

基于 B/S 模式的仓储管理信息系统的设计与实现

周 南 马云龙

(中国农业大学计算机网络中心)

摘 要 为解决跨地区集团级管理信息系统软件平台的需求问题,设计并实现了一个仓储管理信息系统。该系统采用基于WEB的B/s三层技术架构,利用Web服务器、应用服务器、Oracle数据库以及JDBC搭建企业平台,开发工具使用JSP和JavaScript。实际应用结果表明,可以实现信息的网上动态发布和实时交互。

关键词 仓储管理; B/S模式; JDBC; JavaBean; Oracle

中图分类号 TP 315

Design and Implement of a Storage Management Information System Based on B/S Mode

Zhou Nan, Ma Yunlong

(Computer and Network Center, China Agricultural University, Beijing 100083, China)

Abstract The design and realization of a storage management information system was developed and the software-platform application of multi-regional information management system was solved. The system, based on Browsers/Server 3-tier technique structure, constructed the company platform utilizing WEB server, Application server, Oracle database and JDBC. Furthermore, the dynamical publication and exchange of information in the Internet was realized using JSP and JavaScript and played a significant role in the development of company.

Key words Storage management; B/S; JDBC; JavaBean; Oracle

随着我国加入WTO,许多国内企业已强烈地感受到来自国外企业的竞争压力,这些竞争压力来自于国外企业高度的信息化,先进的管理经验和水平等。我国仓储行业作为一个传统产业,其信息化和现代化程度普遍落后于国内其他行业:目前大多处于手工记录,人工核算的阶段;集团与分公司之间没有实现网络互联,使得信息流的交换非常缓慢。在网络化、信息化飞速发展的今天,开发一套仓储管理信息系统作为软件应用平台,构建现代物流信息神经网络,形成跨地区性质的覆盖全国的网络化仓储管理体系具有现实意义。

1 系统方案的确定

长期以来,基于客户机/服务器(Client/Server)结构的软件系统占据应用软件的主导地位,但是,由于这种模式的系统中客户端需要应用程序运算数据处理逻辑,几乎所有的应用逻辑

收稿日期: 2001-11-03

周 南,北京清华东路17号中国农业大学(东校区)110信箱,100083

辑都在客户端实现。随着用户应用需求的提高,运算复杂程度的加深以及数据交换频率的加快,使得这种 C/S 模式的系统在版本升级,软件维护方面极为不便;另外,由于 C/S 模式的每个客户端都必须与数据库保持连接,限制了系统支持的用户数量。因此采用 B/S 模式三层体系结构的仓储管理信息系统作为软件应用平台。

2 系统结构的设计

B/S 模式是一种以 Web 技术为基础的新型的系统平台模式,它把传统 C/S 模式中的服务器部分分解为一个数据服务器和一个或多个应用服务器(Web 服务器),从而构成了三层结构的客户服务器体系^[1]。

第 1 层客户机是用户与整个系统的接口。客户的应用程序精简到一个通用的浏览器软件,如 Netscape Navigator, 微软公司的 IE 等。浏览器将 HTML 代码转化成图文并茂的网页,网页具备一定的交互功能,允许用户在网页提供的申请表上输入信息提交给后台第 2 层的 Web 服务器,并提出处理请求。

第 2 层 Web 服务器将启动相应的进程来响应这一请求,并动态生成一串 HTML 代码,将嵌入的处理结果,返回给客户机浏览器。如果客户机提交的请求包括数据的存取,Web 服务器还需与数据库服务器协同完成这一处理工作。

第 3 层数据库服务器的任务类似于 C/S 模式,负责协调不同的 Web 服务器发出的 SQL 请求。

综上,B/S 模式具有以下特点。

首先,B/S 模式体系简化了客户端。无须像 C/S 模式那样在不同的客户机上安装不同的客户应用程序,而只需安装通用的浏览器软件。这样不但可以节省客户机的硬盘空间和内存,而且安装过程更加简便,网络结构更加灵活。另外,它简化了系统的开发和维护,系统的开发者无须再为不同级别的用户设计开发不同的客户应用程序,而只需把所有的功能都实现在 Web 服务器上,并就不同的功能为各个组别的用户设置权限就可以了。各个用户通过 HTTP 请求在权限范围内调用 Web 服务器上不同的处理程序,从而完成对数据的查询或修改。在维护方面,相对于 C/S,B/S 的维护具有更大的灵活性。当形势变化时,它无须再为每一个现有的客户应用程序升级,而只需对 Web 服务器上的服务处理程序进行修订。这样不但可以提高企业的运作效率,还省去了维护时协调工作的不少麻烦。其次,B/S 模式体系使用户的操作变得更简单。对于 C/S 模式,客户应用程序有自己特定的规格,使用者需要接受专门培训。而采用 B/S 模式时,客户端只是一个简单易用的浏览器软件。无论是决策层还是操作层的人员都无须培训,就可以直接使用。最后,B/S 特别适用于网上信息发布,这是 C/S 所无法实现的。而这种新增的网上信息发布功能恰是现代物流企业所需的。这使得企业与客户之间的信息交流可以在 Internet 上直接进行,从而提高了企业的工作效率。本文中采用的 B/S 三层模式结构见图 1。

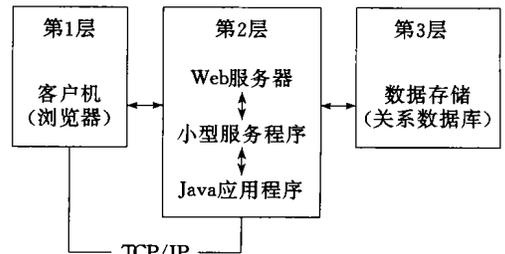


图 1 仓储管理信息系统的 B/S 三层模式结构

3 系统的开发过程

开发工具采用 JSP (Java Server Pages), 是一种由 JSP 容器解释的文本文档, 在 JSP 容器中, 包含了静态的 HTML, 用来动态产生针对 Web 请求的 Web 响应脚本语言命令, 并且用来产生与服务器上的应用程序和数据进行通信的脚本语言命令。JSP 最终将被转换和编译为 Java Servlet, 但是 JSP 比 Java Servlet 更容易开发。与 Java Servlet 一样, JSP 也使用 HTTP 作为缺省请求/响应的通信范例, 可以在 JSP 中定义 HTML 和 XML 以及其他模板数据, 并且在服务器端不需做任何处理就能够直接发送到 Web 客户上, 但是, 在请求页面被传递给 Web 客户之前, 插入到相同页码中的 JSP 脚本语言命令将会由服务器进行处理。JSP 脚本命令语言具有简单的、类似于 XML 的语法, 并且可以在 JSP 页面中使用 Java 代码, 大部分的代码可以在支持 JSP 的不同操作系统的 Web 服务器上运行, 符合“Write once, run anywhere”的 Java 标准。同时, 由于 JSP 技术具有平台和服务器互相独立的特性, 故能够集成各类服务器的组件和数据库产品^[2]。当 Web 服务器收到客户端浏览器对 JSP 页面的访问请求时, 页面中的 JSP 脚本将由 JSP 引擎转译成 Servlet, Web 服务器运行的各项操作, 如访问数据库等, 实质上都是执行 Servlet 的结果。当客户端浏览器首次访问 JSP 页面时, 由于将 JSP 脚本转译成 Servlet 的过程可能需要花费一定的时间, 因此速度相对比较慢, 但是如果多次访问该页面, 即使传递参数的数量或值以及返回结果不同, 响应的速度也会很快, 因为 JSP 引擎会保存转译后的 Servlet 以及与数据库的连接, 所以在频繁访问页面的情况下基本省去了转译及连接数据库的时间, 同时也减轻了服务器的负担, 提高了整个系统的性能。

在应用程序的代码编写过程中, 采用了预先写好的可重复调用的 Java 程序组件, 即 JavaBean 组件模型。JavaBean 与 ActiveX 控件一样, 提供已知的功能且允许重复使用, 如 JSP: `useBean id = "idname" class = "class name" scope = "page" /`。在实际应用中利用 JavaBean 可把常用的数据库功能都封装起来, 比如打开数据库, 关闭数据库, 执行查询, 插入操作等, 而在 JSP 中只需输出返回的结果集即可。这样做的好处是隐藏了复杂的操作, 加快了响应速度。例如, 对于数据库的中文问题, 笔者在 Bean 内定义一个函数, 执行汉字内码的转换。在 JSP 文件中, 可以直接把包含汉字的 SQL 语句传给 Beans, 而不必关心细节, Bean 会自动调用函数, 执行内码转换, 然后再进行数据库操作, 并把结果返回。这样比在每一个 JSP 文件中都要写一大堆的数据库访问代码要好得多, 开发中只需简单地声明一下该 Bean, 并定义它的作用范围, 就可以在很多个页面中使用它操作数据库。使用 Bean 还可以提高系统的安全性, 因为 Bean 是编译为字节码存放, 所以可以将用户帐号及口令等需要保密的信息封装在预编译好的 JavaBean 类中, 从而提高了系统的安全性^[3]。

在数据库接口方面采用 JDBC (Java database connectivity) for Oracle, JDBC 定义了 Java 与数据库之间的接口类库, 接口类库是支持基本 SQL 功能的一个 Java 的通用底层应用编程接口。

JDBC 的体系结构包括 4 个组件。

- 1) JDBC 应用程序: 负责用户与用户接口之间的交互操作, 调用 JDBC 对象方法, 给出 SQL 语句并提取结果。
- 2) JDBC 驱动程序管理器: 为应用程序加载和调用驱动程序。

3) JDBC 驱动程序: 执行 JDBC 对象方法的调用, 呈送 SQL 请求给指定的数据源, 将结果返回应用程序。

4) 数据源: 主要指数据集和相关数据库系统。

在开发阶段, 注意到应用程序服务器为每一个客户机进程提供了一个有状态有连接的服务, 这样虽然提高了客户机与服务器通信的速度, 但是, 速度的提高是以牺牲服务器系统资源为代价的。应用程序服务器要为每一个未结束的客户机进程保持连接, 因此服务器进程中与每一个客户机进程进行通信的子线程都必须运行着, 这些子线程将长期占用系统资源。随着客户机的增多, 若在同一时间内有大量的客户机进程要求服务, 就会导致服务器的系统资源不断被消耗, 服务器的性能将会不断降低。为了改善系统性能, 需要适时可关闭部分连接, 所以本系统中建立了一个对于 Oracle 数据库的连接池 (Connection pool), 当一个用户发来请求时, 该用户立即就可以向连接池要一个连接 (Connection) 来使用, 而不是花时间去创建或初始化一个连接。用户请求结束时将该连接交回给连接池以供其他请求使用, 这样可以节省许多时间。

4 结束语

所设计的仓储管理信息系统已在石家庄、无锡、沈阳等地投入使用, 结果表明企业的工作效率得到提高, 实现了资源共享, 减少了对某一环节或某一岗位的依赖, 提高了信息的透明度, 加强了各个环节间的衔接, 并且可以由该软件平台实时地掌握各类物资的情况与客户资料, 有效地减少了业务差错, 该软件平台的建立使得企业形成了一个全国性的物流服务网络系统, 实现了从局部分散物流服务到全国性综合物流服务的跨越。

参 考 文 献

- 1 谈谈 M IS 系统平台 [http: www. 21cnm. com/new sw e/3f_m ispt.htm](http://www.21cnm.com/newsw e/3f_m ispt.htm). 2001
- 2 Perrone J 著 J2EE 构建企业系统 张志伟, 谭郁松, 张明杰, 译 北京: 清华大学出版社, 2001. 825~ 826
- 3 黄 理, 李积善, 曹林有 用 JSP 轻松开发 W eb 网站 北京: 北京希望电子出版社, 2001. 240~ 250