

基于.NET 混合模式的高考信息管理平台

College Enrollment Information Management Platform Based on .NET Mixed-Mode

王 锋 张祖平 (中南大学 信息科学与工程学院 湖南 长沙 410083)

肖广军 (湖南省教育考试院 信息处 湖南 长沙 410001)

摘 要: 论文提出一种基于.NET 的、B/S 与 C/S 模式相结合的优化设计模型, 以提供密集型高访问量下的高可用高安全性服务。采用连接池、缓存和索引等数据访问优化方式, 提高了并发访问时系统的整体性能; 采用权限过滤和动态菜单生成机制, 保证不同层次用户访问数据的安全性; 采用灵活的系统自动升级技术, 解决了系统的维护和扩展性问题。实际应用表明, 利用这些优化技术, 较大地提高了系统响应速度, 缓解了服务器压力。

关键词: 高考信息管理平台 混合模式 数据访问优化 权限控制 自动升级

1 引言

随着网络技术的飞速发展, 如何实时有效地采集和管理高考数据, 已成为高考工作的一项重要问题。高考信息的网上采集与发布, 属于访问规模大、访问时间密集且分布严重不均的信息交互^[1]。它面向的用户有考生、家长、老师、县招办、市招办、省招办和院校招办等。大量用户同时访问量可达几十万至几百万次, 这种大规模的并发访问易造成服务器负载过重, 出现无法提供服务等问题。目前, 针对访问过载问题, 文献[1,2]提出借助集群技术, 增强服务器处理能力; 文献[3,4]提出优化系统性能, 提高系统响应速度。

结合当前的 WEB 技术, 及实际参与的湖南省高考、成人高考项目开发工作, 论文提出了基于.NET 混合模式的高考信息管理平台优化设计模型。利用连接复用、数据缓存和索引优化等技术, 提高数据库访问性能; 采用权限过滤和动态菜单生成机制有效地防止未授权用户非法访问, 保证资源访问的安全性; 采用灵活的系统自动升级策略, 保证系统的易维护性和易扩展性。基于混合模式的高考信息管理平台为高考工

作提供了高性能、高安全性的智能化管理平台。论文较全面地分析了平台设计思想及关键技术, 并基于实际系统进行了有效验证。

2 系统架构与功能

平台采用 B/S 与 C/S 模式开发, 客户端(Client)和浏览器端(Browser)通过互联网与服务端(Server)进行通信连接, 如图 1 所示。

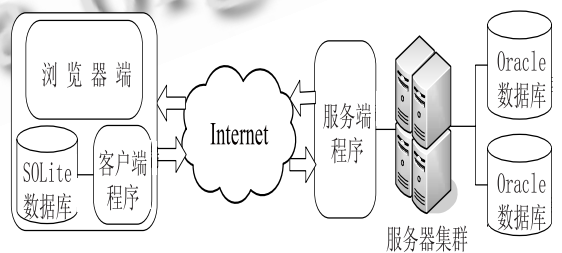


图 1 基于混合模式的高考信息管理平台系统架构

从服务器工作环境角度来看, 采用服务器集群和数据库集群技术, 由多台刀片式服务器构建集群环境, 改善服务器运作性能。服务器集群系统使用八个守候

线程，分别用于处理请求；数据库集群系统使用两个守候线程分别用于处理客户请求、管理认证服务等，以达到负载均衡。

从软件开发平台来看，.NET 与 J2EE 是当前两大主流开发平台^[5]，各有优势。这里选用 .NET，.NET 可以支持高性能桌面应用，以及企业规模快速应用开发。数据库服务器选用大型数据库管理系统 ORACLE 10G。客户端系统数据库选用轻型的开源 SQLite 实现本地数据加密存储，“零安装”，占用资源非常低，但处理速度比 Mysql、PostgreSQL 这两款世界著名的开源数据库管理系统更快，非常适合本地存储。客户端系统联机下载基础数据后，可脱机进行数据存储，操作完毕再批量上传，以缓解服务器过载问题。

平台的功能模块如图 2 所示，是集权限管理、计划管理、报名管理、考务管理、志愿管理、院校端管理、成绩录取管理等为一体的综合信息管理平台。

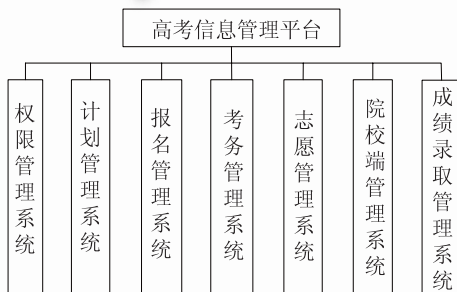


图 2 高考信息管理平台的系统模块

权限管理系统，是整个平台的权限控制中心，主要实现用户管理、权限管理、功能管理、公告管理及日志管理。计划管理系统，是招生工作的指挥棒，主要是用于接收教育部下达的招生计划，产生专业的省标代码，与招生院校进行计划核对交互，生成计划册，统计各类计划情况。报名管理系统，主要完成报名各类信息的采集，各种复杂报表分析统计打印等。考务管理系统，主要完成考点申报、考务数据统计、组考安排、报表打印等。志愿管理系统，主要实现各批次志愿信息采集、打印和确认，信息采集具有快速输入和容错性，保证数据的规范性。院校端系统，主要完成院校与省招办进行计划交互校对，下载基础数据，取消录取名单和查阅公告，是各院校与省招办进行交流的重要渠道。成绩录取管理系统，主要完成成绩库

导入，成绩查询，打印成绩册，录取库导入，查询录取情况，打印录取名册，下载录取数据等。

3 系统关键技术及实现

3.1 数据库存取性能的优化

数据库存取性能是影响系统整体性能的一个“瓶颈”。以下从连接池、缓存和索引三方面出发，阐述这些技术在实际系统中的应用及改进思想。

3.1.1 连接池

频繁建立、释放数据库连接是导致数据库访问效率低的重要原因之一，尤其大数据量访问时更为明显。服务端采用可复用的数据库连接池，是解决这一问题的有效方式。它是将有限的数据库连接资源用资源池模式进行管理，并提供对连接的分配、使用和释放策略。

基本思想：应用系统一启动就在内存中建立一些数据库连接，存储到数据库连接池；当应用程序需要建立数据库连接时，只需从连接池中取一个来用而不用新建；使用完毕后，连接仍然保持而不关闭，只需放回内存即可，从而减少频繁创建对象所带来的开销。

已有系统开发模式中，基本上只采用静态调整配置文件来管理连接池，每次分配一个静态的连接池。这种方式：(1)当连接请求过少时，易造成资源的浪费；(2)当连接请求过多时，易造成资源的不足。作者采用静态预分配和动态自适应调整结合策略，设置了最大限制和恒定的连接数。静态是使用的时候发现空闲连接不够再去检查，动态是使用了一个线程定时监控，根据日志分析优化配置参数。

采用引用计数设计模式来优化连接池管理。具体实现思想为：将连接池分为空闲池和使用池。当连接池初始化时，所有连接均放入空闲池中，并为每一个连接建立一个引用计数来标记使用此连接的用户个数，并给每个连接再加上一个属性值 last_use_time 记录最后被使用时间。当客户请求连接时，首先看空闲池中是否有空闲连接，若有空闲连接则将其分配给客户，并将其移入使用池中，增加其引用计数。若空闲池中无连接，就在使用池中寻找一个引用计数最小的连接给客户。当使用池中引用计数最小的连接个数不只一个，则根据 last_use_time 挑选最大的一个来使用，以提高效率。

3.1.2 缓存

缓存的使用,在高访问量的情况下能够较大地减少对数据库操作的次数,明显降低系统负荷,提高系统性能。

基本思想:首先根据需要定义各块缓存的大小及其所存放的内容,初始时缓存为空;只有被第一次访问时,数据才从数据库读出并存储于缓存;当客户端需要数据时,先到缓存读取信息,缓存根据客户端提供的条件来对应缓存的索引,查找相关信息。如果缓存中有相关信息,则将其返回给客户端;如果缓存中没有,则到数据库中查询,查询后把信息放入缓存,同时返回查询结果到客户端。

由于系统具有时间局部性访问特征,采用 LRU (Least Recently Used)算法进行缓存管理。LRU 算法根据数据的最近访问情况进行缓存管理,在实现时构造一个队列,每个元素表示一个缓存数据项,若缓存新的数据项且缓存不满,把此数据项插入队头(最近访问的数据),如果缓存满则替换队尾的数据项,并把新数据项插入到队头,若访问的数据项在缓存中命中且位于队内,则把该数据项移动到队头。

缓存的使用并不是越多越好,在使用过程中,作者通过限制 Cache 的大小;定时从 Cache 中移走过时的数据;对 Cache 里的数据及时更新;加载常用的数据,提高服务性能。

3.1.3 索引

B/S 与 C/S 模式系统均结合数据访问的频度活动,进行索引的优化。对于条件访问频度较高的字段,适当地建聚簇索引;对非聚簇索引,可权衡索引对加快查询速度和降低更改速度之间的利弊^[2,6]。

定义(活动的频度):任意活动 ar,k 的频度定义为该活动的访问次数与总活动访问次数的比率,即

$$n_{r,k} / \sum_{k \in I} n_{r,k}$$

假设数据访问节点 ar,k 的频度为 freq, 该节点包含 m 个数据访问操作,每个操作对于某列的读写次数分别为 Rs 和 Ws, 则该节点中列的有效读写数值分别为

$$Eff_R_k = \sum_{s=1}^m freq * R_s \quad (1)$$

$$Eff_W_k = \sum_{s=1}^m freq * W_s \quad (2)$$

由式(1)和式(2)得,对于整个数据访问流,某列的有效读写值为流程中所有访问节点的有效读写数值之和,分别为

$$Eff_R = \sum Eff_R_k \quad (3)$$

$$Eff_W = \sum Eff_W_k \quad (4)$$

由于索引对数据的读访问有优化的作用,但也会影响写的速度。因此,在创建索引时可设置 2 个权值: R_Ratio 和 W_Ratio。其中, R_Ratio 表示读访问对索引的需求程度; W_Ratio 表示写访问对索引的排斥程度。因此,列的加权分值为

$$WScore = Eff_R * R_Ratio - Eff_W * W_Ratio \quad (5)$$

根据加权分值的大小来创建非聚簇索引,加权分值越高的列,对索引的需求度越强。而对于条件访问列,其在访问过程中的作用是对该列的值进行给定条件的比较筛选(如=、>、<、group by),只包含对列的读取,不存在读写加权的问题,因此在条件列的筛选过程中,只需考虑有效的访问值。

以高考信息管理平台中志愿库为例,每个考生最多可以填报的志愿数 25 条,而湖南省今年有近 54 万考生,可见志愿库中数据量非常大。根据数据访问频度分析,对(考生号)建聚簇索引,而对(考生号、填报批次代码、批次代码)建非聚簇索引有效地提高了数据访问效率。

3.2 权限控制策略

高考信息管理业务的复杂性决定了简单的权限控制已无法满足应用的需要。根据业务的需求,作者探讨了一种实用的权限控制策略,并将其应用于 B/S 与 C/S 模式相结合的实际系统开发中。

3.2.1 基本思想

目前,流行的权限控制模型有 RBAC(Role-based Access Control)授权模型和 ACL(Access Control List)授权模型^[7]。结合两者优势,提出适合类似业务的通用权限控制模型。权限采用逐级分配原则,由上级将功能下发给下一个级别。每个级别只能操作权限范围内数据,例如级别有省级、市级、县区级、报名点等。每个级别内有不同的组别,例如市级组有长沙市、株洲市等。每个组别包含不同的用户,上一级组别用户根据需要建立下一级组别用户,一个用户可以被分配不同的组别。权限可采用批量授权、指定授权、批量回收、指定回收等方式。

因此,作者在权限数据库设计中,采用了用户表、级别代码表、组别表、权限表、用户组别表、组别权限表、用户权限表,如图 3 所示。

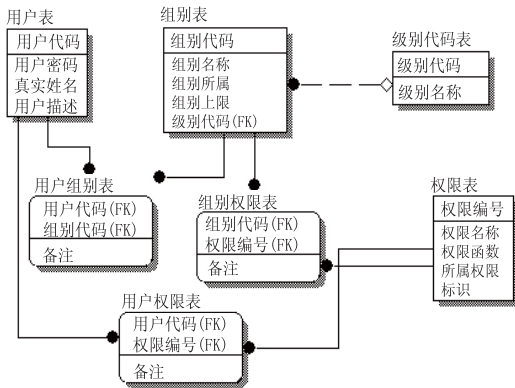


图 3 权限控制 IDEFIX 逻辑结构图

具体用户权限控制分为两层:(1)根据用户权限,动态加载用户特有权限和用户所属组别权限;(2)在单个功能操作过程时,再判断用户是否具有访问权限,有则继续,否则返回出错信息。通过双层过滤机制防止非授权用户恶意攻击。基于功能的权限控制过程中,采用面向对象设计思想,建立通用类:WEB 应用中建立一个 PageBase 基页类,WINFORM 应用中建立一个 FormBase 基窗体类。

3.2.2 技术实现

构建动态系统菜单,是权限控制中的关键技术。动态菜单构建方式:利用树的层次特点,采用前缀相同后缀逐级递增的编码表示节点关系作为其存储结构,一次性将数据表利用 ADO.NET 读入到内存的数据集中,再进行遍历。

权限表中存放权限编号、权限名称、权限函数、权限所属、标识(设 0 表示 B/S 系统、1 表示 C/S 系统)。

(1) B/S 系统动态菜单生成机制

B/S 系统权限存储如表 1 所示。根据用户权限,查询出用户及用户所属组别权限,当使用某功能时,系统根据权限函数,加载对应的页面。

表 1 B/S 系统权限表存储结构实例

Qxbh	Qxmc	Qxhs	Qxss	Bz
1	成绩管理系统			0
1.1	打印		1	0
1.1.1	打印成绩单	Cjgl/Dycjd.aspx	1.1	0

(2) C/S 系统动态菜单生成机制

C/S 系统权限存储如表 2 所示。现有的系统基本上是通过菜单的 Enable 属性来进行权限控制,不利于系统扩充。作者利用反射技术解决动态菜单生成问题。

表 2 C/S 系统权限表存储结构实例

Qxbh	Qxmc	Qxhs	Qxss	Bz
2	报名管理系统			1
2.1	打印		2	1
2.1.1	打印报名表	Bmgl.FrmPrint	2.1	1

反射是 .NET 的重要机制,通过反射技术可以动态获取每一个类的成员。根据权限表读入命令项,动态生成时利用反射得到各命令项所对应 WINFORM 类。关键源码如下:

```
//分解命令项,利用反射机制创建类实例
string[] assemblyName=formName.Split('.');
Assembly asm=Assembly.Load(assemblyName[0]);
object frmObj=asm.CreateInstance(formName);
Form frms=(Form)frmObj;
frms.MdiParent=fm;frms.Show();
```

动态生成菜单功能项,便于系统功能的扩充,如有新模块功能只需将模块功能名按指定规则加入表中即可读出,为用户权限分配、系统灵活扩展提供方便。

3.3 系统自动升级技术

系统自动升级是混合模式开发的一大难点,尤其是 C/S 模式下的客户端如何以较小的代价来控制其系统文件的维护变更。

对于 B/S 模式,升级维护只需在服务端进行即可,操作过程:将修改的页面及重新编译过的动态链接库拷贝到 web 工作目录相应路径。对于 C/S 模式,作者针对客户端自动升级问题给出一种通用的实现策略:(1)创建一个数据表,用来存放客户端文件编号、文件名及版本号;(2)当客户端系统有改动时,只需更改服务器数据表中文件版本号,再将更新后的 DLL 动态链接库等文件拷贝到服务器指定更新包目录下;(3)C/S 客户端,当选择联机与服务器交互时,系统将自动检测服务器数据表是否与本地库中一致,不一致则自动下载并更新本地库中文件。为保证客户端与服

务器的一致性,当服务器有更新时,系统自动提示有更新需重新下载。自动升级的巧妙实现,简化了升级流程,使得客户端系统维护更加简便快捷。

4 结束语

论文提出了基于.NET 混合模式的高考信息管理优化设计模型,并通过实践证明了优化后的高考信息管理平台的优越性。采用连接池、缓存和索引等数据库性能优化方式,提高了并发访问时平台的整体性能;采用权限过滤和动态菜单生成机制,保证不同层次用户访问数据的安全性,且出现功能扩展时具有灵活的可扩展性;采用灵活的系统自动升级技术,为分布式系统的维护和扩展提供了方便。该平台在湖南省2008年高考的实际应用及在西藏高考的应用推广表明其极大地提高了工作效率,完善了用户的实际需求,加快了处理速度,较大地缓解了服务器负载。该平台具有较强扩展性、灵活性、可重用性和可移植性,其中采用的优化技术,同样适用于其他密集型高访问量要求系统,对于相似信息化建设有一定的推广价值。

参考文献

1 邱钊,卢春燕,陈明锐,黄俊.普通高考招生信息网站的负载均衡.计算机工程与应用,2007,43(11):213 - 218

2 Lian J, Naik K, Agnew GB. A framework for evaluating the performance of cluster algorithms for hierarchical networks. IEEE Transactions on Networking, 2007,15(6):1478 - 1489.

3 施智平,胡宏,李清勇,史俊,史忠植.视频数据库的聚类索引方法.计算机学报,2007,30(3):397 - 404.

4 Zirkind P. Genetic database optimization: how data inspection and consideration, provides for index compression and record access optimization of genetic databases. ACM, 2006:68 - 76.

5 Rudzki J, Systs T. Performance implications of design pattern usage in distributed applications: case studies in J2EE and .NET. Proceedings of the ISSTA 2006 Workshop on Role of Software Architecture for Testing and Analysis. ACM, 2006:1 - 11.

6 傅向华,冯博琴,王小民,王志强.一种基于数据访问流的数据库索引优化方法.计算机工程, 2007,33(12): 99 - 201.

7 Chen L, Crampton J. On spatio-temporal constraints and inheritance in role-based access control. Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security.