

城市路灯远程监控系统中服务器端软件的设计与实现

丁 亮, 秦会斌*, 吴红生

(杭州电子科技大学 新型电子器件与应用研究所, 浙江 杭州 310018)

摘要: 针对城市中大规模的路灯远程监控系统, 设计开发了基于C/S与GPRS远程路灯监控系统的服务器端软件。为了解决同时和大量的客户端进行并发的网络通信问题, 将微软的完成端口(IOCP)模型技术应用到了软件的网络通信模块中, 并且采用了多线程机制。在利用ADO技术对Microsoft Access 2003数据库进行读写操作的数据处理模块中, 采用了模块编程后进行分层的思想, 把相同或相似的功能封装成模块, 按照功能的调用划分层次, 使模块间的调用变得有序, 以尽可能降低代码的冗余量, 节约空间。性能测试结果表明, 所设计的服务器软件能够承载大量的客户端, 具有较高的数据处理能力, 在实际项目测试运行中也表现出良好的性能。

关键词: 城市路灯; 远程监控系统; 服务器端软件; 动态数据对象

中图分类号: TP277; TP368.5; TM923

文献标志码: A

文章编号: 1001-4551(2012)12-1498-05

Server software in remote monitoring system of city-street-light

DING Liang, QIN Hui-bin, WU Hong-sheng

(Institute of Electron Device & Application, Hangzhou Dianzi University, Hangzhou 310018, China)

Abstract: Aiming at the large-scale remote monitoring system of city-street-lighting, a set of server software was designed in road lighting control system based on C/S mode and GPRS networking. In order to solve a large number of clients communicating via the Internet with the server at the same time, Microsoft's completion port communication technology and multi-threading mechanism were applied in the network communication module of the server software. Modules programming and hierarchical idea were introduced in the data processing module which uses the active data objects (ADO) technology access to the Access database. The same or similar functions were packaged into a module and were divided into hierarchies according to function calls in order to make calls between modules become the order. The amount of code redundancy could be minimized and memory space would be saved. Performance tests results indicate that the server software can support numbers of clients and has a high data processing capability, and it has good performance in the actual project testing.

Key words: city-street-light; remote monitoring system; server software; active data objects(ADO)

0 引 言

路灯作为现代城市的基础设施, 在城市的日常运作和人们的生活中起着重要作用, 路灯系统的先进程度标志着一座城市的现代化水平^[1]。基于C/S模式和GPRS的城市远程路灯监控系统主要由3部分组成: ①监控操作的客户端; ②保存数据并进行命令转发的服务器; ③道路两旁的多台远程监控终端。对负载要求

较轻的服务器采用每个客户单线程的模型, 服务器创建一个监听线程, 然后为每一个到来的连接请求派生一个工作线程。但是当客户端增多时, 会发生数据的阻塞和丢失, 严重时将会导致服务器端的崩溃。系统在引入微软的完成端口模型后能够大大地提高服务器的稳定性和快速进行数据处理、传输的能力^[2-3]。

本研究采用完成端口网络通信模型和ADO数据库访问技术, 在Visual C++ 2008平台上开发服务器端

收稿日期: 2012-06-07

作者简介: 丁 亮(1987-), 男, 湖北黄冈人, 主要从事电路与系统方面的研究。E-mail: dingliang410@163.com

通信联系人: 秦会斌, 男, 教授, 硕士生导师。E-mail: qhb@hdu.edu.cn

软件。

1 远程路灯控制系统的工作原理

整个远程路灯控制系统的示意图如图1所示。

服务器固定IP地址,客户端通过Internet网络连接服务器,在道路现场的远程监控终端上通过GPRS模块与服务器相连,远程终端可以对每一路路灯的电参数(电压、电流、功率等)测量。另外,远程终端上集成Zigbee模块,与单个路灯控制节点进行通信,完成对单个路灯的开关及调光控制,从而实现客户端对整个路灯系统的远程监控。

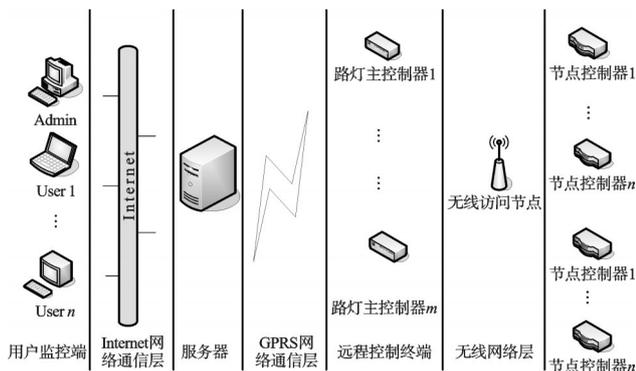


图1 远程路灯控制系统

2 服务器端软件的设计

该系统中服务器端软件的主要作用是充当客户端和远程终端的数据中转站,负责将客户端的控制命令转发给远程的监控终端和响应客户端的查询信息命令,保存整个系统的配置信息和远程终端采集的电参数。难点在于:系统在可以与多个客户端和远程终端上的GPRS模块进行网络通信的同时,须能够对数据库进行读写操作,网络通信模块和数据库读写模块集成在同一个软件中。服务器的整个架构如图2所示,大致分为以下几个设计步骤:

- (1) 利用完成端口技术实现服务器与客户端软件和远程控制终端GPRS模块之间的网络通信;
- (2) 整个监控系统采用TCP协议进行数据传输,在此基础上制定监控系统的通信协议。服务器软件

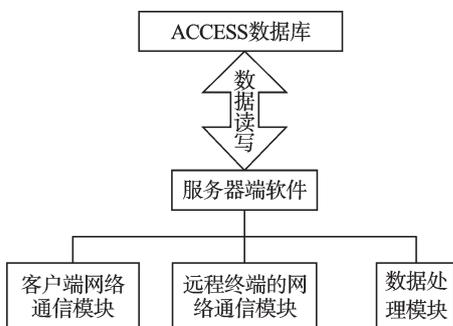


图2 服务器架构图

根据通信协议来分析接收到数据报文,从而对客户端做出响应或者向远程控制终端发送相关的控制指令;

(3) 利用先进先出的队列技术来实现服务器内部的客户端网络通信模块和远程控制终端网络通信模块之间的信息交互;

(4) 在业务逻辑层利用ADO技术来实现对ACCESS数据库的读写操作来保存和获取相关数据。

3 服务器端软件的实现

3.1 完成端口的简介

服务器的网络通信模块是实现整个服务器软件的关键,笔者在此采用完成端口模型。当应用程序必须一次管理多个套接字时,完成端口提供了最好的系统性能。一个完成端口其实就是一个完成I/O的通知队列,由操作系统把已经完成的重叠I/O请求的通知放入这个队列中,按照先进先出的方式进行提取。当某项I/O操作一旦完成,对该操作结果进行处理的工作线程就会收到一则通知,再去完成其他的工作^[4]。这个模型提供了最好的伸缩性,它非常适合用来处理成百上千的套接字。

3.2 完成端口通信模型的实现

程序中网络通信模块使用多线程机制,系统先创建完成端口、接着创建监听线程以及服务线程。在监听线程中系统将完成端口与套接字关联,发起初始的重叠I/O操作,最后在服务线程中等待完成端口的操作结果,处理到达的I/O完成通知。完成端口网络通信模块的主要的工作流程图如图3所示,主要的完成端口API函数和数据结构的设计介绍如下:

(1) 主线程中调用CreateIoCompletionPort函数创建完成端口,并以该I/O完成端口为参数创建多个服务线程。本研究设定该函数的第4个参数为0,从而保证每个处理器仅运行一个服务线程来为完成端口服务,以避免线程上下文切换。这样避免带来资源的耗费,同时保留了线程池模型的优势;另外,在监听线程中也是用该函数来完成Socket套接字句柄与完成端口句柄进行关联,从而使完成端口能够感知发生在该Socket上的事件。

(2) 向完成端口关联套接字句柄后,研究者便可以在套接字上投递重叠发送和接收请求处理。完成端口以先进先出的方式为这些封包排队。本研究在服务线程中使用GetQueuedCompletionStatus函数取得这些队列中的封包,并通过其参数得到传输的字节数量和与套接字句柄关联的唯一数据等信息。

(3) 当有多个线程在套接字上执行I/O操作的时候

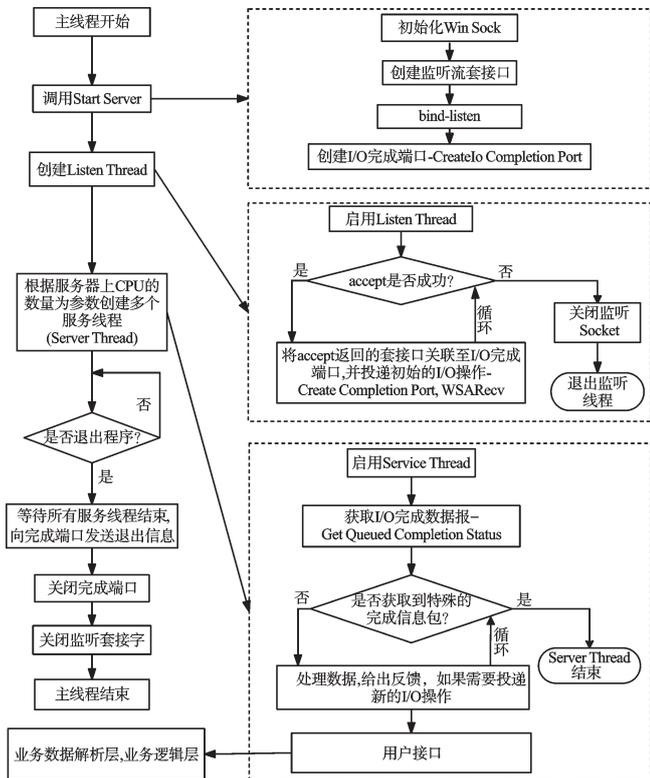


图3 完成端口通信模块流程图

时候,为了避免在重叠操作正在进行时释放它的操作数据,本研究采用的解决方法是在每个套接字句柄上调用 closesocket 函数—所有的未决的重叠 I/O 操作都会完成。一旦所有的套接字句柄关闭,就该终止完成端口上处理 I/O 的服务线程了。通过调用 PostQueued-CompletionStatus 函数向服务线程发送特定的完成封包,这个完成封包通知服务线程退出。

(4) 程序使用 _OverLappedEx 结构来描述 per-I/O 数据,它包含了在套接字上处理 I/O 操作的必要信息。提交 I/O 时,提交的便是 _OverLappedEx 对象。它的定义如下:

```
typedef struct _OverLappedEx{
    OVERLAPPED OverLapped;
    WSABUF wbuf;
    char data[MAX_BUF_LEN];
    IO_OPER oper;
    DWORD flags;
}PER_IO_DATA,*LPPER_IO_DATA;
```

该结构体是扩展的重叠结构,单 I/O 数据。

其中:wbuf 是 I/O 操作的数据对象;data 是实际的数据缓冲区;oper 用于标志 I/O 操作的类型,IO_OPER 枚举型,可以是 SVR_IO_READ 或者是 SVR_IO_WRITE;flags 用于设定或者返回 I/O 操作的标志。

(5) 为了保存每个客户端的连接信息,本研究创建了单句柄数据结构。其定义如下:

```
typedef struct _CONN_CTX{
    SOCKET sockAccept;
    LPPER_IO_DATA pPerIODData;
    struct _CONN_CTX *pPrec;
    struct _CONN_CTX *pNext;
}CONN_CTX,*LPCONN_CTX;
```

其中:sockAccept 是该连接的服务器端服务套接口;pPerIODData 指向该链接的 I/O 操作信息;pPrec 和 pNext 用于形成服务器当前所有连接信息的双向链表,指向链表中的前一个节点和后一个节点。

在该系统中有多个服务线程在同时为客户端提供服务,每个服务线程都处于一个循环,首先通过传输的字节数来决定是否删除已关闭连接的客户端,接下来根据 I/O 操作类型来判断是接收数据或发送数据。服务线程开始工作后有 3 种状态:

(1) 等待状态。如果没有在完成端口上接收到 I/O 操作完成通知包,服务线程将会处于在完成端口上的等待状态;

(2) 服务状态。当接收到 I/O 操作完成通知包时,它将会按照 LIFO 方式被唤醒,为到来的客户端提供服务。完成服务后回到完成端口继续等待;

(3) 阻塞状态。为客户端提供服务时,调用 sleep 函数,线程进入阻塞状态。

3.3 ADO 访问 Access 数据库技术简介

Microsoft Office Access 2003 是由微软发布的关联式数据库管理系统。ADO(ActiveX Data Object)是 Microsoft 数据库应用程序开发的新接口,是建立在 OLE DB 之上的高层数据库访问技术,ADO 最大的优点是使用简便、速度快、内存消耗少和占磁盘空间少^[5-7]。

ADO 库包含了 3 个基本的数据库访问接口:_ConnectionPtr 接口,_RecordsetPtr 接口以及 _CommandPtr 接口。它们分别对应 ADO 的 3 个对象:① Connection 对象,即数据库连接对象。它提供了交换数据所必须的环境,可以利用它来访问数据源和进行事务处理;② Recordset 对象,即字符集对象。该对象代表从数据库的表中得到记录集或命令执行的结果,它是在程序中检查和修改数据的最主要的方法;③ Command 对象,即命令对象,该对象是对数据源进行操作的命令定义,通过该对象可以在数据源中添加、删除或更新数据^[8]。

3.4 数据处理模块的实现

根据需求分析,该系统设计了 8 个表,分别是:管理员信息表 AdminInfo;用户信息表 UserInfo;管理信息区域信息表 AreaInfo;区域下所配置的远程终端信息表 RemoteTerminalInfo;远程终端下所有路的信息表

RoadInfo;路下所有灯的信息表LightInfo;保存报警信息的表WarningInfo;保存开关灯策略的表StrategyInfo。为了方便在服务线程中对数据库里的信息的读/写,本研究将整个数据库访问的模块进行划分,如图4所示,每个模块对应于程序中相应的类的设计及其方法的实现。

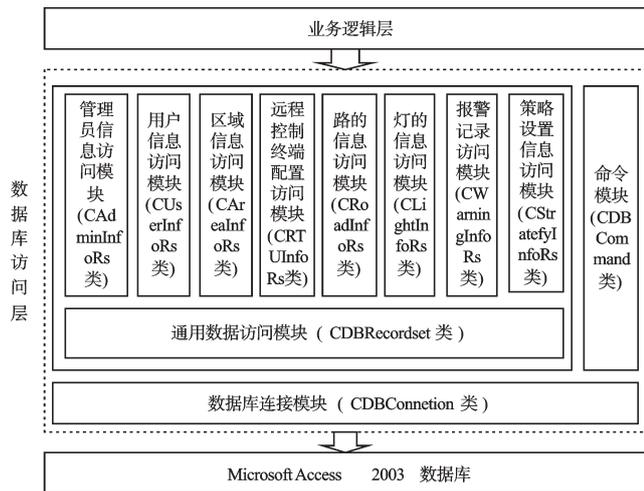


图4 数据库访问模块的划分

ADO是COM(Component Object Model,组件对象模型)对象^[9],使用了很多_variant_t类型的变量。这些变量使用起来非常麻烦,大大降低了程序开发的效率。因此研究者在用ADO对Access数据库进行访问的编程设计中,有关模块再次被细分为数据库连接模块、数据库访问模块以及数据库命令模块。它们分别对应着ADO的Connection对象、Recordset对象和Command对象常用方法的封装。

数据库连接模块是对ADO的Connection对象的封装。该模块使用Visual C++常用的数据类型^[10](如CString)和外界进行交互。另外,系统将数据库连接模块单独提出后,定义一个数据库连接对象,从而在整个应用程序中,只需要一个数据库连接。

在数据库访问模块中,最基础的是“通用数据库访问模块”,该模块是对ADO的Recordset对象的封装。而数据库中每张表的访问模块是更上一层的数据库访问模块,这些模块是根据数据库中表的结构设计的,为表中每个字段的访问都提供了访问的接口。它们在“通用数据库访问模块”的基础上进行继承和扩充,将这些操作封装成模块而不是写到程序的各处,最大的

优点是以后数据库的变更带来了方便。

例如“用户信息表”中存在字段“Name”,那么“用户信息访问模块”就会提供“GetName”和“SetName”这两个方法,供外部访问“Name”这个字段。这样做最大的好处是:当用户修改了数据库字段的属性时,例如将“Name”改为“UserName”,以后修改时不需要到程序各处去修改访问出现“Name”的每个语句,而只需要修改“GetName”和“SetName”两个方法。

由于数据库访问模块对数据库中表间的关系及数据库的增加、修改和删除等操作的支持很弱,为了解决这个问题,本研究设计了对ADO的Command对象的封装。它的主要作用就是完成那些数据库访问模块不能完成或不容易完成的数据库操作。

4 服务器软件性能的测试与分析

本研究主要从服务器程序占用CPU情况、物理内存使用情况和客户端一次连接成功率等3个指标度量来对服务器端软件进行测试和分析。笔者利用一个多线程的客户端来做到并发地发起连接请求,将该服务器软件和原版监控系统中服务器软件(线程池的网络通信模型)安装在服务器上,用于对比测试。服务器的硬件环境为Intel(R)Core(TM)2 Duo CPU,2.93 GHz,2 G内存;服务器软件性能比较如表1所示。

从表1中可以看出,线程池模型的服务器程序和完成端口的服务器程序的3项指标的度量值显示其性能差距明显。就CPU占有情况指标来看,完成端口模型的服务器软件远小于线程池模型服务器软件,而且随着客户端数目增加而增长的相对缓慢。由于对于线程池模型而言不管什么时候服务器系统中都必须有一定数量的线程在运行,几乎所有的线程处于可运行状态,Windows内核在调度处于运行态的线程时引起的线程上下文切换会带来很大的资源消耗^[11],并且随着客户端数量的增加而加剧,有耗尽内存而导致系统崩溃的潜在危险,而完成端口模型无需太多的线程上下文切换,能充分和高效的利用每个工作线程,因而它所使用的物理内存使用情况保持的相对稳定,具有很好的伸缩性。由于两者都具有线程池的优势,本研究预先创建了多个工作线程来处理客户端的请求,因此客户端的一次连接成功率都比较高。目前整个

表1 服务器软件性能测试比较

服务器软件版本	原版服务器软件(线程池模型)					改进版服务器软件(完成端口模型)				
客户端数	50	100	200	500	1 000	50	100	200	500	1 000
CPU 占用情况/(%)	4	7	17	29	42	3	4	7	10	13
物理内存使用情况/MB	135	193	254	310	456	120	126	133	137	141
客户端一次连接成功率/(%)	94	93	91.5	89.5	93	96	96	97.5	98.5	96.8

远程路灯控制系统在测试运行中,服务器软件能够及时地响应客户端的命令要求,控制远程的路灯,运行稳定,表现出良好的性能。

5 结束语

本研究设计开发的的城市路灯远程监控系统服务器端软件涉及 Winsock 网络编程、数据库编程以及如何线程同步和线程间的通信,采用了微软的完成端口模型,能够高效地实现与监控中心的客户端和远程监控终端的网络通信,并通过 ADO 技术实现了对数据库的访问,从而能够对保存在 Access 数据库里的数据实现自动的存储和读出。

本研究在程序设计的过程中重点考虑的是软件运行时的稳定性和后期的可扩展性,且通过调试运行,已经实现了基本的功能。

参考文献(References):

[1] 王 翥,孙嘉宁,范士伟,等. 节能型城市路灯监控系统的设计[J]. 微计算机信息,2009,25(7):10-12.

[2] 王成福,唐晓强. 基于 GPRS 的路灯监控系统的设计与实

现[J]. 电力系统通信,2008,29(8):18-21.

[3] 刘敦东,黄 祚,孙洪飞. 完成端口在路灯监控软件中的应用[J]. 照明工程学报,2010,21(2):66-70.

[4] [美] RICHTER J,[法] NASARRE C. Windows 核心编程[M]. 葛子昂,周 靖,廖 敏,译. 5版. 北京:清华大学出版社,2008.

[5] 吴永明,何 迪. 基于完成端口的服务器底层通信模块设计[J]. 信息技术,2007(3):115-118.

[6] 杜翔雷,跃 明. 基于 IOCP 的服务器端应用程序[J]. 计算机系统应用,2009(2):151-154.

[7] 周 海. 智能路灯节能控制系统研究[D]. 武汉:武汉理工大学自动化学院,2009.

[8] 侯其锋,李晓华,李 莎. Visual C++数据库通用模块开发与系统移植[M]. 北京:清华大学出版社,2007.

[9] 徐 枫,马国之,刘良旭. 基于 ADO 技术的数据库访问研究与实现[J]. 计算机工程与设计,2004,25(1):107-110.

[10] 宋 坤,李伟明,刘锐宁. Visual C++数据库系统开发案例精选[M]. 北京:人民邮电出版社,2006:311-360.

[11] 王艳平,张 越. Windows 网络与通信程序设计[M]. 上海:人民邮电出版社,2006:67-99.

[编辑:李 辉]

(上接第 1497 页)

和图7的对比可见,同一类车辆当运动速度加快时其微多普勒频率范围增大,而时频谱周期缩短。

5 结束语

本研究讨论了轮式和履带式车辆的运动特征,推导了由轮式车辆轮子和履带式车辆履带运动激励的 SAR 回波信号的理论公式,分析了两者的时频特征,并对目标理论回波信号进行了仿真分析;同时通过短时傅里叶变换进行了时频分析,比较了三者时频特性的不同。

理论和仿真结果表明,该车轮转动激励的时频谱具有类似正弦的周期性,而由两类履带运动激励的时频谱存在间断性的类似正弦曲线,可作为区分3类目标的依据。

参考文献(References):

[1] 冀振元,孟宪德. 轮式车与履带车目标的识别[J]. 现代雷达,1992,12(6):17-21.

[2] 丁苏颖,刘宏伟. 基于微多普勒特征的轮式车辆和履带式车辆识别[C]// 第十届全国雷达学术年会论文集. 北京:国防工业出版社,2008:1234-1237.

[3] 黄 健. 履带式车辆多普勒效应与特征提取[D]. 长沙:国防科学技术大学研究生院,2006.

[4] 鄢晗菡,孙 华,刘 川,等. 基于多普勒效应的汽车刹车预警系统研究[J]. 机械,2010,37(11):56-58.

[5] CHEN V C. The micro-doppler effect in radar [M]. Artech House Radar Library, 2011.

[6] 张 安,卢在奇,范红旗,等. 基于散射中心模型的舰船 LFM 雷达回波仿真[J]. 雷达科学与技术,2011,8(5):316-320.

[7] 冯思亮,王志刚,徐继麟. 机载合成孔径雷达回波数据仿真研究[J]. 现代电子技术,2005:28(23):42-48.

[8] 吕静静. 机载合成孔径雷达回波信号模拟研究[D]. 西安:西北工业大学电子信息学院,2006.

[9] 贺思三. 微动目标高分辨雷达信号建模及特征提取[D]. 长沙:国防科学技术大学研究生院,2010.

[10] LI Xiang, DENG Bin, QIN Yu-liang, et al. The influence of target micromotion on SAR and GMTI[J]. IEEE Transactions on Geoscience and Remote Sensing, 2011, 49(7): 2738-2751.

[11] 张 朋. 合成孔径雷达回波信号仿真研究[D]. 西安:西北工业大学电子信息学院,2004.

[12] 张爱兵,计科峰,邹焕新,等. 高分辨 SAR 目标散射中心模型分析[J]. 雷达科学与技术,2009,12(6):416-422.

[13] 陈广锋,张林让,王 纯,等. 复合运动目标微多普勒特征的分析 and 提取[J]. 西安电子科技大学学报:自然科学版,2011,38(3):212-216.

[编辑:李 辉]