基于 JMS 和 MDB 构建异步日志服务

Developing an asynchronous log service based on JMS and MDB

陈智罡 （浙江万里学院计算机学院 315100）
王 磊 （清华同方光盘股份有限公司 100084）
高敬惠 （河北工业大学计算机科学系 065000）

摘要：日志服务是软件中的一个重要组成部分。同步的日志服务会带来额外的开销，而且不能有效地作为分布式日志服务。本文介绍如何使用 JMS 和 MDB 开发一个异步日志服务。其中使用 JMS 和 MDB 实现异步，使用 JDBC 来持久化数据。

关键词：Java 消息服务 消息驱动 Bean JDBC 异步日志服务

1 引言

当今有许多日志框架可以使用，但它们大部分都是在 JMS 问世前开发的，都没有使用异步日志。在传统的同步日志模型里，当日志服务调用没有成功返回前，调用者是不能往下执行的，显然这样会带来额外的开销。分布式计算环境中，客户端是并发运行的，这种情况下记录日志到中心数据库是相当烦琐的。本文将开发一个异步日志服务，日志信息通过网络发到 JMS 提供者，然后 MDB 提取出信息通过 JDBC 持久到数据库。

2 JMS 和 MDB 介绍

2.1 JMS

JMS（Java Message Service）是应用于组件与组件或应用系统与应用系统之间通信的一种技术[1]。它支持分布式通信，是一种松耦合结构。简单的工作机制是：一个发送者（Sender）将消息（Message）发送到一个目的地（Destination），另一个接收者（Receiver）就可以从这个目的地得到这个消息。在发送者和接收者之间存在一个目的地，这个目的地充当发送者和接收者之间的桥梁完成通信，而不是发送者和接收者直接建立连接。正因为如此，发送者和接收者不用知道对方的信息，只需要知道消息的目的地和消息的格式。

JMS 有两种消息的传送模式：点对点（point－to－point）方式和发布/订阅（publish/subscribe）方式。本文采用的是点对点的消息传送方式，在这种方式里发送者将每条消息发送到一个指定的消息队列，接收者从指定的消息队列中获取消息。

2.2 MDB

EJB 2.0 规范定义了一种新的 EJB 类型，即消息驱动 Bean（Message－Driven Bean，简称 MDB），它能够以 EJB 的形式实现 JMS 消息的接收者。消息驱动的 EJB 实现一组新的接口，这组接口使得 EJB 能够异步地接收和处理 JMS 消息生产者发送到队列或话题的消息[3]。EJB 客户程序的构造方式与普通 JMS 消息生产者的构造方式完全一样，也就是说，JMS 消息生产者不必知道消息的消费者是一个 EJB。

相对于会话 Bean 和实体 Bean 而言，消息驱动的 Bean 最大的特点是客户程序不通过接口访问 Bean。与会话 Bean 和实体 Bean 不同，消息驱动的 Bean 只有一个 Bean 类。会话 Bean 和实体 Bean 可以发送 JMS 消息并同步接收 JMS 消息，而不是异步方式。

3 异步日志服务的功能框架

异步日志服务的工作流程是：客户端创建一个日志消息，类型是 JMS 消息类型 ObjectMessage，于是在消息体内嵌入 logMesVO 对象，并将它发送给 LogMesQueue 目的地。当 JMS 消息到达时，容器选择一个 LogMDB 实例，执行其 onMessage 方法，并且将 ObjectMessage 作为一个参数传递给该方法。LogMDB 使用 logDAO 助手类执行相应的逻辑来提取和向数据库

插入消息的内容，完成日志的持久化工作。从中可以看出 JMS 采用的是点对点的方式，交互图如下：

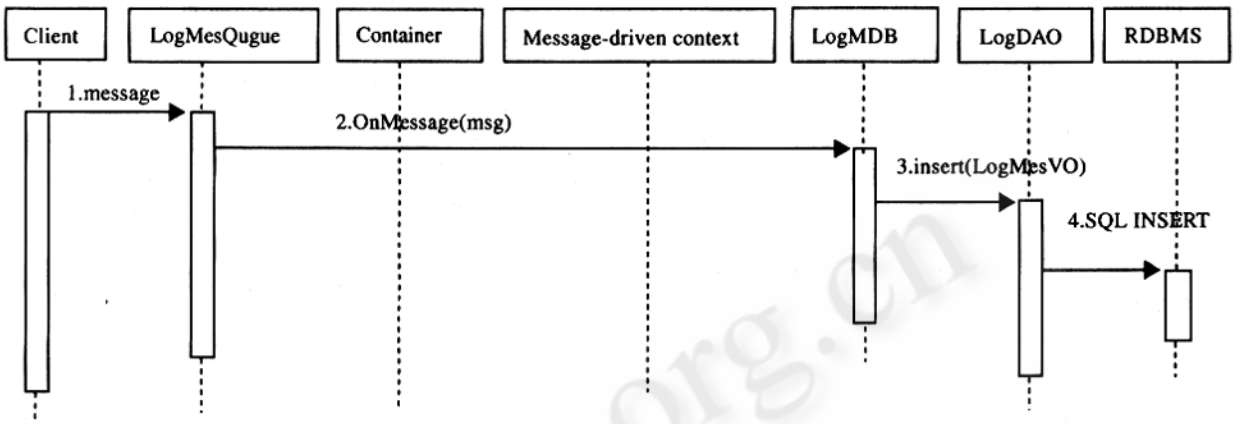


图 1　异步日志服务交互图

4　框架组件实现

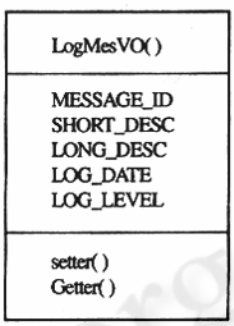
4.1　日志表格

首先创建一个存储日志信息的表，具体内容为：简单描述，详细描述，日志级别，日志时间，还需要有一个 ID，用于区别每条日志消息。

```
create table logTab
(
 MESSAGE_ID VARCHAR2(30) not null,
 SHORT_DESC VARCHAR2(20),
 LONG_DESC VARCHAR2(200),
 LOG_DATE DATE,
 LOG_LEVEL VARCHAR2(10)
)
```

4.2　logMesVO 类

logMesVO 是一个含有日志信息的 Java 对象，它被设计成值对象，用来持有日志信息。这个类与前面创建的表格相对应，每个属性对应表格的一列，同时还要为每个属性创建 set 和 get 方法。如下图：



2　logMesVO 类图

4.3　LogMDB 消息驱动 Bean

LogMDB 是一个消息驱动 Bean，它监听消息队列，一旦消息到达队列，EJB 容器就调用 MDB，访问其 onMessage 回调方法，利用其助手类 LogDAO，将消息插入到数据库。

OnMessage 方法如下：

```
Public void onMessage( javax. jms. Message mes-
sage){
    Try{
      ObjectMessage objMsg = (ObjectMessage)
      inMessage;
      LogMesVO logMsg = (LogMesVO) objMsg.
      getobject( );
      logDAO = new LogDAO( );
      logDAO. insert( logMsg);
    }catch( Exception) {
      t. printStackTrace( );
    }
  }
```

4.4　LogDAO 类

其功能为隔离 LogMDB 对数据库的直接访问，增

强了复用性。JNDI 查找用于在 LogDAO 的构造函数中访问数据源。LogDAO 的方法 getConnection( )、CloseConnection( )、closeResultSet( ) 和 closeStatement( ) 用于管理对数据源的连接。在从消息中提取 logMesVO 对象后，LogMDB 调用 LogDAO 的 insert( ) 方法，将其插入到数据库。LogDAO 使用 JDBC API 的 Prepared Statement 特性来创建 SQL 语句，使用 executeUpdate( ) 来运行 SQL 语句。主要代码如下：

```
Public void insert( logMesVO logMsg ) throws LogDAOException
    {
      PreparedStatement pstmt = null;
      Connection conn = this. getConnection( );
      Try
        {
          pstmt = conn. prepareStatement( "Insert into LogTab( MESSAGE_ID,
    SHORT_DESC, LONG_DESC, LOG_DATE , LOG_LEVEL ) values( ?,?,?,?,?) " );
          Pstmt. setString( 1, logMesVO. getMESSAGE_ID( ) );
          Pstmt. setString( 2, logMesVO. getSHORT_DESC( ) );
          Pstmt. setString( 3, logMesVO. getLONG_DESC( ) );
          Pstmt. setString( 4, logMesVO. get LOG_DATE ( ) );
          Pstmt. setString( 5, logMesVO. get LOG_LEVEL( ) );
          Pstmt. executeUpdate( );
        } catch( SQLException se ) {
        ......
        }
        ......
    }
```

4.5 消息驱动 Bean 的部署描述

部署描述是非常重要的，它告诉容器应该做什么，以下是消息驱动 Bean 的部署描述：

```
<message – driven >
....
<ejb – name > LogMDB </ejb – name >
<ejb – class > com. chzg99. JMSLoggerBean </ejb – class >
    <transaction – type > Container </transaction – type >
    <acknowledge – mode > Auto – acknowledge </acknowledge – mode >
    <message – driven – destination >
      <destination – type > javax. jms. Queue </destination – type >
<subscription – durability > NonDurable </subscription – durability >
    </message – driven – destination >
    <resource – env – ref >
      <description > Asynchronous Log Queue </description >
<reso, urce – env – ref – name > AsynchronousLogQueue </resource – env – ref – name >
<resource – env – ref – type > javax. jms. Queue </resource – env – ref – type >
    </resource – env – ref >
    </message – driven >
    ....
```

5 小结

JMS 的引入是为了解决应用程序客户端调用的“同步”和“异步”问题，而消息驱动 Bean 的引入则是为了集中解决消息的“异步”接受问题。异步消息服务原理可以被应用到许多的服务，比如属性服务、配置服务和其他的数据服务。当然在这里也可以创建一个日志浏览器来浏览持久到数据库的日志信息。持久化数据的方法很多，除了 JDBC 外，还可以选择 EJB 方式。

参考文献

1 Sun Microsystems, Inc. JavaTM2 Platform, Enterprise Edition Specification Version 1.3[EB/OL]. http://java. sun. com . 2000.

2 陈华军，J2EE 构建企业级应用解决方案[M]，人民邮电出版社，2002。

3 Pravin V. Tulachan . Developing EJB2. 0 Components [M]，肖国尊、马擎予译，清华大学出版社，2002。