.prepare(xid1);
..
xares.commit(xid1, status)

```

3.2 分布式交易下各接口之间的关系

如图1所示, 调用XADatasource.getXAConnection方法能够获取可交易连接XAConnection, 而可交易连接在其内部维护着可交易资源XAResource和逻辑连接句柄Connection两个实体。调用XAConnection.getXAResource方法可获取可交易资源, 交易服务使用它来管理分布式交易; 调用XAConnection.getConnection方法可获取逻辑连接句柄, JDBC应用组件使用它来透明地访问各种关系数据库管理系统。

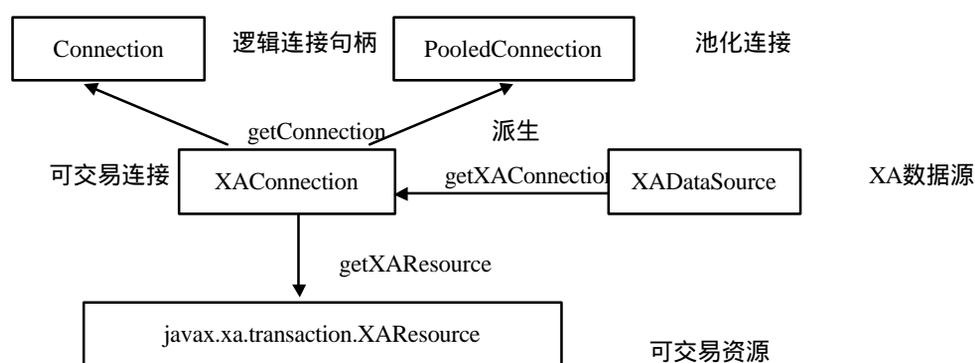


图1 分布式交易下各接口之间的关系

3.3 可交易连接的缓存

可交易连接XAConnection是参与分布式交易的池化连接, 可缓存到数据库连接池中, 设计时常采用JAVA事件处理技术来实现缓存。在这里, 可交易连接被定义为事件源, 连接事件一旦发生, 就通知其注册监听器去处理相应事件, 而实现连接事件监听器接口ConnectionEventListener的类被称为事件监听器, 调用addConnectionEventListener方法可注册一个新的事件监听器, 而连接事件对象ConnectionEvent才是事件源和其注册监听器进行交互的桥梁。当JDBC应用组件调用Connection.close方法时, 逻辑连接句柄就将连接关闭请求转发给事件源, 再由事件源通知其注册监听器调用connectionClosed方法响应连接关闭请求, 同时将XAConnection缓存到连接池中, 从而实现可交易连接的缓存。

4 连接管理模块功能实现

根据规范可知, 为了支持数据库连接池和分布式交易, 各驱动程序供应商应当实现如下接口:

```

javax. sql. DataSource
javax. sql. ConnectionEventListener
javax. sql. ConnectionPoolDataSource
javax. sql. PooledConnection
javax. sql. XAConnection
javax. sql. XADatasource
javax. transaction. xa. XAResource

```

实际上仅有JDBC2.0驱动程序实现了上述接口,而未实现这些接口的JDBC1.0驱动程序并不支持数据库连接池和分布式交易,因此在设计连接管理模块时,通常可将其细分为三层,自上而下为:应用接口实现、连接池以及JDBC2.0伪驱动程序。其中,JDBC2.0伪驱动程序位于各JDBC1.0驱动程序之上并实现了上述接口的标准包装,以支持分布式交易;数据库连接池位于伪驱动程序之上,以缓存可交易连接;应用接口实现在管理连接池和分布式交易功能的同时还为商务应用访问各种关系数据库管理系统提供了一个统一的、高度抽象的数据源接口。连接管理模块还可对某些JDBC2.0驱动程序(例如Oracle)进行特定包装,以便更好地利用现有资源。

5 连接请求流

假设数据源对象在实现DataSource接口的同时还实现了ConnectionEventListener接口,而且注册为事件源XAConnection的监听器。基于上述假设,在分布式计算环境下连接管理模块响应JDBC应用组件所发出的连接请求流程如下:

1) JDBC应用获取JNDI名字服务初始上下文InitialContext,并调用lookup方法查找数据源DataSource并将其对象引用返回给应用;

2) JDBC应用调用DataSource.getConnection方法获取逻辑连接句柄Connection;

3) DataSource对象校验连接池中是否缓存有可交易连接XAConnection。如果有,则连接池更新其内部数据结构并返回一个可重用的可交易连接对象给DataSource;如果没有,连接池就调用XADataSource.getXAConnection方法新建一个可交易连接对象,然后再更新其内部数据结构并将新建的可交易连接对象返回给DataSource;

4) DataSource对象调用XAConnection.getXAResource方法获取可交易资源XAResource,并将获取的可交易资源注册到交易服务中;

5) 交易服务调用XAResource.start方法启动可交易连接和分布式交易的关联;

6) DataSource对象调用XAConnection.getConnection方法获取逻辑连接句柄Connection,并将其返回给JDBC应用;

7) JDBC应用使用逻辑连接句柄来访问数据库。由于该对象是在分布式交易下使用,因此不应当对它显式调用commit、rollback或者setAutoCommit等交易操作;

8) JDBC应用调用Connection.close方法发出连接关闭请求;

9) 逻辑连接句柄将连接关闭请求转发给事件源XAConnection;

10) 事件源通知其注册监听器响应连接关闭请求;

11) DataSource对象通知交易服务解除已登记的可交易资源;

12) 交易服务调用XAResource.end方法结束可交易连接和分布式交易的关联;

13) DataSource对象将XAConnection缓存到连接池中以待重用。

6 结束语

目前,应用程序开发者为了提高数据库访问效率常使用本地连接池来缓存物理连接,但是这种技术有以下几点不足:1) 应用代码移植性差、难维护;2) 不支持分布式交易;3) 应用越复杂,系统资源占用越多,其执行效率也逐渐降低^[5,6]。本文所述的数据库连接服务为应用访问各种关系数据库管理系统提供了一个统一的、高度抽象的应用接口,同时还实现了连接池和分布式交易,因此它完全克服了传统数据库访问方法的缺陷,不仅适用于本地以及分布式计算环境下的多用户并发访问,而且具有良好的系统扩展性和伸缩性,可大大优化软件的整体性能。

参 考 文 献

- 1 Seth White, Mark Hapner. JDBC 2.1 standard API. <http://java.sun.com/products/jdbc>, 1999
- 2 Seth White, Mark Hapner. JDBC2.0 standard extension API. <http://java.sun.com/products/jdbc>, 1999
- 3 Susan Cheung, Vlada Matena. JTA1.0 specification. <http://java.sun.com/products/jta>, 1999
- 4 霍斯曼, 柯内尔著. Java 2核心技术 卷 II: 高级特性. 朱志, 王怀, 赵伟等译. 北京: 机械工业出版社, 2000
- 5 周西京. 基于Java的Internet上数据库存取及实现. 电子科技大学学报, 1999, 28(2): 202-206
- 6 吴跃, 余水, 傅彦, 等. Internet数据库访问技术. 电子科技大学学报, 2001, 30(1): 58-61

Database Connection Service Based on J2EE

Huang Wen Xie Jishi

(College of Mechanical and Electrical Eng., UEST of China Chengdu 610054)

Abstract In this paper the architecture of database connection service in application server is discussed. The standard design and function implementation of connection manager module which supports both connection pool and distributed transaction is illuminated in detail. The design not only makes applications more portable, but also greatly improves the performance of application server. A good choice is provided for efficient and transparent accessing all kinds of relation database management system. It is very fit for concurrent multiuser access in distributed computing environment.

Key words connection pool; distributed transaction; local transaction; transactional connection; transactional resource

• 科研成果介绍 •

激光全息照相新技术研究

主研人员: 王仕璠 刘艺程 莉

激光全息照相新技术紧密结合近年来市场对高质量激光全息防伪标识、商标的需求, 提出了拍摄彩虹全息图的一系列新方法, 采用多种加密技术提高了全息记录的防伪功能, 针对大视场、高亮度的全息记录对光场均匀性的要求, 提出了新的主全息图制作方法、大视场记录技术和叠合高斯光场技术, 并从理论上给予证明, 还对全息图像的本质、准深全息图的联想功能等进行了深入的研究, 并把全息照相术用于地质构造相似模型的无损检测。

• 甬 江 •