

MMU虚拟内存的方式对I/O进行虚拟化。这个硬件解决了DMA引起的问题。

另一种处理I/O操作的方法是让其中一个虚拟机运行标准的操作系统，并把其他虚拟机的I/O请求全部反射给它去处理。当准虚拟化技术得到运用之后，这种方法被完善了，发送到管理程序的命令只需表明客户操作系统需要什么（如从磁盘1中读取第1403块），而不必发送一系列写磁盘寄存器的命令，在这种情况下，管理程序扮演了福尔摩斯的角色，指出客户操作系统想要做什么事情。Xen使用这种方法处理I/O操作，其中完成I/O操作的虚拟机称为domain0。

在I/O设备虚拟化方面，II型管理程序相对于I型管理程序所具备的优势在于：宿主操作系统包含了所有连接到计算机上的所有怪异的I/O设备的驱动程序。当应用程序试图访问一个不常见的I/O设备时，翻译的代码可以调用已存在的驱动程序来完成相应的工作。但是对I型管理程序来说，它或者自身包含相应的驱动程序，或者调用domain0中的驱动程序，后一种情况与宿主操作系统很相似。随着虚拟技术的成熟，将来的硬件也许会让应用程序以一种安全的方式直接访问硬件，这意味着驱动程序可以直接链接到应用程序代码或者作为独立的用户空间服务，从而解决I/O虚拟化方面的问题。

8.3.7 虚拟工具

虚拟机为长期困扰用户（特别是使用开源软件的用户）的问题提供了一种有趣的解决方案：如何安装新的应用程序。问题在于很多应用程序依赖于其他的程序或函数库，而这些程序和函数库本身又依赖于其他的软件包等等。而且，对特定版本的编译器、脚本语言或操作系统也可能有依赖关系。

使用虚拟机技术，一个软件开发人员能够仔细地创建一个虚拟机，装入所需的操作系统、编译器、函数库和应用程序代码，组成一个整体来运行。这个虚拟机映像可以被放到光盘（CD-ROM）或网站上以供用户安装或下载。这种方法意味着只有软件开发者需要了解所有的依赖关系。客户得到的是可以正常工作的完整的程序包，独立于他们正在使用的操作系统、各类软件、已安装的程序包和函数库。这些被包装好的虚拟机通常叫做虚拟工具（virtual appliance）。

8.3.8 多核处理机上的虚拟机

虚拟机与多核技术的结合打开了一个全新的世界，在这个世界里可以在软件中指定可用的处理机数量。例如，如果有四个可用的核，每个核最多可以支持八个虚拟机，若有需要，一个单独的（桌面）处理器就可以配置成32结点的多机系统，但是根据软件的需求，它可以有更少的处理器。以前，对于一个软件设计者来说，先选择所需的处理器数量，再据此编写代码是不可能的。这显然代表了计算技术发展的新阶段。

虽然还不普遍，但是在虚拟机之间是可能实现共享内存的。所需要完成的工作就是将物理页面映射到多个虚拟机的地址空间当中。如果能够做到的话，一台计算机就成为了一个虚拟的多处理机。由于多核芯片上所有的核共享内存，因此一个四核芯片能够很容易地按照需要配置成32结点的多处理机或多计算机系统。

多核、虚拟机、管理程序和微内核的结合将从根本上改变人们对计算机系统的认知。现在的软件不能应对这些想法：程序员确定需要多少个处理机，这些处理机是应该组成一个多计算机系统还是一个多处理机，以及在某种情况下最少的内核数量需求到底是多少。将来的软件将处理这些问题。

8.3.9 授权问题

大部分软件是基于每个处理器授权的。换句话说，当你购买了一款程序时，你只有权在一个处理器上运行它。这个合同允许你在同一台物理机上的多个虚拟机中运行该软件吗？在某种程度上，很多软件商不知道应该怎么办。

如果某些公司获得授权可以同时能在 n 台机器上运行软件，问题就会更糟糕，特别是当虚拟机按照需要不断产生和消亡的时候。

在某些情况下，软件商在许可证（license）中加入明确的条款，禁止在虚拟机或未授权的虚拟机中使用该软件。这些限制在法庭上是否有效，以及用户对此的反应还有待考察。

8.4 分布式系统

到此为止有关多处理机、多计算机和虚拟机的讨论就结束了，现在应该转向最后一种多处理机系统，