一个基于 C/S 与 B/S 混合架构的应用实例解析

易任重　刘晓海　廖晓昕
(华中理工大学,武汉 430074)
E-mail:yirenzhong@163.net

摘　要　文章分析了传统 C/S 模型和三层 B/S 模型各自的特点。在一实际应用系统中采用了两种模型相结合的策略,并作了进一步的改进,通过引入文件服务器等独特方式,充分发挥其各自优势,取得了良好的效果。

关键词　C/S 模型　B/S 模型　运行环境　WRB　PL/SQL　Cartridge　PL/SQL Agent

文章编号 1002-8331-(2001)16-0159-03　文献标识码 A　中图分类号 TP311

An Application's Analysis Based on the Combination Model of Traditional Client/Sever and Three-Tier Browser/Sever

Yi Renzhong　Liu Xiaohai　Liao Xiaoxin
(Huazhong university of science and technology, Wuhan 430074)

Abstract: In this paper, we will compare traditional C/S model with three-Tier B/S model and analyze their respective characteristic in detail, Present a application system which adopts a tactics that combines two models, and introduce some special methods such as files server to exert their advantages respectively and gain a better result.

Keywords: C/S Model, B/S Model, Run Environment, WRB, PL/SQL Cartridge, PL/SQL Agent

1　技术背景

80 年代末兴起的 C/S 计算方式以其所具有的强壮的数据操纵和事务处理能力,以及严密的数据安全性和完整性约束而成为通用的企业信息系统体系结构。但在多年的实际应用中,C/S 方式也暴露出其不少的弊端,主要有以下几个方面:

占用客户端资源多。客户机需安装客户方软件,客户端工作包括用户交互、数据显示、处理应用逻辑,这对客户机性能提出了较高要求。

不易安装、维护。应用程序必须在每台客户机上安装,系统配置工作也不得不在众多客户机上逐台进行。特别是系统投入运行后,对系统的维护和升级将带来巨大的工作量。

代码不能重用。系统通常基于专用的平台,使用特定开发工具,源代码难以移植到其它系统。

针对传统 C/S 方式的缺陷,人们在客户端与数据库服务器之间加入中间层,即应用服务服务器。这样由逻辑层 客户机)、业务逻辑层 应用服务服务器)、数据逻辑层 数据库服务器)构成了典型的多层结构——三层模型。其中客户机负责用户界面及与应用服务器的交互;应用服务器处理应用逻辑,接受客户请求并化为向数据库的请求,传递操作结果给客户端,充当联系纽带 数据库服务器根据应用服务器发送的请求进行数据库操作,并向应用服务器返回操作的结果。三层 C/S 模型提供了一种能使应用程序的不同组件放在不同的执行系统上的机制,克服了传统 C/S 模型的不足,具有更大的灵活性。

Browser/Web Server /Database Server 体系结构是三层模型的一个常见的实例。Web 技术具有很好的信息发布方式,而数据库技术拥有对大量数据的组织管理能力。两种技术具有一定的互补性,两者的融合发展使 Web 由分布式超文本环境演变为搭载各类应用的综合平台。但 B/S 体系结构同样有它的不足之处:

HTTP 协议使得客户端界面简单、规范,却又导致界面不能控制自如。单纯的 HTML 语言不能模拟复杂、特殊功能。下载中间件的方法也有诸多不便。

Web 从最初起源就是开放体系,各页面之间的独立性使得安全机制难于保证。当既要求系统的安全保密性,又要求灵活的控制机制,便常常陷入两难。

在简单的 Web 应用中,各页面之间除了链接无其他关系。在传统 C/S 方式中可使用全局变量、参数传递等来实现复杂信息系统中多个界面之间的相互联系。但在 Web 页中使用 Cookie、Hidden 时很不方便且容易产生安全隐患。

HTTP 是非常简单的无态协议,Web 服务器原本只是提供文档查询服务,并不适于事务处理,实现相互依赖的操作麻烦,事务处理的连续性、数据的完整性难以保证。

由上几点可以看出,要得到一个功能更加强大而且处理方式更加灵活,编程更加容易的应用系统,单纯采用哪一种技术是不够的或是不现实的 传统的两级 Client/Server 计算与 Internet 计算共存仍将继续存在)。因此有必要在概念上进行扩展,采用 C/S 模型与 B/S 模型相接合的方式来构建系统,以期扬长避短,在实际应用中得到良好的效果。

2　C/S 模型与 B/S 模型相结合的应用实例

2.1　概述

在一个实际的信息系统中有两类用户,一是企业内部用户需要复杂的业务操作,另一类是企业外部人员需要一些简单信息查询和数据录入,为此采用 C/S 模型与 B/S 模型相结合的策略。即降低开发难度,同时也满足用户需求。其中 C/S 结构在实现方法上与经典模型有较大差异,主要是应用程序并不直接安装在客户机上,而是存在于一公用的文件服务器,客户机上只存在运行环境。具体实现方法如下。

数据库服务器：安装 Windows NT、Oracle 8

备份服务器：与数据库服务器相同

Web 服务器：安装 Windows NT、Oracle Web Application Serve

文件服务器：安装 Windows NT

应用程序文件（包括 Fmx、Mmx、Rep、Ogd、Brw 等）

资源文件（图标、图片、pll 等）

客户机（企业内）：主要安装

Windows 98、Internet Explorer 浏览器

Oracle 运行环境（Form Runtime、Reports Runtime、Graphics Runtime）

Oracle 数据库连接工具 SQL Net Easy Configuration

指向文件服务器主窗体（Form/ Fmx 文件）的快捷方式

另外为了实现导出任意数据块中所见数据至 EXCEL，特意安装 Query Builder。

企业外部客户机可任意安装操作系统与 Web 浏览器。

系统结构如图 1 所示。

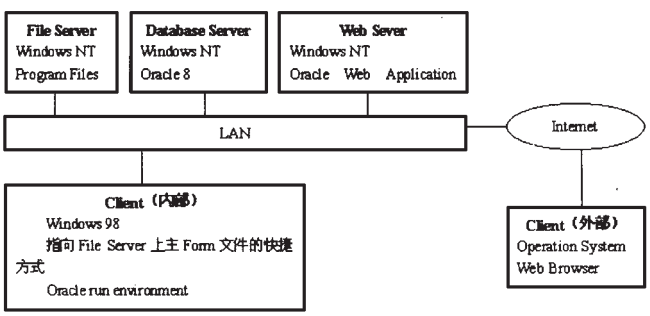

图 1

2.2 C/S 部分的结构

系统的主要前端开发工具为 Developer2000，与其它常见开发工具相比它所生成的应用程序有两个显著不同点。一是编译后的文件不是可单独运行的 exe 文件，而是不能脱离 Oracle 独特运行环境的 Fmx、Mmx、Rep、Ogd 文件；二是其分散性。每一窗体（Form）、菜单（Menu）、报表（Report）、图表（Graphics）都分别编译成单独的 Fmx、Mmx、Rep、Ogd 文件，它们相互之间可使用 call_form、run_product 等内置子程序相互调用，图标、图片、pll 也不编译入窗体、菜单、报表等内部。

这两个特点一方面给系统带来了运行速度慢、不能脱离运行环境的弱点。另一方面也便利了程序文件与运行环境的分离，即程序文件及资源文件由一台公共的文件服务器管理，而众多客户机只安装运行环境和添加一指向文件服务器上程序文件的快捷方式。文件服务器的引入带来不少优点，首先系统的安装、维护、升级的工作量大大减少，因为只需要保证一台服务器上文件的正确性。其次客户机上硬盘空间占用也减少了，程序虽依然在客户端运行，但客户机终究变瘦了。客户机在读取服务器上文件过程中会增加网络通讯量，但整个系统由许多分散的文件组成，绝大部分文件在 10K 至 300K 之间，每次操作通常只调用一个文件，对与处于文件服务器同一局域网的客户机而言，这部分网络延迟可以忽略。文件服务器并非为本系统专用而新添设，利用原有设备不必增加硬件投资。

Developer2000 另一特点是与 Oracle 数据库结合紧密，这使得应用系统在安全性上具有突出优点。应用系统从 NT 操作系统、窗体、菜单、到后台数据库都存在一连串安全机制，用户开机时需登录网络，以便访问文件服务器上的程序。数据库中各角色具有访问相应数据库资源的权力，各用户分配了合适角色，但平时角色未被激活。运行主窗体后用户填写用户名口令连接数据库，并在程序运行时才激活所分配角色（用户不能绕过应用逻辑，利用自己的用户名口令直接操作数据库）。系统中菜单也仅为指定角色才能使用。

使用其它前端开发工具时，常使用的一种作法是用户输入伪用户名、口令进入系统，程序中再使用真实用户连接数据库。这样真实用户名、口令在程序中设定，不便更改，系统对开发人员不设防。伪用户名、口令虽被加密，但仍有被破解和源程序泄密的可能。

2.3 B/S 部分的结构

Web 服务器为企业外部客户提供了使用该系统的途径。为便利安全管理，系统使用 Oracle Web Application Server 作为 Web 服务器。它主要由以下三个部分组成：

Web Listener（Web 收听器）：它是一个 HTTP 服务器，用来接受用户请求并提供 Web 服务，即提供静态页面和文档服务，执行 CGI 程序，并可激活分布在多台机器的服务器应用程序。

Web Request Broker（Web 请求代理）：WRB 为 Web 服务器的核心，它是一个符合 CORBA 标准的对象请求代理（ORA），也是可由浏览器调用并执行的接口。正是通过 WRB 服务，服务器上应用——Cartridge 被调用和执行，结果也是通过 WRB 返回浏览器。

Cartridge（部件）：Oracle Web Application Serve 含有 PL/SQL、Java、LiveHTML、ODBC、Perl 五个 Cartridge（该应用系统使用 PL/SQL Cartridge），它们是 Web 服务器上的效率高于 CGI 的可执行应用程序，作为中间件访问其它语言（如 PL/SQL）编写的应用，或直接提供用户业务逻辑。

为访问 Oracle 数据库，Oracle Web Application Server 需要进行配置。DAD（Database Access Descriptor 数据库访问描述符）用于定义数据库、用户名、口令、环境变量等。PL/SQL Agent 定义了 PL/SQL Cartridge 在调用时如何连接 Oracle 数据库的参数，实际是一个对 PL/SQL Cartridge 的具体调用。PL/SQL Cartridge 通过 PL/SQL Agent 取得 DAD 中的数据库信息来定位、连接、访问数据库。

当浏览器发送一个需通过 WRB 访问 PL/SQL Cartridge 提供的服务的 HTTP 请求后，Web Listener 接受并分析 URL 后将请求传给 WRB，WRB 根据 URL 中虚拟路径判断要调用 PL/SQL Cartridge，PL/SQL Cartridge 根据 URL 中 PL/SQL Agent 的定义连接 Oracle 数据库，激活数据库中的存储过程（Store Process），并同时将参数传递给存储过程，存储过程在数据库端运行后的结果为含有基表（Table）数据的 HTML 语法流，构成 Web 页面的结果先存放于响应缓冲区，再经沿 PL/SQL Cartridge、WRB 、Web Listener 返回浏览器。如图 2 所示。

这种结构使得系统具有几个优点：

高效：运行在数据库服务器上的存储过程操作数据的速度相当快。

安全：Oracle Web Application Server 提供身份验证与加密机制。并且不管用户请求的 URL 虚拟路径中存储过程名为何或者过程是否存在，都将受到 Oracle 数据库的安全机制检查。不可能出现用户绕过安全检查页面，直接链接后面内容的漏洞。（URL 为 http://Web 服务器名 / PL/SQL 代理名 / plsql / 存储过程名）

简便：用 PL/SQL（标准 SQL 的扩展）编写的存储过程访问 Oracle 数据库自然是再方便不过了。

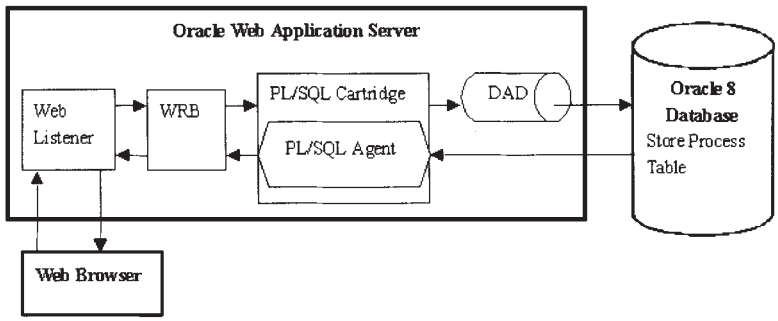

图 2

用 PL/SQL Cartridge 开发 Web 应用需要 htp、htf 等开发包，调用其中的过程、函数，用于输出 HTML 语法，执行数据库操作，格式化操作结果。这样带来两个弊端，一是开发时完全不可视化，代码编写繁琐，调试麻烦。二是存储过程由开发包特有的过程、函数编写，需开发人员重新学习，且代码不可移植。

在系统实际开发时并未真正完全依赖开发包，具体方法如下：先搁置来源于数据库的动态数据，用 Front Page 制作页面静态主体部分，获取其 HTML 代码，该长篇幅的 HTML 代码作为过程 htp.print 的参数。当执行 htp.print（HTML 代码′）语句时即产生合适的 HTML 语法流，而动态部分用其它 PL/SQL 语句镶入。总个存储过程大体分为三部分，即 htp.print（页面前部 HTML 代码′），产生动态数据的 PL/SQL 语句，htp.print（页面后部 HTML 代码′）。如此，开发难度会大大降低。

3 结束语

该系统最大的特点是 C/S 与 B/S 体系结构的结合，融合两种技术模型的优点，同时使用一定技巧避免各自弊端，如文件服务器的引入，过程 htp.print 的使用等。而且数据库服务器、前端开发工具、Web 服务器同为 Oracle 产品，相互间结合紧密，配合良好，Oracle 优秀的安全性得到了充分的发挥。

（收稿日期：2000 年 5 月）

参考文献

1.晓通网络数据库研究所.Oracle8 系统开发与管理[M].内蒙古人民出版社
2.David lockman.健莲工作室译.Teach Yourself Oracle 8 Database Development in 21 Days[M]
3.於长华等.基于三层 C/S 模型的大型关系数据库应用系统优化设计技术[J].计算机工程与应用，1999；35（1）：90-91

（止接 78 页）

中实现了多媒体视频和音频的通信和控制。作者在 windows98、windows NT、Red Hat Linux 以及 Solaris 系统中进行了测试和分析，得到了良好的效果。如图 3、4 所示。

5 结束语

基于异构环境下的多媒体通信使得处于不同操作平台（UNIX 工作站和基于 Windows 平台的 PC）的用户可以在 Internet 上进行拥有视频和音频功能的会议通信，支持了在现有网络环境下人们之间更广泛、更有效的合作交流。使用 Java 和 JMF 可以使用户不必关心所采用的媒体格式，甚至所使用外部设备的兼容性，而多址广播和分布式控制策略更有效地支持了分布式协作。因此，对于异构环境下的分布式多媒体通信的研究和试验，不仅具有一定的研究价值，也具有广阔的应用前景。

（收稿日期：2000 年 5 月）

参考文献

1.SUN MicroSystems.Java Media Framework Programmer′s Guide[M]. 1999.11
2.SUN MicroSystems.Java Media Framework API Data Sheet[M]. 1999.11
3.SUN MicroSystems.Java Media Framework API White Paper[M].1999.11
4.H Abdel-Wahab,O Kim,P Kabore et al.Java-based Multimedia Collaboration and Application Sharing Environment[M]
5.H Abdel-Wahab,b Kvande,S Nanjangud.Using Java for Multimedia Collaborative Applications[M]
6.王峰.Java 多媒体程序设计[M].清华大学出版社

（止接 158 页）

像交互以及互操作基于设备所支持的 DICOM 接口进行，借助 Microsoft SQL Server7.0 数据库和 FTP 机制组织管理图像文件。

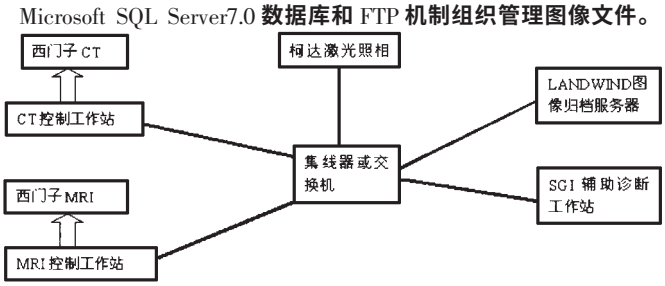

图 1

在设计和实现这个 miniPACS 过程中，作者掌握了设计和实现 PACS 的一些经验，总结了在设计和实现时，着重考虑了上述的几个关键问题，同时也加深了对 DICOM 标准和 PACS 的认识，为下一步的工作打下了坚实的基础。

（收稿日期：2000 年 6 月）

参考文献

1.赵喜平，郑崇勋，毛松寿.PACS 的发展趋势[J].中华放射学杂志，1998；32（）
2.卫生部影像装备专家组.83 届北美放射学会医学影像学设备发展动态[J].中华放射学杂志，1998；32（）
3.Digital Imaging and Communications in Medicine - 99′s final draft [EB].PS3.1-PS3.8 http：//global.ihs.com/
4.PC-Based DICOM Workstation Software--PIVIEW 3D[EB].Canada Mediface CO.LTD,http：//www.mediface.com,1999