

高负载 B/S 系统的设计实例分析^①

Analysis and Design on Heavy Load B/S System

曹 平 葛一鸣 (浙江工业大学 之江学院信息分院 软件学院 浙江 杭州 310023)

摘 要: 围绕浙江省中小学生困难补助发放系统的开发,分析了该系统的应用特点,提出在高负载环境下的设计方法和策略。对于类似的政务信息化工程,及高负载 B/S 系统的设计有一定参考价值。对高负载处理及系统管理起关键作用的模块,如数据上传模块、分布式系统管理模块等做了深入的分析和讨论。实际运行效果很好,验证了系统设计方案的可行性。

关键词: B/S 系统设计 高负载 电子政务 多 Agent 系统 信息系统

为了配合基础教育资助体系的建立,浙江省财政厅从 2005 年开始启用中小学生困难补助发放系统,将贫困学生申报,补助资金发放等工作信息化。

系统用户为省内基础教育的所有类型学校(包括小学、初中、高中、九年一贯制、完全中学、职高等)、县区级财政局、县区级教育局、市级教育局和财政局、教育厅和财政厅。系统的工作目标是收集全省中小学生数据,采集贫困学生数量,统计学生、学校、县市的补助资金,生成相应的报表。

但是由于该系统设计时没有充分考虑实际负荷情况,难以支撑全省用户同时操作系统时所产生的巨大网络流量和计算压力。两年来,系统运行异常缓慢,频繁宕机,影响了各级部门工作效率。为解决这个问题,在仔细分析系统应用特点的基础上,我们重新对系统架构进行了设计和开发。相比原系统在减少硬件投入的情况下,系统性能有了大幅度的提高,从 2007 年的运行情况来看,完全满足了实际的需要。

1 系统的特点分析

1.1 工作流程

贫困学生补助工作是由学校,教育局,财政厅三方共同协作完成的。其简单流程图如图 1 所示:

首先由学校用户输入本校所有的学生信息,通过校验确认无误后,上传服务器。然后,输入收费项目,

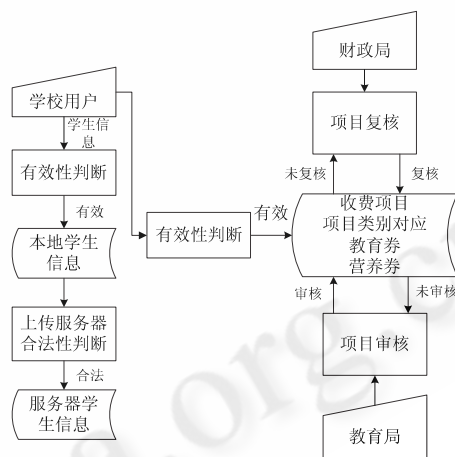


图 1 简单工作流程

项目类别对应,教育券,营养券等信息,等待教育局和财政局的审核和复核。

网上申报工作结束后,教育局和财政局,可以取得相关的补助报表,打印教育券和营养券(图中未标出)以配合补助资金的发送和获取。

1.2 数据量大

全省中小学学生人数约为 500 万左右,按照贫困学生 10% 计算,营养券约 50 万,每位贫困生能领取 1-4 种教育券,因此教育券数量在 150 万左右。因此,核心数据量约为 700 万。

1.3 用户数量多

本系统的实际用户包括浙江省内(除宁波地区

^① 基金项目:浙江工业大学浙江省财政厅专项(Z13413502)

外)所有的中小学,职高,县级,市级教育局和财政局,共约 8000 多名用户。因此,本系统在工作高峰时,同时在线人数预计在 1000 人左右。

1.4 单用户操作工作量大

不仅在线人数多,而且,单个用户的操作往往工作量很大。比如,学校用户的学生批量上传,常常涉及 500 - 2000 条学生记录的导入。财政局用户的报表统计,又经常包含上百所学校,数十万学生。

1.5 数据操作的地域性

虽然全省数据量很大,用户操作产生的负载也比较重,但是用户所操作的数据具有明显的局部性。即某学校用户,不会去改写其它学校的数据,极少去查看其它学校的数据。某教育局或财政局用户,极少去查看不在它管辖范围内的学校数据,如杭州市教育局用户不会去查看,统计湖州市各个中小学的情况。

1.6 瞬时负载重

所有用户的操作,有很强的时间性,即统一集中在各个部门的工作时间。周一到周五的负载比周六,周日重。就每个工作日而言,早上八点到晚上六点的负载较重,其中,又以工作高峰,上午 10 点左右,下午 2 点左右,达到系统负载的峰值。晚上 11 点到早晨 7 点,负载通常比较轻。

1.7 分布式系统管理困难

由于系统使用六台计算机作为 web 服务器(详见 2.1 节),根据需要,每台计算机可能又对外提供不同的服务,比如 2.3 节和 2.4 节所述的学生信息录入服务和报表生成服务可能运行在不同的服务器上。同时,作为负载均衡的参考数据,还需要不断取得服务器的资源情况。因此如何有效地管理分布式集群也是本系统要解决的问题之一。

2 系统设计

2.1 系统物理结构

系统物理系统物理结构如图 2 所示:

系统由六台 Dell1950 组成 web 服务器群,采用 IBM 小型机作为 Oracle 服务器。短信网关,作为辅助服务工具,作为政务消息的发送和用户操作的反馈。

其中,六台 web 服务器使用负载均衡策略,合理的将用户请求平分在各台服务器上。

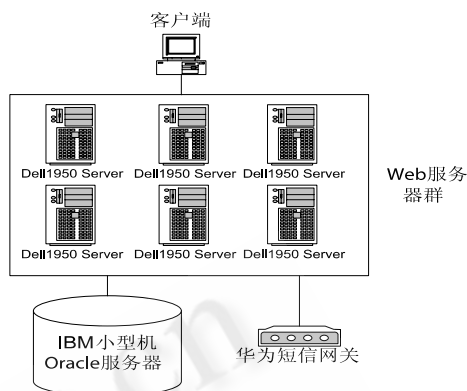


图 2 系统物理结构

2.2 学生表的设计

由于学生表的记录数达 500 万以上,各类用户又经常对学生表进行大量的操作,所以对学生表的处理成为系统中负载最重的部分。通常涉及上千条数据的插入,删除,以及数十万条数据的统计和查询。因此将全省学生存在单独的物理表中,会严重影响系统的性能。根据用户对系统操作的地域特性,根据不同的地区,以市一级为单位,将学生表划分成 12 张小表,每张表包含 50 万左右的数据。从而减缓了数据库对大表操作带来的性能损耗。

2.3 学生数据上传模块的设计

2.3.1 基本设计

学生数据上传是整个系统中负载最重的一项操作。遇到的主要问题有:首先,学校每学期要上传新生数据并删除毕业生数据,通常,估计在这一阶段,数据流动量在 100 至 150 万之间。其次,这一工作具有很强的时间局部性,都在开学前后一个月内完成,会给系统带来较大的瞬时负载。第三,学校上传的学生数据,又必须满足完整性,一致性的约束,因此,必须进行大量的校验工作。学生数据校验会涉及多表连查等操作。如,学生的年级信息要和年级表的连查,专业信息要与专业表的连查,大量的校验会使得系统运行异常缓慢。最后,在学生数据上传至服务器时,由于用户的操作过于集中,可能会导致服务器崩溃。

基于以上原因,考虑将校验工作分布到各个客户端进行,以减轻服务器的压力。基本的设计如下:

(1) 学校用户下载学生信息校验程序(程序中包含各个学校校验所需的数据)。

(2) 学校填写包含学生数据的 Excel 表格。

(3) 校验 Excel 表格, 校验成功后生成二进制学生数据文件 Zxx。

(4) 将 Zxx 文件以附件形式发送到邮件服务器。

(5) 服务端程序接受邮件, 只需做简单验证, 即可导入数据库。

(6) 导入成功后, 以邮件和短信形式通知用户。

基于以上操作, 将数据校验的负载从服务端转移到各个客户端进行, 同时将数据上传的压力从 web 服务转移到邮件服务器, 通过使用网易等公司提供的邮件服务, 可以很好的缓解大量数据上传时的压力, 使得系统效率和稳定性有了很大的提高。

2.3.2 实现原理

(1) 客户端校验程序的实现

如图 3 所示, 是客户端应用程序在系统中的地位。它在进行 Excel 表格检验后, 将代表用户视图的 Excel 表格, 转为 Oracle 可识别的逻辑视图表格。在转换过程中, 首先使用 ADO^[1] 技术读入表格进行校验, 然后, 使用 Excel VBA^[2] 产生对应于逻辑视图的 Excel 表格。最后, 将产生的表格加密, 生成二进制文件 Zxx。防止用户无意或恶意篡改校验过的文件。

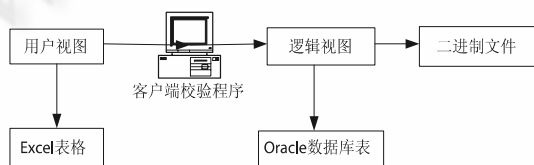


图 3 客户端程序功能

(2) 服务端程序的实现

服务器处理程序的工作环境如图 4 所示。负责从邮件服务器接受学校提交的学生数据, 并插入 Oracle 数据库。

开发可靠的邮件接受程序是一项比较繁重的工作, 会给项目开发带来很大的工作量。而 Office 套件中, 已经包含了优秀的 Outlook 邮件客户端程序。因此, 使用 Outlook 作为邮件接受程序对提高系统开发效率, 增强系统稳定性有一定帮助。



图 4 服务端程序功能

Outlook 提供完善的对象模型, 使用 Outlook 的对象模型可以将 Outlook 的大部分功能集成到特定的应用程序中。Outlook 提供的顶层对象如下所示^[3]:

```

Microsoft.Office.Interop.Outlook.Application
Microsoft.Office.Interop.Outlook.Explorer
Microsoft.Office.Interop.Outlook.Inspector
Microsoft.Office.Interop.Outlook.MAPIFolder
Microsoft.Office.Interop.Outlook.MailItem
Microsoft.Office.Interop.Outlook.AppointmentItem
Microsoft.Office.Interop.Outlook.TaskItem
Microsoft.Office.Interop.Outlook.ContactItem
  
```

然而, 出于对隐私的保护, 通过 Outlook 对象模型取得发件人地址或者进行邮件发送时, 会要求用户进行安全确认。对于后台工作的无人值守的服务程序而言, 这是不合适的。为了绕过 Outlook 的安全认证, 直接通过 Outlook 取得发件人地址并回复邮件, 这里使用 Redemption 组件^[4]进行程序开发。

2.4 报表统计模块的设计

县、市级财政局需要统计贫困学生情况, 如贫困学生人数, 分布情况等, 这些统计往往涉及数十万条学生记录。以一个普通的县级财政局为例, 完成一次统计需要 5 分钟左右。如果在网页上直接提交统计请求并下载报表, 将会出现网页超时的错误。为此, 可采用两种方法处理这种情况。

(1) 采用后台服务程序, 在每天凌晨进行全省的数据统计, 由于数量庞大, 必须采用多线程并行执行的形式才能在规定时间内完成, 统计结果写入数据库临时报表, 当用户需要下载报表时, 从临时报表中取出数据, 提高网站的运行效率。

(2) 通过网页异步调用服务程序进行报表统计。使用 WSH 对象^[5], 调用服务器的报表统计程序, 由于程序调用的异步性, 服务器不会等待统计工作完成而立即返回, 达到数据报表统计的事实性。用户只需提交请求, 等待统计完成, 下载报表即可。

最终综合考虑各方面的因素, 在实际应用中采用第一种处理方式。

2.5 基于多 Agent 的分布式系统管理

为了有效地管理 web 服务器群, 使用 Java 开发了基于结构化 P2P 网络的多 Agent 框架^[6], 并基于该框架, 实现了资源信息收集 Agent, 学生信息录入 Agent,

教育券生成 Agent, 网站监控 Agent。其中, 资源信息收集 Agent 在所有需要进行资源信息统计的服务器上运行, 其余 Agent 均只在提供服务的特定服务器上运行, 以维护该服务。图 5 表示多 Agent 管理系统运行状态。多 Agent 分布在 web 服务器群上, 维护各自的系统服务, 用户 Agent 运行在分布式平台管理服务器上, 接受管理员指令, 并负责与相应的 Agent 通信, 以实现特定服务的管理。这种结构简化了对分布式系统的管理。其中, 分布式平台管理服务器为一台逻辑主机, 可以由 web 服务器中任意一台物理主机担当, 且, web 服务器群以结构化 P2P 的网络模式组织在一起, 因此, 具有很强的容错性, 任意主机的失效, 理论上不会对系统造成致命打击。

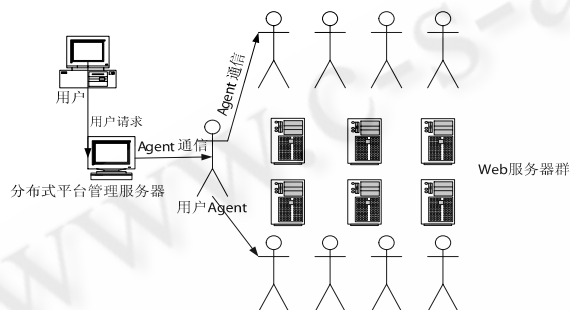


图 5 多 Agent 系统管理示意图

3 应用效果

从 2007 年该系统在浙江省应用的情况看, 该系统运行正常。对于负载最重的学生数据上传, 网站完全可以接受每天 20 万数据的上传负载。在系统运行的负载峰值阶段也能较好地工作, 达到了设计初衷, 可见这样的设计是有效合理的。多 Agent 系统的应用也很好的简化了分布式系统管理的复杂度, 减轻了系统管理的工作量。

4 总结

在各级部门的配合协作下, 2007 年浙江省贫困学

生申报过程已经顺利完成, 标志着本系统的可用性和实用性。本文所总结提出的一些高负载网站设计方法和管理策略, 同时也可以应用于其它各种形式的 B/S 系统。

参考文献

- 1 William R. Vaughn. Visual Basic 与 SQL Server 权威指南. 北京大学出版社, 2000. 55 - 58
- 2 Microsoft Corporation. 全面掌握 Microsoft Office 2000 应用程序开发. 清华大学出版社, 2000. 149 - 191.
- 3 Microsoft Corporation. Microsoft Office XP Developer 开发指南. 北京大学出版社, 2001. 134 - 148.
- 4 Dmitry Streblichenko. Outlook Redemption. (2007 - 7 - 23) [2007 - 12 - 16] <http://www.dimastr.com/redemption>
- 5 马文刚. WSH 实用技术与应用. 清华大学出版社, 2002. 39 - 49.
- 6 王汝传. 智能 Agent 及其在信息网络中的应用. 北京邮电大学出版社 2005. 145 - 166.

(上接第 78 页)

参考文献

- 1 Content Management System. http://en.wikipedia.org/wiki/Content_management_system
- 2 Our Get Fit website, <http://83.138.132.68:8080/index.jsp>
- 3 JBoss official website <http://www.jboss.org>
- 4 JBoss 3.0 下配置和部署 EJB 简介. http://www.huihoo.org/jboss/jboss3_ejb.htm
- 5 陈军, 段辉良. 基于 J2EE 的通用 Web 信息系统框架设计与实现. 计算机系统应用, 2007, 16(1)6 - 9.