

## 高速数字图像记录系统的实现\*

郎非<sup>1,2</sup>, 孙锐<sup>1</sup>

(1.南京邮电学院 信息工程系, 江苏 南京 210003; 2.中国科学院 沈阳自动化研究所,  
辽宁 沈阳 110016)

**摘要:**在弹道测量中, 高速数字图像数据的获取有重要意义。在实现记录功能的基础上, 论述实现记录系统的实时性优化问题。对 Windows 的部分性能做出了调整, 解决了 Windows 的非实时性对记录系统的影响。提出了记录系统的线程池模型, 提高了系统对实时性记录任务的吞吐率。制定记录系统的丢帧检测方案, 测试结果表明: 整个系统可完成 50 MB/s 的高速图像记录任务, 工作稳定正常。

**关键词:** 图像采集; 实时性; 线程池; 异步 I/O

**中图分类号:** TP391 **文献标识码:** A **文章编号:** 1007-2276(2005)02-0227-05

## Development on high-speed digital image acquisition\*

LANG Fei<sup>1,2</sup>, SUN Rui<sup>1</sup>

(1.Department of Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;  
2.Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China)

**Abstract:** High-speed digital video acquisition is important for the image inspection in ballistic trajectory. How to optimize real-time characteristic of recording system is discussed. Contraposed windows operation system's non-real-time characteristic, several methods are proposed to reduce non-real-time characteristic of operation system. Thread pool model of system is proposed to enhance throughput ratio of recording task. Scheme of losing image ratio detection is established. Testing result shows that grabbing velocity breaks through 50 MBps and the whole system works steadily.

**Key words:** Image grab; Real-time; Thread pool; Asynchronous I/O

### 0 引言

近年来, 随着 CCD、CMOS 等成像器件工艺的不断进步完善, 相机不论帧频速度、分辨率还是动态范围和灵敏度都有了较大的提高。其中高分辨率相机广泛应用于科学研究、医疗数字影像和航天航空等领域, 高速面阵相机可用于弹道学研究和爆炸力学研究等<sup>[1,2]</sup>。相机不论采用高帧频还是高分辨率都对提高

测量精度起着重要的作用。但是高速度、高分辨率都会使用于传输图像数据的带宽提升, 用于存储图像的磁盘空间加大, 这对图像记录系统的硬件和软件结构设计提出了考验。

### 1 记录系统的硬件构成

记录系统主板采用 Virtual Micro 公司的 P4DC6+,

收稿日期: 2004-04-09; 修订日期: 2004-06-10

\* 基金项目: 中国科学院创新课题资助项目 (F020409)

作者简介: 郎非 (1977-), 男, 吉林长春人, 硕士, 主要从事图像处理与模式识别的研究工作。

中央处理器采用 SMP 双 CPU 服务器处理器模式。采用 SCSI 接口的磁盘阵列(并行 3 块磁盘),图像采集板 PC-DIG,相机 Dalsa CA\_D6。选用该硬件体系结构有三个显著特点,记录系统硬件构成如图 1 所示。

(1) 以往 SCSI 接口总线都接到南桥 (ICH),而 P4DC6+ 主板把 SCSI 接口总线安排到北桥 (MCH),这样距离 CPU 和内存更近,传输数据将更加迅速。

(2) 内存和北桥采用双通道连接,读写各一个通道,可以并行进行读写工作,速度更快。

(3) 以往外部设备的 I/O 中断要从南桥经过 PCI 总线,到达北桥之后,再送到 CPU。而 P4DC6+ 主板,采用专用的 APIC 总线,从南桥直接到达 CPU。采用这种方式 CPU 对 I/O 中断请求反应更加迅速。

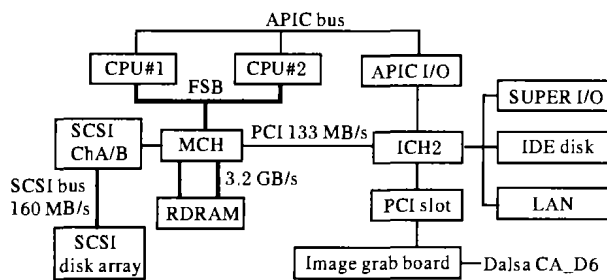


图 1 记录系统硬件构成总图

Fig.1 Architecture of recording hardware system

## 2 记录系统的主要技术指标

该记录系统是为实时记录高速摄像机的图像数据而设计的,用在弹道测量中。图像分辨率为  $512 \times 512$  以上、灰度图像深度 8 位,帧频速度 200 fps 以上,图像接收时间不受存储容量的限制。用 PC 机采集和存储图像数据,采集存储速度要求在 50 MB/s 以上。实验平台所选的主要构件,2 个 CPU,2 GHz;内存容量,1 GB;磁盘,3 块 SCSI 接口硬盘,15000 rpm,单盘容量:18 GB;SCSI 适配器,ADAPTEC ULTRAL 160,最高速度:160 MB/s。

## 3 针对图像记录系统的操作系统实时性能的调整与优化

### 3.1 记录系统对实时性的要求

高速图像数据流所占用的传输带宽非常大,如此

惊人的数据存储速度,对整个系统实时响应能力提出了非常高的要求。以前实现高速数据采集往往通过设计基于单片机或 DSP 的专用硬件从外围设备接收数据,处理后存入存储介质。由于通过硬件完成工作可准确估计延时且不会出现不同软件争用硬件的情况,所以这样的系统实时性极强,数据传输率高,但这种方案也有明显的缺点,即开发周期长,过于复杂,难于调试,通用性差,开发成本也过高。相比之下,采用基于 Windows 操作系统的通用 PC 体系结构,实现同样的功能相对简单,绝大部分功能可通过软件实现,因而易于调试和扩展。不足之处是由于受到 Windows 非实时性的制约,数据传输率较直接用硬件实现低,不能准确估计延时,实时性相对较差。由于本工程指标对系统资源要求十分苛刻,所以能否减小 Windows 非实时性的影响成为系统设计成功的关键<sup>[3,4]</sup>。

### 3.2 Windows 非实时性因素

(1) 强实时性要求对硬件资源的独占访问,而这是 Windows NT 操作系统所不允许的。必须请求排队,逐一占有硬件资源。这样设计会造成在关键时间资源被别的进程占用,从而造成对资源访问的延时。对于高速图像流,如果程序的延时过长,会造成卡上缓冲写满或内存溢出,而这时驱动程序还没来得及提取数据,最终导致丢失图像。

(2) 操作系统中的文件系统也是影响实时性的主要原因。操作系统动态分配磁盘空间,从而导致数据在磁盘上的物理存储空间不连续,磁头要反复寻道才能找到存储位置,磁头寻道会浪费大量时间,寻道时间一般比磁盘读写时间高 2 个数量级,因此会大大降低存储的速度。

(3) Windows 中的页交换技术也会影响实时性。当换页频繁时,会造成磁头反复寻道,频繁的磁盘 I/O 操作会严重影响数据的存储速度。

### 3.3 操作系统实时性能的调整与优化

对于操作系统本身,非实时性是固有的,所以只能做到尽量减少其影响。可以通过调整和优化 Windows 2000 系统内部的一些特殊的机制达到。

#### (1) 异步 I/O (Asynchronous I/O)

采取异步 I/O 设计的目的,是提高 CPU 的利用率

和系统对任务的吞吐量。异步 I/O 完成 (也称 overlapped I/O) 是 Win32 的一项技术, 可以要求操作系统为用户传送数据, 并且在传送完毕时通知用户, 在数据传输期间, 应用进程继续工作。

I/O 完成端口 (I/O Completion Ports) 技术是 Microsoft 提供的一种异步 I/O 技术。I/O 完成端口最显著的特征是将启动 overlapped 请求的线程和提供服务的线程拆开。完成端口提供了就绪工作线程用于服务 I/O 完成请求通告。

### (2) 初始化 NTFS 磁盘

由于磁盘被反复地用于不同的记录任务, 很多存储区域会不断地被重复读写, 长时间经过这样操作, 造成了磁盘物理空间的不连续性, 影响了记录的连续性。在方法改进中, 在每次记录存盘操作之前, 应该先将硬盘进行一次初始化整理 (不同于 format 命令), 去掉磁盘碎片, 保证记录区域的连续性。另外考虑到如果硬盘上有足够的连续区域, NTFS 内部算法也会首先安排文件占用同一磁道的连续位置, 避免了磁头寻道浪费的大量时间。

### (3) 避免 Windows 2000 中的虚拟页交换

解决的办法是在分配内存时锁定它, 使它始终驻留在内存中。一般通过调用 Win32 系统函数 VirtualAlloc() 分配整页内存, 用 VirtualLock() 锁定内存。

## 4 图像实时记录系统的软件实现

### 4.1 基于线程池技术的多线程读写方案

记录过程就是 (从采集卡) 读和 (往硬盘) 写的过程, 也就是一个 I/O 过程。前面提到异步 I/O 相对于同步 I/O, 可以提高 CPU 的利用率和系统对任务的吞吐量, 最终增强系统的实时性。异步 I/O 实质是基于多线程技术。应用程序分别采取读写两个线程, 完成一幅图像的记录任务。由于相机速度 200 fps, 如此多的线程创建和销毁极浪费时间, 需要花费更多的代码和精力去控制线程以及实现线程之间的轮循和状态转换, 多线程引发的问题很容易暴露出来 (如死锁、互斥和饥饿), 可用线程池技术对多线程进行优化。

所谓线程池就是应用进程在启动或运行过程中

创建一定数量的线程, 放入到一个“池子”中, 当有客户请求到达时, 不是创建一个新线程为其服务、而是从“池子”中选择一个空闲的就绪线程为新的客户请求服务, 服务完毕后, 线程进入空闲线程池中。利用线程池后, 创建线程的开销几乎可以忽略不计, 加快了服务器的响应时间, 增强了整个系统的实时性<sup>[5]</sup>。

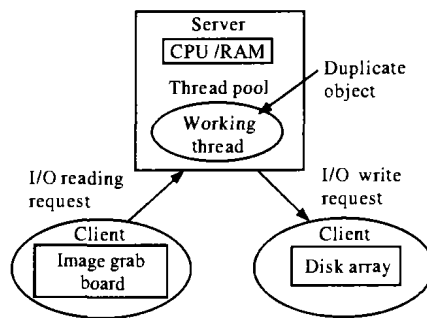


图2 记录系统的线程池

Fig.2 Thread pool of recording system

### 4.2 程序所使用的数据结构与工作线程

(1) 任务队列: 读采集卡数据任务请求进行排队, 记为 QReadTask; 存盘任务请求进行排队, 记为 QWriteTask;

(2) 存储队列: 在主存中, 设置队列 QRAM1 和 QRAM2 存放 RAM1 和 RAM2 的存储单元, RAM1 存放已读出或欲读出的单幅图像数据, RAM2 存放欲存盘的单幅图像数据;

(3) 线程池与工作线程: 创建读线程池和写线程池, 分别创建载入读工作线程和写工作线程;

(4) 完成池与工作线程: 创建读完成池和写完成池, 分别创建载入读完成线程和写完成线程;

(5) 工作线程的就绪队列。四个池中等待线程以 FILO 次序排列 (先进后出), 存放读工作线程和写工作线程的就绪队列, 记为 QReadThread 和 QWriteThread; 存放读完成线程和写完成线程的就绪队列, 记为 QCReadThread 和 QCWriteThread, 它们都由系统自动管理;

(6) IRP 与 ICP 队列: 由 NT 内核管理的队列, 读、写 I/O 请求包 (IRP) 队列, 分别记为 QRIRP 和 QWIRP; 读、写 I/O 完成包队列, 分别记为 QRICP 和

QWICP。由系统管理,列出它们是为了数据流的完整性。

### 4.3 图像数据流程控制算法设计

(1) 必要的准备工作,为 2 个线程池和 2 个完成池分别产生指定数目的工作线程,线程分别被放入就绪队列;为避免文件系统动态分配磁盘,调用 `Create File(……, NO_BUFFERED)`;设置队列 `QReadTask` 初始长度等于相机帧频;

(2) 由主线程为任务请求队列 `QReadTask` 加上个数等于相机帧频数目的读数据任务,每秒钟重复一次这个操作;

(3) 读线程池中等待就绪状态的读工作线程,不断地以 FILO 的次序从 `QReadThread` 就绪队列取出,然后从 `QReadTask` 队列取出读任务,并且从存储队列 `QRAM1` 中申请一块 `RAM1`,同时对 `RAM1` 加锁,紧接着发起一次异步 I/O 请求。系统的 I/O 管理器在为此请求生成 I/O 请求包、作 I/O 前的准备并排队到 `QIRP` 后返回,然后读工作线程重新处于就绪状态。后面的工作驱动程序来完成,读工作线程继续重复以上步骤,直到读线程池中的工作线程全部用完或自己的时间片用完;

(4) 只要上述 `QIRP`(队列)不空,并且当图像采集板的图像同步时钟到来,驱动程序从中逐一取出 I/O 请求包,把数据从图像采集卡缓冲中取出,发送给主机。此间,当读操作完成,读完成池将回收到一个 I/O 完成包,这就是 I/O 完成通告;

(5) 读完成池上就绪的读完成线程持续不断地执行调用 `GetQueuedCompletionStatus()` 函数,用它接收 I/O 完成包。然后此线程负责把接收成功的 `RAM1` 数据取出,经过必要的处理,然后在 `QRAM2` 中申请一块 `RAM2`,这时把 `RAM1` 中经过处理后的数据拷贝到 `RAM2` 中。随后把 `RAM1` 解锁交还给 `QRAM1`。至此一个读数据异步 I/O 循环结束。如果 ICP 中存放的是接收失败的信息,需要把该丢失图像帧做记录(取同步时钟),通知给用户;

(6) 开始完成一幅图像从内存转移到磁盘的写异步 I/O 循环,后面的过程与(3)相似,写线程池中等待

就绪状态的写工作线程,不断地以 FILO 的次序从 `QWriteThread` 就绪队列中取出,然后从 `QWriteTask` 队列取出任务。首先对 `RAM2` 加锁,其后验证 `RAM2` 的容量是否是 512B 的整数倍,如不是则需补位,再在写工作线程中执行 `WriteFile()`,申请 I/O 存盘操作。如果还有新的 `RAM2`,则重复这一过程;

(7) 文件系统会调用磁盘驱动把 `RAM2` 存入硬盘。当写操作完成,过程与(4)一样,读完成池将回收到一个 I/O 完成包;

(8) 读完成池上就绪的读完成线程操作过程同(5),持续地执行调用 `GetQueuedCompletionStatus()` 函数,用它接收 I/O 完成包。然后此线程负责把接收成功的 `RAM2` 数据解锁,交还给 `QRAM2`;如果 I/O 中存放的是接收失败的信息,需要把此次操作 `RAM2` 进行解锁,放入 `QWriteTask`,等待再次被存盘,至此一个存盘异步 I/O 循环结束;

(9) 主线程独立于以上各线程。用于启动、停止整个过程。

### 4.4 算法说明

(1) 对 `QReadTask` 队列中的任务由用户程序直接生成即可。但要注意单位时间内生成任务的速度有一定要求,如果 `QReadTask` 上没有任务请求,采集卡上的数据在规定的时间内没有被取走(即 `IRQ` 队列上有请求),就会发生丢帧。要保证 `QReadTask` 队列不空,可采取队列“一次生成法”。具体来说,首先要求初始化 `QReadTask` 的长度等于每秒钟相机拍摄图像的个数;然后当主线程启动后每一秒钟为 `QReadTask` 加上个数等于相机帧频数目的读数据任务。不难看出,此种算法的目的是让 `QReadTask` 中一直保证有限个数的任务,但这会造成读线程池中的读工作线程都处于运行状态(但在处理 IRP 时这些处于运行态的线程大部分会进入等待状态),程序运行一段时间后,几乎没有处于就绪状态的线程。因为由用户产生的任务,保证 `QReadTask` 队列不空,就是为了线程池中产生足够运行线程对象,为 IRP 队列服务;

(2) 经过“标记的 `RAM1`”特征信息由 `ReadGetQueuedCompletionStatus()` 函数的 `pKey` 进行传递,用于

服务写 ICP 的任务请求的工作线程是读完成线程,其主要作用是当有存盘失败的现象发生时,负责将丢失的图像信息做记录。值得注意的是:pKey 包含在完成池回收到的 ICP 中,pKey 含有此次 I/O 操作的内存块地址和实际传输的字节数;

(3) QWriteTask 任务的数据结构 RAM2 相同,设置 QWriteTask 目的是为了能够起到暂时存放存盘失败的数据帧,而不被读完成线程所发来的数据 RAM1 覆盖掉,使有机会再次被存盘。

(4) 由于 RAM 是共享设备,不同线程要同时操作一块存储区域,就会产生冲突。当读写线程对存储队列中的 RAM1、RAM2 的数据进行传输操作时,要对 RAM1、RAM2 进行加锁,防止其他线程操作对其覆盖,当数据成功传输完毕,要对 RAM1、RAM2 进行解锁,释放资源;

(5) 优先级设置及抢占策略,如表 1 所示;

(6) 线程数目的安排:读线程池中 10 个读工作线程,写线程池中 10 个写工作线程,读完成池中 6 个读完成线程,写完成池中 6 个写完成线程。

表 1 线程及中断的优先级设置  
Tab.1 Priority levels of threads

Thread name	Priority level	IRQL
GrabbingDriver		DIRQL DISPATCH_LEVEL
ReadThread	TIME_CRITICAL	PASSIVE_LEVEL
WriteThread	PRIORITY_HIGHEST	PASSIVE_LEVEL
MainThread	PRIORITY_HIGHEST	PASSIVE_LEVEL
ReadCompletionThread	PRIORITY_NORMAL	PASSIVE_LEVEL
WriteCompletionThread	PRIORITY_NORMAL	PASSIVE_LEVEL

### 5 系统丢帧率检测方案

在系统测试分析前,必须有准确的检测记录系统性能指标的方法,这是下一步进行数据分析的前提。

检测项目有:(1)记录可靠性(丢帧率);(2)记录

图像与系统信息数据(时间、方向角等)的同步。通过第二项的检测数据可以得到第一项的检测结果。具体方法如下(以记录帧频 200 fps 为例):

用 200 Hz 信号同步触发相机,使相机工作在 200 fps 状态下。该同步信号及计数器产生的数据同时接入主机系统,同步信号作为计算机硬件中断,记录系统依靠设计的驱动程序提供中断服务例程,读取计数器的计数值,并与同时到达主机系统的图像对应存储起来。通过判断图像与数据的对应关系以及计数值的连续性,可以计算检测上述两个指标。以上过程如图 3 所示。

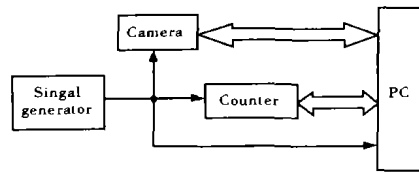


图 3 检测方案框图

Fig.3 Chart of detecting

### 6 结 论

本记录系统是靶场记录弹道数据工作的重要组成部分,经过实际的反复测试,整个系统可以完成 50 MB/s 的高速图像记录任务,且工作稳定正常。

#### 参考文献:

- [1] 黄立胜,刘月花.遥感数据的高速实时记录器[J].遥感学报,2003,2003,7(1):19-25.
- [2] 苏建忠,孙再龙.激光雷达图像采集处理技术研究[J].红外与激光工程,2001,30(5):370-373.
- [3] 张李超,韩明,董扬斌,等.Windows NT 的实时性的研究[J].计算机工程与应用,2002,38(5):41-42.
- [4] 何勇,王青,朱日宏,等. Windows 环境下干涉图像采集和处理[J].红外与激光工程,2003,32(2):203-206.
- [5] 李昊,刘志镜.线程池技术的研究[J].现代电子技术,2004,27(3):77-80.
- [6] John L Hennessy,David A Patterson.Computer Architecture A Quantitative Approach[M].N Y:Morgan Kaufmann Publishers,Inc,1996.
- [7] Chris Cant.Writing Windows WDM Device Drivers [M].N Y:R&D/ Miller Freeman, Inc,1999.