

C/S 结构监测信息管理与分析系统的关键问题研究

吴振君, 王浩, 王水林, 汤华

(中国科学院武汉岩土力学研究所岩土力学重点实验室, 湖北 武汉 430071)

摘要: 针对水电站地下厂房监测的信息化、共享化要求, 开发了基于 C/S 结构的地下厂房监测信息管理与分析系统。介绍了系统的结构框架, 着重探讨了基于 C/S 结构的管理系统实现的关键问题, 如数据库安全性控制、多用户并发操作、网络环境下的执行效率、数据库恢复技术以及数据库服务器安全性等问题。系统具有完善的监测数据、属性数据、施工进度信息和工程附件等管理、分析功能, 采用 GIS 技术实现附件和监测、属性数据的双向可视化查询, 在网络环境下可高效运行, 为水电站地下厂房安全监测信息化管理提供了有效的解决方案。

关键词: 地下厂房; 监测信息 C/S 结构; 管理系统; 并发控制; 数据库安全

作者简介: 吴振君(1977-), 男, 博士研究生, 主要从事岩土工程监测、滑坡灾害方面的研究工作。

0 引言

岩土工程监测信息管理系统现在已经广泛应用于一些大型岩土工程监测项目^[1-4]。监测部门的协作、监测数据共享以及监测信息及时反馈都对管理系统的开放性和网络化提出了新的要求, 因此开发基于网络环境的监测信息管理系统非常必要。目前常见的网络环境下的岩土工程监测信息管理系统, 主要实现方式是 C/S (客户/服务器) 结构、B/S (浏览器/服务器) 结构, 以及 C/S、B/S 混合结构^[5-7], 但这些管理系统均以考虑监测信息的发布为主, 未涉及网络条件下数据库安全性控制、多用户并发操作、运行效率、数据库恢复技术以及数据库服务器安全性等关键问题。本文介绍了作者开发的基于 C/S 结构的水电站地下厂房监测信息管理与分析系统, 解决了网络版开发中的关键技术问题。

1 网络模式选择

C/S 结构, 数据库存储在远程的服务器, 在客户机上运行软件。它采用两层结构: 前端是客户机, 即用户界面接受用户的请求, 并向数据库服务提出请求, 后端是服务器, 将用户请求结果返回给客户端, 客户端将数据进行分析计算并将结果呈现给用户。这种结构的特点是交互性强、具有安全的存取模式、网络通信量低、响应速度快、利于处理大量数据。

B/S 结构, 安装维护一个服务器, 客户端采用浏览器软件。它是随着 Internet 技术的兴起而发展起来的, 主要利用 WWW 浏览器技术, 结合脚本语言和 ActiveX 等技术, 是一种全新的软件系统构造技术。但 B/S 模式执行效率低, 交互性较差, 并且数据库的安全问题尚无法完全解决。

SQL Server 是 Microsoft 公司推出的一个优秀的数据库管理系统, 由于其优良性能、灵活和可伸缩性、便捷的可管理性和强大的可编程性, 已成为众多应用程序开发的首选数据库系统。SQL Server 所提供的 Transact-SQL 使客户端能通过多种方法访问服务器上的数据, 极大地提高了

应用程序的实用性。

由于水电工程监测需要处理大量的数据, 数据库安全性也需要保证, 因此考虑在 SQL Server 数据库基础上, 开发基于 C/S 结构的地下厂房监测信息管理与分析系统。

2 系统结构设计和功能

2.1 系统结构设计^[8]

(1) C/S 结构。系统软件可运行于 intranet 和 internet 两种网络环境, 需要实现数据库安全性控制, 考虑多用户并发操作、运行效率、数据库恢复技术以及数据库服务器安全性等问题。

(2) 数据库设计。数据库结构设计是系统的核心, 为有效管理水电工程监测对象的层次关系, 设计了工程项目、标段、单元工程、断面、组、测点类型、测点以及掌子面等八种对象来描述和组织数据库。每种类型的监测数据都存储于一张表中, 通过关键字段和属性、图形数据建立关联, 并逐层向上和父级对象联系起来, 这样每个测点都有着明确的从属关系。所有的数据都存储在一个数据库中, 为网络实现提供了方便。

(3) 系统功能模块化。对不同的监测仪器, 提供相应的数据输入、处理、查询、图形和报表输出模块。

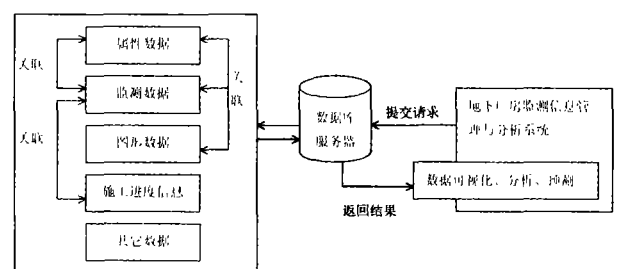


图 1 系统的结构框架

系统的简单结构框架见图 1。开发平台是 Windows

收稿日期: 2006-07-30

XP, 以 Visual C++6.0、MFC、ADO 为开发工具, 后台数据库采用 SQL Server 2000, 可运行于 Windows ME/2000/XP/Server 等环境。

2.2 系统的主要功能

系统按功能可以划分为如下 6 个模块。①数据库管理模块, 主要包括监测数据、属性数据、图形数据、施工进度等相关数据、信息的管理。②数据处理模块, 包括各种仪器的原始监测数据整编、数据初步统计分析等功能。③数据分析预测模块, 用常用的预测模型对数据建模并预测。④数据库查询模块, 按查询方式不同分为 3 种: 数据查询, 属性查询, 工作量查询。⑤图形可视化模块, 可以导入现有的 CAD 工程图件, 图形对象可以和监测对象建立 GIS 关联, 从而实现可视化定位、查询。⑥报表自动生成模块, 可以生成特定日期范围的包含监测数据和过程曲线的报表和统计分析报表。

3 C/S 结构地下厂房监测信息管理与分析系统实现的关键问题

3.1 数据库的安全性

数据库的安全性是指保护数据库以防止不合法的使用所造成的数据泄密, 更改或破坏。在作者开发的地下厂房监测信息管理与分析系统中因为大量数据集中存放于数据库服务器中, 并为多用户直接共享, 从而使安全性问题更为突出。数据库的安全性主要包括用户管理、身份验证和存取权限控制两方面。设置 SQL Server 用户采用 SQL Server 和 Windows 混合验证模式。设计用户表及用户权限分配表, 属于某一组的用户拥有该组的所有权限。把系统内的用户分成三级, 即 3 个用户组: 系统管理组、数据操作组、访客组。用户分别隶属于相应的组。只有管理员具有用户管理权限, 可以在系统内部新建, 删除用户以及修改用户权限。具体的实现方法是 (见图 2): 创建用户, 在用户表中增加用户, 同时创建和用户名同名的数据库登录, 然后为该登录分配服务器角色, 授予数据库访问权限, 再设定相应的数据库角色, 数据库角色功能非常强大, 可以精确地控制角色下用户对各个表、存储过程的访问权限。另外, 因为用户表和权限分配表存储了重要的信息, 需要限制非管理员级别用户访问。这样就可以实现 SQL Server 和系统软件的双重验证: 一方面, 用户的权限在软件内部做了限制, 不同级别的用户具有不同的软件界面, 防止越权操作; 另一方面, 用户在 SQL Server 上也只具有被授予的权限, 保证了数据库的安全性。图 3、4 分别是系统登录和用户管理界面。

3.2 多用户并发操作问题

为了充分利用系统资源, 发挥数据库资源共享的特点, 应该允许各个用户并行地存取数据, 提高系统效率。但这样就会产生多个用户并发存取同一数据的情况。若对

并发操作不加控制就可能导致存取不正确的数据, 由于相互的干扰和影响, 可能引发错误的结果, 从而破坏数据库的一致性。并发操作带来的数据不一致性包括 3 类: 丢失更新、不可重复读和读“脏”数据。目前常用地并发控制技术主要有:

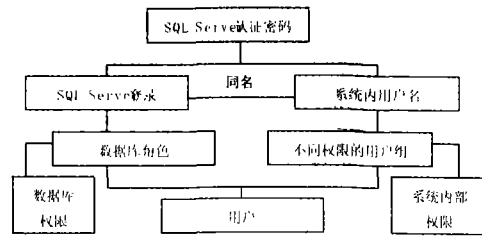


图 2 系统的用户验证模式

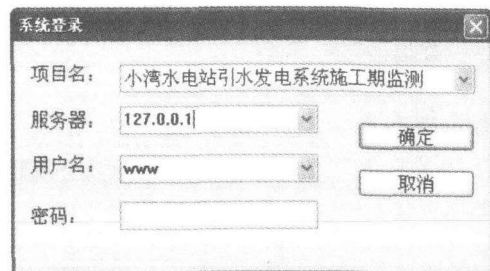


图 3 系统登录界面

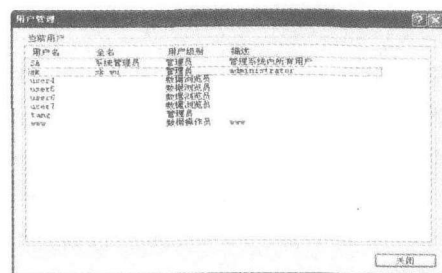


图 4 用户管理

3.2.1 事务和锁

事务是用户定义的一个数据库操作序列, 这些操作要么全做要么全不做, 是一个不可分割的工作单位。锁是为了防止其它事务访问指定资源的一种手段, 是实现并发控制的一个非常重要的技术。所谓锁即是在一段时间内禁止某些用户对数据对象做某些操作, 以避免产生数据的不一致性问题。即事务 T 在对某个数据对象, 如表、记录等进行操作之前, 先向系统发出请求, 并对其加锁。加锁成功后, 事务 T 就对数据对象有了一定的控制权, 在事务 T 释放它的锁之前, 其它的事务就不能更新此数据对象。SQL Server 遵从三级锁协议, 从而有效的控制并发操作可能产生的丢失更新、读“脏”数据、不可重复读等错误。

3.2.2 软件内部控制

(1) 规定数据录入人员只能修改自己所建立的记录, 那么就不会出现并发操作中的各种错误, 因为这时各个不同的用户所能更新的记录不会发生重合。这种情况下, 需

要在数据库表中增加用户列。在用户浏览记录时, 将用户列作为一个过滤条件, 对应用程序的 SQL 语句做相应的调整。但这种策略的作用有限, 因为在很多情况下, 并发控制不可避免。

(2) 仿照锁的基本思想调整应用程序和数据库结构, 在需要进行并发控制的数据库表中增加一个锁字段, 这个字段可以是一个布尔型变量。当查询这个表时, 可以修改表的记录, 为了防止其它在该用户编辑某记录期间修改这个记录, 那么就需要客户在浏览到该记录的数据时, 给该记录加锁, 修改完毕后释放锁。别的客户要修改这个表的记录的话就先检测一下该记录有没有被加锁, 如果已经加锁, 则不能进行修改。如果锁字段空闲, 那么首先给该记录加锁, 然后取记录给客户浏览、编辑, 在此期间别的客户不能修改记录。这就很有效的防止了丢失修改。这种方法很常用, 但是存在并发度低的缺点。

3.2.3 时间戳技术

SQL Server 的 Timestamp 提供了解决并发控制的机制: 如果用户试图修改某一行, 则此行的当前 Timestamp 值会与最后一次提取此行时获取的值进行比较。如果值发生改变, 则服务器就会知道其他用户已更新了此行, 会返回一个错误, 更新不成功。如果值是一样的, 服务器就执行修改。

Timestamp 是一个数据类型, 它是一个二进制数字, 表示数据库中更改的相对顺序。每个数据库都有一个全局当前时间戳值: @@DBTS。当用户更改带有 timestamp 列的行时, SQL Server 先在时间戳列中存储当前的 @@DBTS 值, 然后增加这个值。如果某个表具有 timestamp 列, 则时间戳会被记到行级, 它的值会随着表的更新操作变化, 可以比较某行的当前时间戳值和上次提取时所存储的时间戳值, 从而确定该行是否已更新。

由于事务和锁机制可能造成死锁问题, 需要进行逻辑控制, 而时间戳技术则不存在这个问题, 结合系统软件高度并发性的需求, 系统内部采用时间戳技术来实现并发控制。

3.3 C/S 结构下的系统运行效率问题

在网络环境下, 所有查询请求都是提交到数据库服务器上进行处理, 如何减少数据传输、提高系统的运行效率是一个非常重要的问题, 下面简要介绍一下作者开发过程中采用的方法:

(1) 采用 TCP/IP 连接方式而非命名管道, 因为在局域网环境下, 两种连接方式性能基本相当, 但在广域网下 TCP/IP 连接的效率明显优于命名管道。

(2) 根据查询条件, 建立索引、优化索引、优化访问方式, 限制结果集的数据量。注意填充因子要适当 (最好是使用默认值 0)。索引应该尽量小, 使用字节数小的关键字

段建索引。

(3) 查询时只返回必要的行、列, 以减少数据传输。

(4) 因为开发的管理系统是基于 GIS 的, 其中包含大量的工程图件, 图件本身占用空间就比较大, 如果处理不好, 将极大地降低系统的效率, 这也是大量数据传输中的常见问题。在数据库内用 Image 字段保存这些图件, 插入二进制图件到 Image 列, 采用存储过程来处理, 而不用内嵌的 Insert 来插入, 因为这样软件内首先将二进制值转换成字符串 (大小是二进制文件的两倍), 服务器接收到字符后又将其转换成二进制值, 而存储过程就没有这些动作, 方法是: Create procedure SP_insert as insert into table (图形文件表) values (@PicFile), 在客户端调用这个存储过程传入二进制参数, 这样处理速度明显改善。

(5) 没有必要时不使用 DISTINCT 和 ORDER BY 语句, 这些操作可以改在客户端执行, 因为它们增加了额外的开销。

(6) 尽量将数据的处理工作放在服务器上, 广泛地使用存储过程, 以减少网络传输的开销。存储过程是编译好、优化过的存储在数据库中的 SQL 语句, 运行速度很快。反复执行的动态 SQL, 可以使用临时存储过程, 该过程 (临时表) 被放在 Tempdb 中。

3.4 数据库恢复技术

由于电脑硬件的故障、管理系统软件的错误、操作人员的失误以及恶意的破坏是不可避免的, 这些故障则造成运行事务非正常中断, 影响数据库中数据的正确性, 重则破坏数据库, 使全部或部分数据丢失, 因此系统必须具有把数据库从错误状态恢复到某一已知的完整状态的功能, 这就是数据库的恢复。为保护数据免受意外的损失, 需要定期对数据库进行备份操作, 作为数据库恢复的依据。系统内部集成了数据库备份和恢复操作, 另外, 基于网络的备份也是非常重要的, 下面是把数据库备份到指定客户机的存储过程部分代码:

```
-- 备份设备名称
DECLARE @DeviceName VARCHAR(128)
SET @DeviceName = @UserName + '@' + @RemoteIP + '\'+
@ShareName + '\'+ @SharePath + '\'+
-- 备份设备路径
DECLARE @DevicePath VARCHAR(512)
SET @DevicePath = '\\'+ @RemoteIP + '\'+ @ShareName +
'\'+ @SharePath + '\'+ @BackupFile
-- 添加备份设备
EXEC @Result = Sp_AddumpDevice 'Disk', @DeviceName,
@DevicePath
-- 添加共享连接命令
DECLARE @AddShare VARCHAR(512)
SET @AddShare = 'NET USE \\'+ @RemoteIP + '\'+
```

```
@ShareName + ' ' + @Password + ' /USER:' + @UserName +
'@' + @RemoteIP
EXEC @Result = xp_cmdshell @AddShare
--备份数据库
BACKUP DATABASE @Database TO @DeviceName
```

3.5 SQL Server 数据库服务器的安全问题

3.5.1 修改默认监听端口

SQL Server 默认安装情况下监听 TCP 端口 1433。修改默认端口为其它如 8077, 这样可以避免端口扫描软件扫描到电脑上开启的 SQL Server 服务, 达到隐藏服务的效果, 减少被攻击的可能。默认端口改变后, 需要修改数据库连接字符串, 在其中添加服务监听的端口, 否则无法连接到服务器, 具体修改方法如下: “Provider=sqloledb;Network library=DBMSSOCN;Initial catalog=DatabaseName;UID=UserName;PWD=PassWord;Data Source=数据库服务器 IP 地址, 端口”。

3.5.2 加装补丁

SQL Server 2000 最新的补丁是 SP4, 主要更新了 32 位数据库组件, 包括数据库引擎、复制、客户端连接组件和工具以及升级了 MSDE, 使数据库服务器运行更加安全。

3.5.3 使用 NTFS 文件系统

NTFS 是最适合安装 SQL Server 的文件系统。它比 FAT 文件系统更稳定且更容易恢复, 而且它还包括一些安全选项, 例如文件和目录 ACL 以及文件加密 (EFS)。

4 结 论

本文介绍了开发的基于 C/S 结构的地下厂房监测信息管理与分析系统, 简要介绍了其结构设计框架和功能, 着重探讨了 C/S 结构系统软件开发中的关键技术如数据库安全性控制、多用户并发操作、网络环境下的执行效率、数据库恢复技术以及数据库服务器安全性等问题:

(1) 用户分级管理, 采用软件和 SQL Server 联合控制用户的存取权限, 可以有效保证数据库的安全性。

(2) 分析了各种并发控制技术的优缺点, 采用时间戳技术来实现多用户控制。

(3) 介绍了提高系统运行效率的方法, 采用 TCP/IP 的数据库连接方式, 通过索引、优化查询以及使用存储过程等方法来提高运行效率。

(4) 系统内集成了数据库恢复和备份技术, 提高了数据库的抗风险能力, 并实现基于网络备份数据库。

(5) 通过修改 SQL Server 默认监听端口、加装软件补丁以及选用 NTFS 文件系统提高数据库服务器的安全性能。

参考文献:

- [1] 顾冲时, 吴中如. 大坝安全监测专家系统的理论及其应用[J]. 水利水电科技进展, 1997, 17(3): 35 - 40.
- [2] 李元海, 朱合华. 岩土工程施工监测信息系统初探[J]. 岩土力学, 2002, 23(1): 103 - 106.
- [3] 徐竹青, 酆能惠. 上石坝安全监测分析评价预报系统[J]. 水利与建筑工程学报, 2003, 1(4): 9 - 12.
- [4] 徐 伟, 李文涛, 丁勇敏. 清河水库大坝安全监测数据管理及建模分析系统[J]. 水电自动化与大坝监测, 2005(2): 42 - 44.
- [5] 李东亚, 王桂寅. 客户/服务器结构下的小浪底水环境监测信息系统[J]. 分析仪器, 2002, 2: 9 - 11.
- [6] 陈振飞, 邓昌铁. 基于 Intranet/Internet 的大坝安全监测信息查询系统[J]. 水电自动化与大坝监测, 2002, 26(1): 54 - 56.
- [7] 顾永明, 郑东健, 于鹏. C/S 与 B/S 混合模式下的大坝安全监测信息管理系统[J]. 水利与建筑工程学报, 2005, 3(2): 16 - 18.
- [8] 王 浩, 吴振君, 汤 华, 等. 地下厂房监测信息管理、预测系统的设计与应用[J]. 岩土力学, 2006(1): 163 - 167.