基于消息机制的实时屏幕共享技术

杜俊勇　王国胤

（重庆邮电学院计算机学院　重庆 400065）

摘　要　实时的屏幕共享技术广泛使用在远程控制中，但是目前大部分的屏幕传送是采用静态定时的方式，效果并不理想，特别是在实时性方面。本文介绍了一种结合了 Windows 消息机制的屏幕共享技术，它采用了动态的消息触发机制，传送变化的部分图像，大大地改善了效果。

关键词　屏幕共享，消息，动态捕获，网络传输

A Method for Real Time Screen Sharing Based on Message Mechanism

DU Jun-Yong　WANG Guo-Yin

(Chongqing University of Posts and Telecommunications, Chongqing 400065)

Abstract　Real time screen sharing technology is used widely in remote controls. In most systems, screen information is transferred as static image. There is some delay for the image transmission. In this paper, a method using message mechanism of Windows operation system to share screen information dynamically is developed. Its performance is better than old methods.

Keywords　Screen share, Message, Dynamical capture, Network

1　引言

屏幕共享技术出现在图形化的用户界面(GUI)之后。最早的 GUI 概念由苹果公司提出，并很快得到了业界认可。现在，绝大多数流行的操作系统都提供了一定的图形用户界面。作为计算机信息显示的主要方式，屏幕显示信息本身开始成为一种新的信息源。随着网络应用的发展，人们开始利用网络去实现诸如远程控制、监视等方面的应用[5,7]。这方面的应用需求直接促进了屏幕共享技术的发展。

屏幕共享的目的是在远端客户机上重现主机屏幕的界面显示内容。或者是以一种虚拟终端的概念出现，使得远端的用户感觉与在主机旁边一样。基于屏幕共享技术，目前有很多的产品。比如 Xwindow、Windows 2000 的远程终端服务、NetMeeting 、Symantec 的 PC Anywhere[6]、Remote Administrator 以及国内的红蜘蛛教学系统[2]等。本文首先介绍了几种流行屏幕共享技术，分析了它们各自的优缺点。然后介绍了 Windows 操作系统的消息运作机制和钩子技术[4]，并在此基础上提出了一种结合消息系统的屏幕共享技术。

2　屏幕共享技术介绍

目前常见的屏幕共享实现方式[1]：利用操作系统底层的 GUI 矢量指令、利用拷屏和无失真压缩算法、利用拷屏和特殊有失真压缩算法。

2.1　利用操作系统底层的 GUI 矢量指令

Xwindow 是这类技术的典型代表。Xwindow 是运行在 UNIX 下的图形用户界面系统。在工作的时候，用户登录到主机上，然后启动本地的 Xwindow Server，而主机上的用户进程则创建一个 Xwindow 的 Client。随后，用户的操作指令通过这种 C/S 结构传输给主机，操作的结果生成一个图像的显示，然后 Xwindow 将显示拆分成原子的矢量绘图命令传输给 Xwindow Server，并在用户本地机器上正确解码显示。

Windows 2000 提供的远程终端服务采用了类似的原理。Windows 2000 Professional 使用终端服务客户端软件登录到 Windows 2000 Server 的终端服务。然后，Server 会给每个客户端开辟一个虚拟的桌面(Desktop)，然后将该用户的操作结果显示重定向到他所在的 Desktop，这些显示操作被分解成若干 WindowsGDI 指令，然后传输到 Client 解码显示。而在客户端的感觉是如同真正的操作 Server 一样。

利用操作系统底层的 GUI 的矢量指令实现屏幕共享，能够仅在网络中传输少量的数据就可以得到极好的效果。但是它的缺点是设计到操作系统的底层实现起来比较复杂。

2.2　利用拷屏和无失真压缩算法

最典型的利用拷屏和无失真压缩的软件是 PCAnywhere。PCAnywhere 是一个远程控制软件，其作用相当于一个通用的远程终端。同 Windows 2000 的终端服务一样，PCAnywhere 采用的也是 C/S 的模式。管理员从远程通过 TCP/IP、COM 口、局域网等多种方式连结和登录到服务器。然后进行远程控制。与 Windows 2000 的终端服务不一样，当管理员在控制主机的时候，主机屏幕会随着变化；同时 PCAnyWhere 理论上只能允许同时接入一个连结。这些不同是因为 PCAnywhere 采用了不同的工作原理：屏幕显示的内容不是通过拆解成具体的绘图命令来实现的，而是通过拷屏和数据压缩处理后，作为一个图片传输到客户端的。PCAnywhere 采用的是无失真的简单压缩方法。这一类方法的代表是 RLE(运行长度编码)和 LZW 等。

PCAnyWhere 运行在 MS Windows 平台上，在这个平台上多数应用具有简单的 GUI 界面。其纹理和色彩具有单一、规则等特点，所以非常适合用上述无失真的压缩办法处理。其压缩率一般可以达到 30～50 倍。

上述算法的另一个优点是具有很快的运行速度，这在远

杜俊勇　硕士研究生，研究方向为计算机网络；王国胤　教授，博士生导师，研究方向为计算机网络、人工智能等。

程控制应用里面显得非常重要。特别是RLE算法，尽管压缩效率不是最高，但是速度却非常快，在现代主流配置的PC上可以在耗用很少CPU时间情况下满足一般的屏幕共享要求。无失真的算法对于文字的恢复质量很好，由于计算机屏幕显示内容多为文字和简单图形，所以这类算法在客户端可以获得很清晰的图像恢复效果，但缺点是对于图像部分则显得力不从心。

2.3 利用拷屏和特殊有失真压缩算法

比较典型的算法是采用JPEG标准进行压缩，但由于JPEG标准是应用于自然图像压缩，所以并不适合GUI这样的简单图像。在处理界面上没有图片的时候，JPEG算法往往不及简单压缩算法效率高。

但如果界面上存在较多的图片部分，JPEG算法则表现出比无失真高得多的压缩率。所以一般情况在画面质量要求不是很严格的条件下，利用有失真的图形压缩算法也是可行的。

3 基于消息机制的屏幕共享技术

基于消息机制的屏幕共享技术实际上也是上面所述利用拷屏技术的一种，但是与采用以整屏幕为单位的传送画面的方法不同，它采用分块的方法传送变化的画面，而且在画面的传输激发机制上也完全不同。

首先先介绍一下Windows操作系统的消息运作机制以及能截获各种消息的钩子(hook)功能。

3.1 Windows系统的消息机制

Windows系统是一个消息驱动的操作系统。一个消息由一个消息名称(UINT)和两个参数(WPARAM,LPARAM)组成。当用户进行了输入或是窗口的状态发生改变时系统都会发送消息到某一个窗口。例如当菜单选中之后会有WM_COMMAND消息发送，WPARAM的高字中(HIWORD(wParam))是命令的ID号，对菜单来讲就是菜单ID。当然用户也可以定义自己的消息名称，也可以利用自定义消息来发送通知和传送数据。消息必须要由一个窗口接收。在窗口的过程(WNDPROC)中可以对消息进行分析，对自己感兴趣的消息进行处理。例如你希望对菜单选择进行处理那么你可以定义对WM_COMMAND进行处理的代码，如果希望在窗口中进行图形输出就必须对WM_PAINT进行处理。

可以为窗口编写默认的窗口过程，这个窗口过程将负责处理那些用户不处理的消息。正因为有了这个默认窗口过程程序员才可以利用Windows的窗口进行开发而不必过多关注窗口各种消息的处理。例如窗口在被拖动时会有很多消息发送，但程序员都可以不予理睬让系统自己去处理。

系统通过窗口句柄来在整个系统中唯一标识一个窗口，发送一个消息时必须指定一个窗口句柄表明该消息的接收窗口。而每个窗口都会有自己的窗口过程，所以用户的输入就会被正确处理。例如有两个窗口共用一个窗口过程代码，你在窗口A上按下鼠标时消息就会通过窗口A的句柄被发送到窗口A而不是窗口B。

Windows操作系统将会维护一个或多个消息队列，所有产生的消息都会被放入或是插入队列中。系统会在队列中取出每一条消息，根据消息的接收句柄而将该消息发送给拥有该窗口的程序的消息循环。每一个运行的程序都有自己的消息循环，在循环中得到属于自己的消息并根据接收窗口的句柄调用相应的窗口过程。而在没有消息时消息循环就将控制权交给系统所以Windows可以同时进行多个任务。

当该程序没有消息通知时getMessage就不会返回，也就不会占用系统的CPU时间。在16位的系统中只有一个消息队列，所以系统必须等待当前任务处理消息后才可以发送下一消息到相应程序，如果一个程序陷入死循环或是耗时操作时系统就会得不到控制权。这种多任务系统也就称为协同式的多任务系统。Windows3.X就是这种系统。而32位的系统中每一运行的程序都会有一个消息队列，所以系统可以在多个消息队列中转换而不必等待当前程序完成消息处理就可以得到控制权。这种多任务系统就称为抢先式的多任务系统。Windows95/NT/2000就是这种系统。

3.2 系统消息的捕获

钩子的本质是一段用以处理系统消息的程序，通过系统调用，将其挂入系统。钩子的种类有很多，每种钩子可以截获并处理相应的消息，每当特定的消息发出，在到达目的窗口之前，钩子程序先行截获该消息、得到对此消息的控制权。此时在钩子函数中就可以对截获的消息进行加工处理，甚至可以强制结束消息的传递。

系统钩子具有相当强大的功能，通过这种技术可以对几乎所有的Windows系统消息进行拦截、监视、处理。这种技术广泛应用于各种自动监控系统中。

HOOK在系统中表现为一条链子，多个HOOK串在这条链子上。HOOK有不同的种类，每种HOOK串分开在不同的链子上。应用程序调用SetWindowsHookEx向系统申请创建一个HOOK。应用程序必须定义一个类似LRESULT CALLBACK HookProc(int nCode, WPARAM wParam, LPARAM lParam)的HOOK回调函数，当前应用程序的HOOK被响应时供系统调用。其中nCode为HOOK的编号(HOOK的种类)，HookProc使用它的值来决定进行什么操作，后面的两个参数的值由nCode决定表示什么。HookProc可以决定是否把当前的消息传到链子上的其他HOOK(使用CallNextHookEx函数)，但是有的消息是由系统发送的，这样它就没有权利决定是否发送了。HOOK分为全局和当前进程两种，全局的HookProc由于它要监听系统所有进程的消息，所以必须放在DLL文件里，进程级的HOOK可以和应用程序放在同一个模块里。(模块：经过编译后的单独的.exe或.dll)

3.3 结合消息机制的动态屏幕共享技术

屏幕实时传送主要涉及到两个方面的问题，屏幕实时的抓取和屏幕实时的传送。目前一般屏幕传送的方式是使用主动触发的方式，在服务器端定时地进行屏幕的截取和存贮，然后再传送到客户端显示。由于是使用定时的抓取，必然存在不连续的情况，遇到网络状况不好的情况，会有大量的数据被丢失。

但是，如果采用消息触发则是被动触发的方式。屏幕发生变化部分的区域大则传送的数据量大，如果发生变化的区域小则传送的数据量小，如果屏幕没有发生变化则不需要传送数据。

Windows操作系统对屏幕重绘操作的基本原理是使用某些消息来通知操作系统屏幕上哪些部分发生了变化，然后在发生变化的区域中进行重绘操作。基于这个原理我们可以截获这些消息并且取得这个区域，这样就可以仅仅传输这个变化的部分的图像即可。由于Windows操作系统大部分时间都是进行的区域的重绘操作而不是全屏的刷新，于是我们就可以大大减少传输的数据量从而极大地提高了传送的实时性。

在服务端利用全局钩子函数截获屏幕变化的矩形。其中需要利用两种类型的钩子WH_GETMESSAGE和WH_

CALLWNDPROC 。WH_GETMESSAGE 类型钩子定义回调函数 LRESULT CALLBACK GetMsgProc (int code , WPARAM wParam , LPARAM lParammessage) ,在返回被检索的消息到调用者前,系统传送消息给钩子回调过程。WH_CALLWNDPROC 类型钩子定义回调函数 LRESULT CALLBACK CallWndProc(int nCode, WPARAM wParam , LPARAM lParam) 在执行窗口过程函数之前,系统调用这个函数。

能够得到变化矩形的消息包括 WM_PAINT 、WM_MENUSELECT、WM_CAPTURECHANGED 、WM_NCPAINT 等。同时还必须包括击键和鼠标移动的坐标,所以还需要鼠标钩子 WH_MOUSE 截获 WM_MOUSEWHELL、WM_KEYDOWN 等消息传送出鼠标的坐标位置和鼠标的形状类型值。

下面一段伪代码演示钩子如何在回调过程中截获屏幕发生的变化

```
LRESULT CALLBACK GetMsgProc ( int nCode, WPARAM wParam, LPARAM lParam)
{
switch(uMessageType)
{//使用 SWITCH 语句将各种消息分开
    case( WM_PAINT ‖ WM_NCPAINT): // 截获 PAINT 和 NCPAINT 消息
        GetUpdateRgn(…); //得到变化的区域
        if(SIMPLEREGION ‖ COMPLEXREGION) // 如果是单一区域或者复杂区域
        {
          GetRegionData(…); //得到这个区域内的所有矩形
          SendMessage(…); //将矩形大小数据发送出去
        }break;
    case (WM_KEYDOWN): //当击键时截获发生击键事件的整个窗口
        SendMessage(…); //将矩形大小数据发送出去
        break;
    case(…) //其他一些消息的截获
        ……
        Break;
    }
 return CallNextHookEx(…); //调用下一个钩子
}
```

具体的处理流程如图 1 所示。

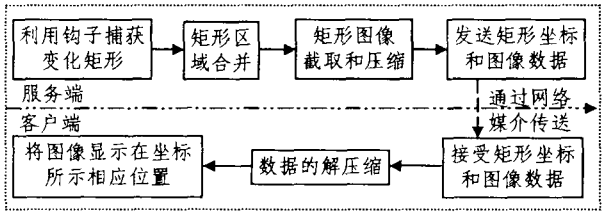

图 1 屏幕图像处理流程

图 1 中各个模块的功能如下:

矩形的区域合并。由于消息的传递是非常快的,有可能在上一个消息还没有处理的时候,又出现了下一个消息,这个时候矩形的区域合并就非常重要,将上一个矩形与下一个进行或操作取其并集,这样可以提高效率。

矩形的截取。利用拷屏技术取得该矩形部分的图像,然后利用失真的压缩算法或者利用无失真的压缩算法将图像数据压缩。

延时的处理。在进行数据发送的时候必然会有一定的延时,但是如果利用这段时间来进行矩形合并后的数据压缩工作,整个过程就可以呈并发处理,大大地加快了流程。

数据的传送。数据的传送使用 UDP 或者套接字。当使用 UDP 套接字的时候可以使用广播[8]的方式,在整个局域网的广播域内进行广播。

鼠标的处理流程如图 2 所示。

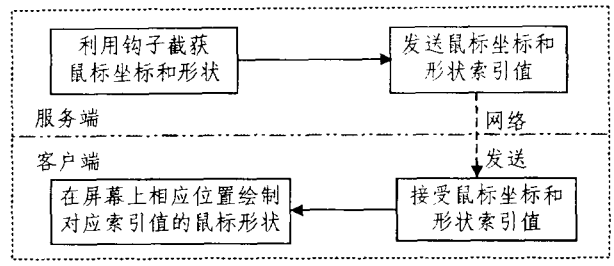

图 2 鼠标处理流程

图 2 中各个模块功能如下:

鼠标的截获。鼠标的信息包括鼠标的位置和鼠标的形状,利用鼠标钩子 WH_MOUSE 监视鼠标的移动同时在发生移动的时候捕获鼠标的位置坐标和鼠标的形状。鼠标的形状采用 Windows 操作系统对应鼠标形状的索引值来表示。

鼠标的传递。鼠标需要传递信息量比较少,因此采用直接传送的方法。

鼠标的绘制。在客户端得到鼠标坐标和鼠标形状的索引值后,利用索引表查找鼠标的形状,在屏幕对应的坐标位置上绘制鼠标。

最后在通过钩子函数传递参数时有一点要十分注意。因为是使用的全局钩子函数,所以传递参数时需要使用 COPYDATASTRUCT 结构或者使用内存映射文件[4]。

4 性能分析

目前按照前面所述的方法我们研制了测试软件 ScServer/Client,实现了屏幕共享的功能。与其他的功能软件(NetMeeting、红蜘蛛、PcAnyWhere、radmin)在同等的网络环境(windows2000 操作系统、100M 网络带宽)下做了屏幕共享性能对比测试,如下表:

表 1

性能 \ 软件名称	Net Meeting	PcAny Where	红蜘蛛	radmin	ScServer /Client
鼠标灵敏度	中	中	优	优	优
图像变化连续性	差	中	优	优	优
图像传送的实时性	中	差	中	中	优
对网络负载的影响	优	差	中	优	优
占用系统资源	差	差	中	中	中

从上表可以看出,在鼠标灵敏度方面明显优于 NetMeeting 和 PcAnyWhere,与红蜘蛛网络教室、radmin 达到同等效果,连贯性和实时性得到很好的体现。在图像变化的连续性方面明显优于 NetMeeting 和 PcAnyWhere。在显示连续的变化图像时优于红蜘蛛网络教室和 radmin。在图像传送的实时性方面明显优于 PcAnyWhere、NetMeeting 和 radmin。在对网络负载的影响方面明显低于 PcAnyWhere、红蜘蛛网络教室和 radmin。在占用系统资源方面表现一般,但明显优于 PcAnyWhere 和 NetMeeting。

此外,在不使用硬件加速的环境下播放 Mpeg、avi 等制式的多媒体课件时同样可以达到良好的连续性和实时性,而这一点是 NetMeeting、PcAnyWhere 和 radmin 等软件所没有达到的。

(下转第 236 页)

2 是通过服务器端拦截器实现的，具体的实现类似于上述客户端请求拦截器。

完成 Interceptor 定义后的任务就是服务器端和客户端 ORB 初始器的构造。ORB 初始器的构造，可以通过实现接口 ORBInitializer 来完成。最后将两个 ORB 初始器分别注册到客户端 ORB 和服务器端 ORB 中。上述设计分别在两种 CORBA 平台 Jacorb1.40 和 Orbix2000 上进行了实现。

5 系统实现与测试

将 JSP/JAVA Bean 技术与 CORBA 技术相结合，我们建立了基于 Web 的支持协同设计的数据库安全共享平台，如图 4 所示。

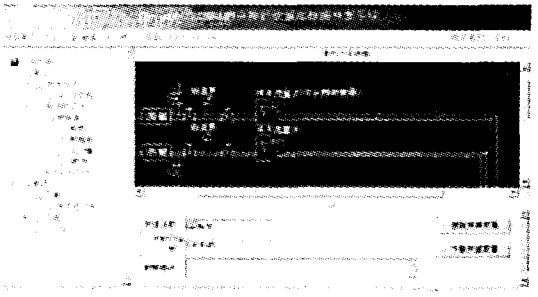

图 4 系统实现结果示例

系统采用了基于 Web 的客户端-应用服务器-数据库服务器的三层体系结构。在应用服务器中利用 JavaBean 作为客户端代理与各个实现共享数据操作的数据库代理交互。协同系统中的用户既可以通过基于 Web 的界面提交共享数据、获取并显示自己有权进行访问的共享数据，还可以通过应用程序编程接口，在应用程序中访问和操作共享数据。

另外，对于数据库安全网关进行了性能测试与分析。测试环境如下：

• 硬件 应用服务器：CPU 1.5Ghz，内存 512M；数据库服务器：CPU 800Mhz，内存 256M。

• 软件 操作系统：Windows 2000 Professional；应用服务器：Tomcat4.1.24；数据库服务器：SQLserver2000；CORBA：JacORB1.4。

选取的测试点是应用服务器中用 JavaBean 封装的 CORBA 客户端获取共享数据所需要的时间。测试中访问的共享数据大小为 100k。分别对三种情况进行测试：(1)共享数据访问中不使用任何安全控制，即不启用数据库访问网关；(2)采用固定的 MAC 策略进行访问控制；(3)采用动态支持多策略的柔性化访问控制；(4)在柔性化访问控制基础上增加了数据加密。表 1 中记录了 6 次测试结果。

表 1 数据库访问网关性能测试数据 （单位：ms）

	1	2	3	4	5	6
无安全控制	515	500	547	515	546	531
MAC 策略	625	656	625	630	657	641
动态访问控制	718	672	719	657	672	703
动态访问控制、加密	859	828	859	828	844	829

测试结果表明，在数据库共享平台中增加数据库访问网关，将增加系统 40%～60%的响应延时，虽然一定程度降低了系统的效率，但也是可接受的。目前正对数据库访问网关的实现进行优化，以提高系统的响应效率。

小结 本文针对航空型号产品异地协同设计中数据库共享的特殊需求，研究并实现了基于 CORBA 的数据库安全共享平台。数据库访问网关的设计与实现，符合协同设计环境中的数据库管理模式，实现了数据共享所需的各种安全技术的有机集成，并且通过利用 CORBA 的可移植拦截器机制，实现了数据访问网关与 CORBA 软件的有机融合，保证了数据库访问网关的有效性。本文在数据库访问网关有效性的研究中，提出了一种增强型安全产品的研究方法。该方法的思想是把具有自主知识产权的安全产品融合到国外先进的安全基础软件中，使增强型安全软件能够快速、直接地应用到国民经济建设的实际应用中，显著提高这些系统的安全保护能力。

参 考 文 献

1 Object Management Group. The common Object Request Broker: Architecture and Specification, Version 2.6, Dec. 2001

2 ISO/IEC 10181-3 Security frameworks for open systems—Access control framework, 1996

3 郎波，吴琦，李伟琴. 分布式对象柔性化访问控制方法研究. 北京航空航天大学学报，30(5)

4 Marchetti C, Verde L, Baldoni R. CORBA Request Portable Interceptors: A Performance Analysis. Conf. On Very Large Data Bases, Santiego, Chile, 1994. 24～35

5 Othman O, O'Ryan C, Schmidt D C. The Design of an Adaptive CORBA Load Balancing Service. IEEE Distributed Systems Online, Apr. 2001.2

6 Brose G, Muller S. JacORB 1.4 Programming Guide Version1.1.9. Mar. 2002, 20

（上接第 225 页）

综合上面所述，可以发现结合消息机制的屏幕共享方法所达到的效果在各个方面都优于其他功能软件，是一种很好的实现屏幕共享的方法。

结束语 屏幕共享技术是一个非常有潜力的应用技术，各种基于屏幕共享技术的软件产品必将在远程教学、监控和多媒体应用中发挥越来越大的作用，从而具有广阔的应用前景和市场前景。该论文旨在解决基于 Windows 操作系统的屏幕同步共享问题。通过对该方案的实施可以达到非常好的同步效果。实现真正的屏幕同步共享。

参 考 文 献

1 陈琦，李凡，朱光喜. 屏幕共享技术及其在多媒体通信中的应用. 华中科技大学电信系图像教研室

2 红蜘蛛软件制作室. Red Spider. http://www.forclass.com

3 徐挺，吉逸，金胜昔，等. 计算机网络技术在远程教学系统中的应用研究. 见：[第十届中国计算机学会网络与数据通信学术会议论文集]. 南京，1998. 335～337

4 王建华，张焕生，侯丽坤. Windows 核心编程. 机械工业出版社

5 张友生. 远程控制编程技术. 电子工业出版社

6 SYMANTEC 公司. PcAnyWhere. http://www.symantec.com/

7 Zahariadis Th, Voliotis S. Networking multimedia classromm initiative. Video/Image Processing and Multimedia Communications 4th EURASIP-IEEE Region 8 International Symposium on VIPromCom 16-19 June 2002. 35～38

8 Pekowsky S, Andorfer A. Multimedia data broadcasting strategies. Communications Magazine. IEEE April 2001, 39(4): 138～145