

ORACLE 在福田保税区管理局 MIS 中的应用

卢健鹏 (深圳市福田保税区管理局)

摘要: 性能优异、使用便捷的数据库是一个管理信息系统 (MIS) 得以顺利实施、扩展的必要条件。本文概括地介绍在 UNIX 的 TCP/IP 网络环境下, 如何利用 ORACLE RDBMS 设计“福田保税区管理局信息管理系统”这一大型分布式数据库应用系统的总体设计及方法, 总结了一些应用开发中的经验和体会。

关键词: 福田保税区 MIS ORACLE Developer/2000

车辆监管、驻区企业管理、计划统计、劳动人事档案管理及各种综合信息实现了计算机管理。系统采用了先进的 ORACLE 数据库为主数据库, 利用 ORACLE 提供的 PRO*C、Developer/2000 以及 Sun 的 Java 等工具为主要开发工具。在开发过程中, 我们遇到了不少问题和困难, 也积累了不少经验, 这对我们今后应用 ORACLE 开发新的应用系统提供了一个很好的实践。

2 系统的总体结构

保税区由管理局、局属企业以及驻区企业组成。管理局有六处一室一中心, 集中在八层的办公大楼内, 约有 200 多个节点。管理局的业务流程如图 1 所示。

1 前言

福田保税区作为深港衔接和深圳特区进一步发展的“先行点”和“试验场”, 经过几年的基本建设和功能开发, 特别是从 1996 年进入实际运作以来, 福田保税区得到了高速发展, 至今已有 24 个国家和地区的 600 多家企业在保税区投资设厂。福田保税区管理局 (以下简称管理局) 作为深圳市人民政府的派出机构, 全权负责保税区内的各项行政事务。随着保税区的高速发展, 管理局业务迅速增多、覆盖面增宽, 总的特点是信息量大、信息交流快, 管理这样范围内的数据就提出了对计算机资源联网、实现分布式管理、共享资源等要求, 面对这些新的要求和变化, 我们实施了“福田保税区管理局信息系统网络工程”, 采用了最流行的 TCP/IP 网络协议及先进的 ATM 技术, 建成了管理局 Intranet, 实现了管理局各职能部门之间的联网, 并开发出管理局管理信息系统 (FFTZ-MIS), 对管理局的

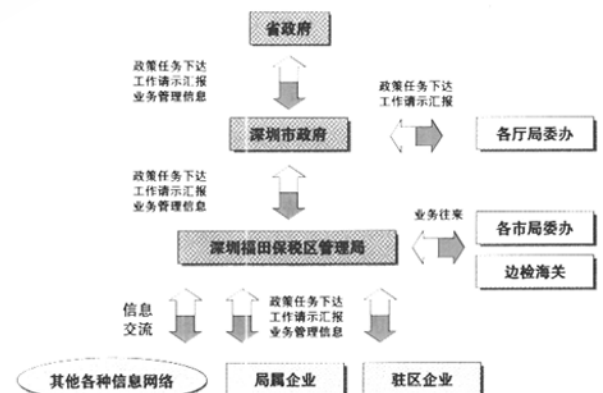


图 1 业务流程

3 系统的功能

根据管理局具体的实际业务情况,我们把图1的业务流程转换成图2的功能图。



图2 系统功能图

3.1 办公自动化(OA)系统

OA系统贯穿管理局所有处室部门,完成管理局公文处理工作,是实现办公电子化、无纸化的手段。OA系统以当今最先进的办公群体LOTUS DOMAIN /NOTES作为平台,由“通用公文处理子系统”、“政务管理、追踪子系统”等子系统组成,并与专业业务系统及综合业务系统通过LOTUS和ORACLE提供的API及OCI有机地结合在一起,实现资源共享,避免了数据的重复建设和冗余。

3.2 专业业务系统

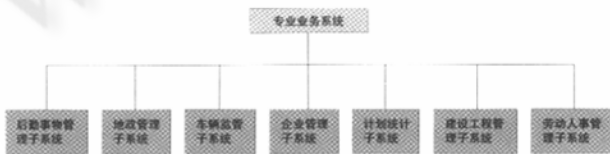


图3 专业业务系统结构

如图3所示,专业业务系统包括:地政管理子系统、劳动人事管理子系统、企业管理子系统、车辆监管子系统、建设工程管理子系统、后勤事务管理子系统、计划统计子系统。

3.3 综合业务系统

如图4所示,综合业务系统包括:保税区基本情况查询子系统、视频信息点播子系统。



图4 综合业务系统结构图

4 系统的实施

管理局信息系统(FFTZ—MIS)是根据管理局自身特点,采用ATM技术及TCP/IP网络协议、以ORACLE为主数据库的Intranet。系统结构图如下(图5):

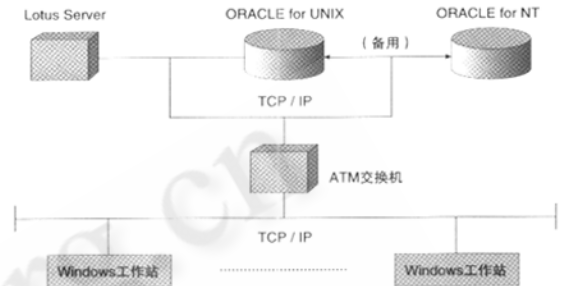


图5 系统结构图

在系统的实施过程中,我们根据自身要求,针对ORACLE RDBMS的特点,在技术上采用了一些有别于常规的方法:

(1) 由于系统存储的数据都涉及到各部门的核心业务,因此,系统对数据的完整性及实时性要求较高。所以,我们需要利用冗余的资源减少系统停工时间。根据自身特点,我们采用了备用数据库(Standby Database)的方法。同时,为了节省资源,备用数据库所在的服务器并不采用跟主数据库(Primary Database)服务器一样的配置和操作系统(OS),而采用了操作系统异构的形式:主数据库是For UNIX的,而备份数据库是For NT的(图6)。

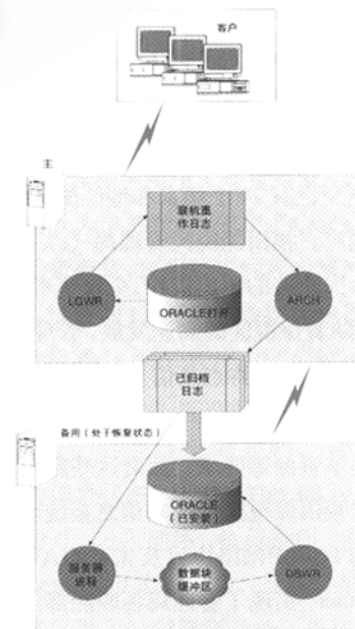


图6 数据库备用策略结构图

(2) 存储策略: 虽然 RAID 0 代表了 RAID 级别中最高的存贮性能, 但由于它不能提供数据冗余, 所以目前的大多数系统都采用了 RAID 5 硬盘系统来存储数据。然而, 虽然 RAID 5 能够为系统提供安全保障, 但如果单纯采用 RAID 5, 会给数据库操作带来了巨大的额外开销, 特别是在 OLTP 模式下, 严重地影响了数据库的性能。本系统是一个典型的 OLTP 应用系统, 有大量的经常需要写操作的应用程序。因此, 如果单纯采用 RAID 5, 虽然能够提供数据安全保障, 但每个写操作都必须带有读取原来奇偶校验数据、计算新奇偶校验数据、写入新奇偶校验数据的额外开销, 然后才真正写入数据, 在进行最后两次写入操作时, 被写入的两个硬盘驱动器都必须同时被上锁。这些额外开销就是 RAID 5 中的写损失 (write penalty)。Write Penalty 可能会使得写操作变得很慢。同时如果 RAID 5 配置中的某个硬盘失败, 那么这种失败所导致的 I/O 性能损失 (进行硬盘重建) 就非常大, 使 ORACLE 的性能急剧下降并有可能使系统停工的时间延长。基于以上考虑, 我们采用了 RAID 0+5 方案, 使到 ORACLE 数据库的性能及安全性得到有力的保障。

(3) 数据库的异构: 由于 OA 系统采用 Lotus 软件作为平台, 而专业业务系统与综合业务系统采用 ORACLE 数据库, 因此必须解决数据共享的问题。我们采用了 ORACLE 提供的 ODBC 接口、OCI 接口及 Lotus 提供的 API, 实现了数据的透明访问及调用。最终用户在 OA 系统中能方便地访问专业业务系统与综合业务系统的数据, 而在这二个系统中又能快捷地访问到 OA 的内容。在实现数据共享的同时, 系统并不破坏异构数据库中各个组成部分自身的自治性, 每个数据库系统仍保持自己的应用特性、完整性控制和安全性控制。

5 系统的应用开发

在系统开发中, 我们采用了 ORACLE 的 Developer/2000 作为客户端软件的主开发工具, 它是一个基于标准的新一代 Client/Server 开发工具。ORACLE Developer/2000 通过快速应用开发 (RAD) 技术、面向对象技术, 基于计算机的内部学习, 以及统一的客户、应用服务器和数据服务器体系结构, 为客户/服务器和 Web 开发建立了新的易于使用的生产力标准。而它的报表构建器 (Report Builder) 则可帮助开发人员轻易建立或修改 (有表状的、分组的、邮件标签的、格式的以及矩阵/多表格等各种类型报表) 的全过程。Web 向导还可以构造 HTML 和 Adobe

Acrobat 报表结构。同时, 它的图形构造器 (Graphics Builder) 能够跟 Form 及 Report 紧密地结合在一起, 方便地产生出用户所需的各种漂亮的图表。在此, 我谈谈在 Forms 以及 Reports 开发过程中的一些小小体会。

5.1 在 Forms 中模糊查询的实现

在系统的数据查询、统计分析等模块中, 通常用户是按若干条件来查询的, 这些条件有时是相当模糊的, 假如使用别的开发工具, 可能需要编写大量语句, 而在 Developer/2000 中的 Form Builder 中则可轻易实现。在利用 Form Builder 设计时, 首先设计一个条件设置块 (此块对应数据库中的一个物理对象或逻辑对象——Table 或 View), 然后在块中设置项 (Item) 对应数据库对象中的列 (Column), 利用该块提供一个简洁明快的交互界面来接收条件。对任一查询时, 可以在所有项中任意组合地输入不同内容, 而内容可以是模糊的 (只需知道内容的前几位或确定的代码段就行了), 条件关系可设可不设 (“或”和“与”的关系)。并在用户查询条件设置过程中对一些项利用 When-Item-Validate 触发器进行立即有效性检验, 最后使用 Form Builder 中的 “Enter-query” 及 “Execute-query” 内置包 (Build-in Package), 则可顺利地查询到所需的结果。

5.2 报表的实现

Developer/2000 的 Report Builder 提供了在客户机上直接执行远程报表服务器上的报表并具有从不同的数据库服务器上存取数据的能力。在报表设计过程中, 我们通过指定一个特别的域 (Formula Column) 作为一系列数据的总计或通过一个公式 (Placeholder Column) 计算一个或多个列的值而无需写任何代码。另外, 通过设定一个非块的 UPI 连接到数据库来执行查询, 在 Forms 里通过一个 “中断” 对话框可以中断已在运行的又长又复杂的查询, 方便了程序设计及用户使用。其次, 可以在 “重复框架” 中设定 “每页最小纪录数” 和 “最小孤立纪录数” 来灵活控制查询及显示结果。

5.3 安全性的实现

在分布式环境下, 随着信息交换和资源共享的进一步扩大, 安全保密问题越发突出, 为确保数据安全, 我们分别在数据库端和客户软件端作了相应的安全保密措施。

在 UNIX 和 ORACLE 级上, 所有高特权的用户仅由系统管理员和 DBA 掌握口令, 口令做不定期更改; 对不同用户进行不同级别的授权, 用户需通过口令才能连接到数据库, 并把开发用户和使用用户分开, 只给使用用户授予

最保守的权限,使最终使用用户只能使用DML语句而不能使用DDL及DCL语句,保证了数据库对象的安全;利用ORACLE提供的审计功能,对系统数据字典和存放用户重要数据的表进行审计,审计任何非法操作来改善数据库的安全保密性。

另外,我们除了通过在数据库级上对存放用户重要数据的表进行审计外,在程序开发过程中,我们还利用ORACLE提供的触发器谓词(INSERTING、UPDATING和DELETING)和“伪记录”(Pseudo-records)——:old和:new,在数据库行级上编写了“审计”触发器(audit trigger),对用户某些重要的表所做的所有变动进行记录。除了记录所做的变动外,它还记录谁进行变动,变动的操作类型(INSERT、UPDATE和DELETE)。该触发器也可以用来检测用户是否有限进行修改,如果没有的话,将会引发一个错误(使用RAISE_APPLICATION_ERROR)。

6 结束语

目前,系统已正式运行,并已取得较好的效果,有效地提高了机关工作效率,大大减轻了工作人员的工作量。我们也将随时根据业务的变化,不断地完善系统。随着局属企业管理信息系统的建立,我们将逐步利用数据库互联技术把这两个系统构建成一个复杂、大型的分布式数据库系统,为福田保税区经济高速增长铺就“e”路。■

参考文献

- 1 《实践与探索》,福田保税区管理局。
- 2 《ORACLE产品白皮书》,ORACLE公司。
- 3 《保税区管理信息系统的总体设计》,何心远、邱名卿。
- 4 《Oracle8: The Complete Reference》。
- 5 《Object-Oriented Systems Analysis And Design》 Ronald J. Norman。