文章编号:1001 - 9081(2005)08 - 1948 - 03

一种基于主备机快速切换的双机容错系统

吴 娟[1],马永强[2],刘 影[3]
(西南交通大学 计算机与通信工程学院,四川 成都 610031)
(jerry81_wu@yahoo.com.cn)

摘 要:针对现有容错系统中主备机切换过程时延较大的问题,设计了一种主备机快速切换的容错系统。在该系统中,主备机使用相同 IP 地址和 MAC 地址,它们同时接收网络数据,但备机不发送任何网络数据,且该过程对上层应用透明。

关键词:双机容错;可靠性;主机/备机;Linux;心跳协议
中图分类号: TP302.8 **文献标识码**:A

Duplicated fault tolerance system based on active/standby fast-switching

WU Juan[1], MA Yong-qiang[2], LIU Ying[3]
(*College of Computer Science and Communication Engineering, Southwest Jiaotong University, Chengdu Sichuan* 610031, *China*)

Abstract: To shorten the time-delay of active/standby switching in the fault tolerance system, an active/standby fast-switching system was designed. In this system, the active and the standby used the same IP and MAC address in order to receive the network data simultaneously, however, the standby was prohibited to send any network data, both processes were transparent to upper layers.

Key words: duplicated fault tolerance system; reliability; active/standby; Linux; heart-beat protocol

0 引言

双机容错系统的目的在于数据永不丢失和系统永不停机,这就需要不断地提高系统可靠性。在现有的应用于 TCP/IP协议网络的无共享磁盘阵列双机容错系统中,当主机发生故障时,备机要在修改 IP 地址,广播 ARP 报文,路由器修改 ARP 缓存后,才能切换成主机,该切换过程时间过长并且容易丢失数据。为了实现系统的高可靠性,在这里介绍一种新的应用于双机容错系统中的主备机快速切换技术。在该系统的设计中使用了 Packet Driver 编程接口,并对 Linux 内核的网络部分进行了修改,为实现主备机的相互监测,设计了心跳协议,并采用了双心跳线的心跳线冗余方式。

1 现有无共享磁盘阵列双机容错系统分析

现有的图 1 所示结构的无共享磁盘阵列双机热备容错系统,主机和备机同时运行程序,主机与备机之间通过串口、以太网、SCSI 等通道相互监测对方的运行状态,主机还要通过串口、以太网、SCSI 等通道把网络数据转发给备机。当主机出错时,从主机出错到备机侦测到主机出错之间有一段时延,这段时间主机收到的网络数据不一定全部转发给备机,就可能丢失数据。并且,当主机出错时,要在备机修改 IP 地址、发送 ARP 广播、路由器更改 ARP 表之后,备机才能切换为主机继续运行应用程序,这就又造成一个时延。

由于网络数据要通过主机转发给备机,这个过程有可能丢失数据,容错系统应当尽可能避免丢失数据;并且,由于备机切换到主机的过程需要一段时间,容错系统应当尽量减少这段时延。针对现有双机容错系统的不足,本文设计了可靠性更高、主备机切换时延更小的双机容错系统。

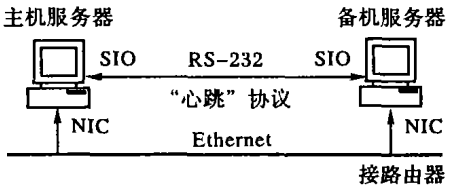

图 1 双机双工容错系统(SIO:串口,NIC:网卡)

2 系统设计

2.1 总体设计

主机和备机使用相同 MAC 地址和 IP 地址,都接到同一个总线型局域网上;只有主机能够接收和发送数据,备机只能接收数据而不能发送数据。这样,主机和备机可以同时接收到网络数据,可以防止网络数据通过主机转发方式中可能的数据丢失,主备机以双机双工的工作模式处理网络数据。当主机出错时,备机直接把自己标识为主机,即可继续正常工作,而不会等待路由器更改 ARP 表的那段时延。这样的双机容错系统就可以达到可靠性更高、主备机切换时延更小的目的。

2.2 处理重复 IP 地址的措施

针对本系统的专用性——备机需要接收到主机所接收的所有网络数据,因此只要主备机使用相同 IP 地址就可以达到目的。由于 TCP/IP 协议是不支持两台计算机使用相同 IP 地址连接到网络的,如果主机与备机使用相同的 IP 地址接入网络,处于同一个冲突域和广播域,那么在运行中,主备机必然都会检测到网络中还有一台使用了与本地主机相同的 IP 地址的网络设备,就会报告地址冲突。这里就需要介绍一下操作系统检测重复地址冲突的工作原理。

在 Windows 中,对重复 IP 地址检测采用主动工作方式,

收稿日期:2005 - 03 - 02;**修订日期**:2005 - 05 - 12

作者简介:吴娟(1981 -),女,四川安岳人,硕士研究生,主要研究方向:双机容错、网络与信息系统; 马永强(1963 -),男,福州人,教授,博士,主要研究方向:车载检测技术、网络与信息系统; 刘影(1980 -),女,山东烟台人,硕士研究生,主要研究方向:密码算法、信息安全.

其实现过程如下:当首次初始化栈或添加新的 IP 地址时,就会免费广播本地主机 IP 地址的 ARP 请求,在该 ARP 请求分组中,发送方硬件地址为本地主机硬件地址,发送方 IP 地址为本地主机 IP 地址,目标硬件地址为全 0,目标 IP 地址为本地主机 IP 地址。发送 ARP 报文的次数是由注册表参数 ArpRetryCount 控制的,默认值为 3。如果另一个主机回复了其中一个 ARP 广播,则表示此 IP 地址已被其他设备使用。发生这种情况时,基于 Windows 的计算机仍然启动;但是使用了相同 IP 地址的接口会被禁用,并生成系统日志项和显示错误消息。如果拥有该地址的那台主机也是基于 Windows 的计算机,则生成系统日志项,并在该计算机上显示错误消息。为了修复可能对其他计算机上 ARP 缓存造成的破坏,原发送 ARP 请求的计算机要重新广播另外一个 ARP 报文,并将此 ARP 请求的源硬件地址字段设置为发送 ARP 应答计算机的硬件地址。

在 Linux 内核版本中,检测重复 IP 地址是通过 DHCP 完成的,在 RFC 2131 中有相关说明。当首次初始化栈或添加新的 IP 地址时,就会免费广播本地主机 IP 地址的 ARP 请求,在该 ARP 请求中,发送方硬件地址为本地主机硬件地址,发送方 IP 地址为全 0。这样的 ARP 请求广播,不会对其他计算机的 ARP 缓存造成破坏。如果本网段中有另一台计算机已经使用了相同 IP 地址,就调用以下函数发送 ARP 应答:arp_send(ARPOP_REPLY,ETH_P_ARP,tip,dev,tip,sha,dev - > dev_addr,dev - > dev_addr)。该应答中,发送方硬件地址为本接口硬件地址,发送方 IP 地址为本地主机 IP 地址,目标硬件地址为发送 ARP 请求的设备的硬件地址,目标 IP 地址为本地主机 IP 地址。发送 ARP 请求的计算机接收到应答后,发现 IP 地址重复,就向 DHCP 服务器发送 DHCPDECLINE 消息,并重启配置程序以请求新的 IP 地址。这样就可以避免重复 IP 地址冲突。

通过分析以上两种处理重复 IP 地址冲突的工作原理可以发现,当备机不发送 ARP 请求和应答时,主机、路由器和其他计算机都是不能察觉到备机的存在的,也就不会发生重复 IP 地址冲突。但备机是能够接收到主机发送的 ARP 请求广播报文的。如果备机的 MAC 地址与主机的 MAC 地址不相同,那么备机会报告重复 IP 地址冲突。因此,设置备机的 MAC 地址与主机相同,它就不会造成重复 IP 地址冲突,其原因如下:位于同一网段中的所有计算机都能够收到广播报文,当然发送 ARP 广播的主机也不例外,但主机并不会报告重复 IP 地址冲突,由于主机和备机的 MAC、IP 地址相同,所以备机会认为接收到的主机 ARP 广播是备机本身发送的 ARP 广播,故 Linux 操作系统并不会报告重复 IP 地址冲突。

计算机能否接收到本机发送的广播报文,这一点作者通过使用 Winpcap 开发包编程进行了测试。将网卡工作方式设置为:NDIS_PACKET_TYPE_DIRECTED + NDIS_PACKET_TYPE_BROADCAST + NDIS_PACKET_TYPE_MULTICAST,这是网卡的一般工作方式。测试结果表明:在该工作方式下,计算机能够接收到本机发送的广播报文。由此可知,备机会认为接收到的主机 ARP 广播是备机本身发送的 ARP 广播,故 Linux 操作系统不会报告重复 IP 地址冲突。

2.3 Linux 内核的网络部分的修改

在双机双工工作模式下,要禁止备机发送网络数据,并且这个过程应该对上层应用透明,就需要对 Linux 内核的网络部分进行修改。在传输层、IP 层、网络接口层都可以通过修改源代码,达到禁止备机发送数据的目的,但考虑到需要禁止 ARP 广播。ARP 报文虽然需要进行 IP 封装,但由于其特殊性,需要分开处理 ARP 报文和普通 IP 报文。因此,必须在网络接口层修改 Linux 内核源代码。在网络接口层中,首先判断本机是否是备机,如果是备机,就直接 free 发送缓冲区并正确返回,这样就可以禁止备机发送数据,并对上层应用透明。

2.4 "心跳协议"的设计

主机与备机需要相互监测对方的工作状态,通过RS-232串口通道,按照"心跳协议"相互监测各自的运行情况。当出错时,待修复完成以后,就可以按照日志恢复数据。

"心跳协议"规定,当备机收到主机的出错信息,或者是连续 5 次接收不到心跳数据时,备机切换为主机接管主机的工作;原主机修复后按照"主/备机自动识别协议"设置为备机,按照备份日志恢复数据。当主机收到备机的出错信息,或者是连续 5 次接收不到心跳数据时,仍然继续正常工作,备机修复后按照"主/备机自动识别协议"设置为备机,按照备份日志恢复数据。

3 系统实现

3.1 标识主/备机

本系统使用的是将 Linux 源代码按照系统的需要加以修改后编译的 Linux 操作系统,内核版本为 2.4.22。在源代码中,加入了变量 ActiveStandby,ActiveStandby 为 1 表示主机,ActiveStandby 为 0 表示备机。编写函数 SetActiveStandby(int flag)作为对外的接口。通过该接口设置 ActiveStandby 的值,Linux 就可以根据 ActiveStandby 的值确定该计算机究竟是主机还是备机,执行相应的功能。

3.2 修改 MAC 地址

首先了解网卡的寄存器结构。每个网络适配器上都配置有数据缓冲区 RAM,长度从 16KB 到 64KB 不等。硬件地址寄存器固化在网卡上的 PROM 中,PROM 的长度为 32 字节,其中 6 个字节的硬件地址占据了 PROM 的最低 6 个地址。PROM 和缓存 RAM 在网卡上统一编址,PROM 处于低地址部分,缓存 RAM 处于高地址部分。PROM 属于 EEPROM,可以通过编程进行修改,也就能修改了 MAC 地址。

修改 MAC 地址可以使用 Packet Driver 编程接口来实现。由于直接网卡编程的通用性差,针对某一种网络适配器编写的程序只适用于相同或相似的网络适配器。为了屏蔽网络适配器的内部实现细节,使用户与网卡的通信更为方便,几乎所有的网卡生产厂家都随网卡提供相应的网卡驱动程序,其中包含了 Packet Driver 编程接口,由它来屏蔽网卡的具体工作细节,在上层应用软件和最底层的网卡驱动程序之间提供一个接口,这个接口是对各种不同网卡的抽象。Packet Driver 编程接口,其中提供功能号为 19H 的功能调用,就可以修改网络设备的 MAC 地址。程序片断为:

```
...
asm push es
asm push ds
asm mov ah, 0x19                    //功能号,用途是修改硬件地址
asm mov bx, PktDrv_Handle                                  //句柄
asm mov cx, 0x06                                  //物理地址长度
//设置网卡物理地址存放地址 ES: DI, macAddr 中存放要修改成
//的硬件地址
asm mov di, offset macAddr
asm push ax
```

```
asm mov ax, seg macAddr
asm mov es, ax
asm pop ax
asm mov dh, 0                                    //错误代码(清 0)
asm int 62h
//调中断, 初始化网卡前必须检查 Packet Driver 是否//安装在
0x62 上 asm pop ds
asm pop es
asm jc end                                       //出错
end:
asm pop di
asm pop si
...                                              //恢复现场
return result;
...
```

因篇幅有限,不详述修改 MAC 的细节。

3.3 修改 Linux 内核的网络部分

Linux 内核的网络分层结构如图 2 所示。按照从底层向上的顺序做以下介绍。

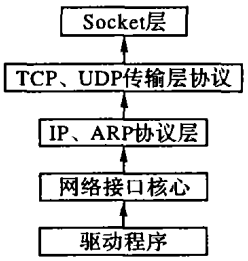

图2 Linux 内核的网络分层

其中,驱动程序的源代码在 linux/drivers/net/目录中。例如:3c501.c 文件就是 3Com3c501 以太网卡的驱动程序。其中,el_start_xmit( ) 函数就是实际向 3Com3c501 以太网卡发送数据的函数,具体的发送工作就是对网卡寄存器的读写。el_receive( ) 函数是用来接收数据的函数,它首先要判断收到数据包是否正确,如果是正确的,就为包分配一个缓冲结构(dev_alloc_skb( ))。这样,驱动程序的接收工作就完成了,通过调用位于 net/core/dev.c 的上层的函数 netif_rx( ),把包交给上层。

网络接口核心,可以认为是对不同种类网卡的抽象。它的上层是具体网络协议,下层是驱动程序,功能是接收和发送数据。网络接口核心层通过位于 net/core/dev.c 的 dev_queue_xmit( )函数,向上层提供统一的发送接口,无论 IP 还是 ARP 协议,都是通过这个函数把要发送的数据传递给该层。dev_queue_xmit( )最后落实到 dev->hard_start_xmit( )调用实际的驱动程序来完成发送任务。实际上,dev->hard_start_xmit( )就是调用了 el_start_xmit( )。netif_rx( )接收上层发送来的数据,并把数据包向上层发送。通过 static struct packet_ptype_base[16]这个数组传递数据给上层,这个数据包含需要接收数据包的协议,以及它们的接收函数的入口。可以看到,IP 协议通过 ip_rcv( )函数接收数据,ARP 协议通过 arp_rcv( )函数接收数据。

IP、ARP 协议层,IP、ARP 协议是需要直接和网络设备接口打交道的协议。TCP、UDP 传输层协议是直接利用 IP 协议,从 IP 协议接收数据,而且要向上层 Socket 层提供直接的调用接口。

要禁止备机发送数据,只要修改网络接口核心层的 dev_queue_xmit( )函数即可。在该函数中增加以下代码。

```
if (ActiveStandby = =0)
  { kfree_skb(skb);                              //释放缓冲区
    return -ENETDOWN;
                  //表示发送成功,即可做到对上层应用透明
  }
```

3.4 心跳协议

使用 RS-232 串口通道传递心跳数据,心跳协议包括以下部分:

(1) 以 100ms 为单位发送本机状态,包括正常状态和出错状态;

(2) 当备机收到主机出错信息,或者是连续 5 次接收不到心跳数据时,备机切换为主机接管主机的工作;原主机修复后按照"主/备机自动识别协议"设置为备机,按照备份日志恢复数据。

(3) 当主机收到备机出错信息,或者是连续 5 次接收不到心跳数据时,仍然继续正常工作,备机修复后按照"主/备机自动识别协议"设置为备机,按照备份日志恢复数据。

(4) 无论主机备机,都需要把接收到的所有网络数据记录到备份日志中;当需要恢复数据时,就可以按照备份日志恢复数据。

(5) 当计算机启动时,按照主/备机自动识别协议将本机标识为主机或备机。

(6) 当出错计算机修复后,按照主/备机自动识别协议将本机标识为备机,备机向主机发送恢复数据请求,主机把备份日志传给备机,备机按照备份日志恢复数据。

在大多数双机容错系统中都应用到了心跳协议。由于篇幅有限,在这里就不再详述。详细内容可以参考文献[2]。

3.5 双心跳线

双机容错系统的设计方法,无非是为单点故障设备提供冗余,心跳线当然也不例外。在本系统中,采用双心跳线的心跳线冗余方式。

当接收不到心跳信号时,故障原因可能有两种,一种是对方计算机软硬件的故障;另一种是心跳线本身的故障。要直接判断究竟是哪一种故障比较困难,于是本系统采用双心跳线加仲裁的方法来进行故障判断。

用两根串口线作为心跳线连接主备机,心跳数据在两条心跳线中同时发送与接收,该过程分别由两个线程控制。用第三个线程与这两个线程通信,进行仲裁,按照两个线程能否接收到心跳数据来判断故障种类。当两个线程都能收到心跳数据时,说明没有任何故障;当两个线程中的任意一个线程不能收到心跳数据时,就说明是心跳线故障;当两个线程都不能收到心跳数据时,就说明是对方计算机软硬件的故障。

提供冗余心跳线是很有必要的,否则当心跳线出现故障时(如串口线松动等),主备机就都不能收到对方的心跳数据,都会认为对方出错,备机就会切换到主机。这时就会出现同时存在两台主机的异常情况,所以必须避免这种情况,为心跳线提供冗余。

3.6 主/备机自动识别协议

如图 3 所示,开机后启动热备软件,如果在定时器内收到心跳数据,就表明已经有主机在运行,所以设置本机为备机;如果定时器内没有收到心跳数据,表明还没有主机运行,所以设置本机为主机。无论主机还是备机,都按照心跳协议工作。

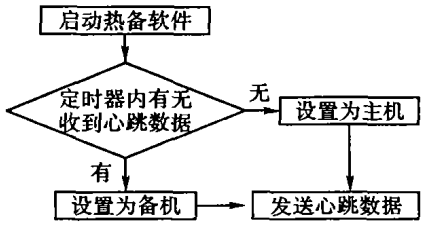

图3 主备机自动识别协议

通过接口可以设置主机、备机的标识变量 ActiveStandby,容错软件就可以按照 ActiveStandby 的值进行相应处理。

文章编号:1001 - 9081(2005)08 - 1951 - 03

虚拟现实中一种基于运动混合的实时同步算法

王忆源[1],陈福民[2]
(同济大学 计算机科学与技术系,上海 200092)
(tjyuanyuan@ hotmail. com)

摘 要:在虚拟现实中,采用运动捕获系统建立基本运动库,然后通过运动编辑技术对基本运动进行处理。运动混合技术是编辑技术中最为实用也最为复杂的一种。提出了基于运动混合的实时同步算法,以便更好地动态混合运动,避免产生未预期的效果,以创建复杂的虚拟动画。

关键词:虚拟现实;运动混合;群体动画;时间偏差;优先级
中图分类号: TP391.9　　**文献标识码**:A

Real-time synchronization algorithm based on blending of motions in VR

WANG Yi-yuan[1], CHEN Fu-min[2]
(*Department of Computer Science and Technology, Tongji University, Shanghai* 200092, *China*)

Abstract: In virtual reality, a fundamental motion library is built by using motion capture system. Then appropriate fundamental motions are selected from the library, and desired motions are synthesized from these fundamental ones. One of these motion editing techniques, motion blending, is the most practical and complex one. A real-time synchronization algorithm was presented based on blending module. The basic idea is to make the data set more flexible and to dynamically blend motions to avoid unexpected ones. As a result, complex virtual animations are created from above.

Key words: VR(Virtual Reality); motion blending; group animation; time warping; priority

1 运动混合技术中的同步问题

当今的虚拟现实系统中,运动捕获技术已经作为一种强大的制作手段被广泛运用于视频游戏和电影制作等相关行业中。当然,这项技术遵循某种动态规则,而且通过运动编辑技术对捕获的运动进行处理,使得虚拟动画更为逼真。在众多编辑技术中,运动混合技术使得我们可以从捕获的简单运动中创造出复杂的动画效果。但目前的混合算法还不成熟,往往带来超出预料的运动错误描述。比如,当双脚都接触地面,将左脚的运动与右脚的运动混合,作为混合结果的运动将无法正确反映原始双脚的状态,而是在地面上整体滑动。因此,为了避免类似错误的出现,运动的动态同步混合就显得相当重要。

在实时运动的动态环境中,动画所采样的基本运动不仅是交互自适应的,而且受到外部条件的制约,随着外部环境的变化而变化。单个角色已经如此,群体动画中不同形态的角色动画就更为复杂。基于运动混合的实时同步算法提出,就是为了改善多角色多运动之间的协调,尽可能的减少运算的复杂度。此算法采用了基于运动相位的时间偏差技术[1],同时通过为基本运动设置优先级,实现在任何时刻可以使用开/关命令激活/解除运动的有效性。在运动混合技术和群体动画领域显得相当重要。

2 一种基于运动混合的实时同步算法

2.1 相关工作

之前对同步运动的研究工作主要通过两种途径:识别运

收稿日期:2005 - 02 - 05;修订日期:2005 - 04 - 26

作者简介:王忆源(1980 -),男,江苏人,硕士研究生,主要研究方向:网络多媒体; 陈福民(1945 -),男,上海人,博士生导师,主要研究方向:网络多媒体.

3.7 备份与恢复备份

备份:主机和备机都要进行备份工作,将所有接收到的网络数据都写入备份日志;当接收到正确心跳数据时,就将该时间点写入备份日志,该时间点是用于备份恢复的。

恢复备份:按照日志内容,从最后一次收到正确心跳数据的时间点开始恢复备份。

在大多数双机容错系统中都应用到了备份与恢复备份。由于篇幅有限,在这里就不再详述备份和恢复备份的内容。

4 结语

使用相同 IP 的双机容错系统,减小了 IP 切换的时延,实时性更强。在双机双工工作模式下,通过修改 Linux 内核的网络,使备机不发送数据,且对上层透明,备机丢失网络数据的可能性也得到了降低。该系统相对于通过主机转发网络数据给备机的双机容错系统,具有更高的可靠性。本系统可广泛应用于数据量不太大的网络服务环境,为服务器提供持续、稳定的服务。

参考文献:

[1] Droms R. Dynamic Host Configuration Protocol, RFC 2131[S]. 1997.
[2] 徐立云,邵惠鹤. 双机容错系统的一种实现途径[J]. 计算机工程,2000,26(9).
[3] Comer DE. 用 TCP/IP 进行网际互联 第一卷:原理、协议与结构(第四版)[M]. 林瑶,蒋慧,杜蔚轩,等译. 北京:电子工业出版社,2001.
[4] 骆耀祖. Linux 操作系统分析教程[M]. 北京:清华大学出版社,2004.
[5] 王存祥,樊婷婷. 校园网 IP 地址冲突的分析及解决方案[J]. 中国电化教育,2004,(7).