

# 从资源共享到服务共享:透明计算的机遇与挑战

张尧学, 周悦芝

(清华大学计算机科学与技术系 清华信息科学与技术国家实验室, 北京 100084)

**[摘要]** 随着计算机和网络技术以及应用的不断快速发展, 计算模式从最初的以主机计算为主, 发展到以桌面(PC)计算为主, 再发展到现在的以网络计算为主, 乃至普适计算。然而, 迄今为止的计算模式还是遵循以单机或单机联合为基础的单机思维模式, 因此, 目前的计算模式还是一种以“资源共享”为主要特征的计算。在网络和普适计算的新时代, 用户关心的主要是服务和资源共享, 计算机也将以“资源共享”为特征的计算过渡到以“服务共享”为特征的计算。笔者提出一种适应于“服务共享”的计算模式——透明计算, 并对其概念、体系结构及相关关键技术进行介绍。此外, 还对透明计算的未来发展及影响进行展望。

**[关键词]** 普适计算; 资源共享; 服务共享; 透明计算

**[中图分类号]** TP 202 **[文献标识码]** A **[文章编号]** 1009-1742(2009)08-0010-08

## 1 前言

计算模式的发展与变迁总是在两个驱动因素下不断发展变化的结果, 一个是技术进步, 另一个就是应用需求。应用需求的发展拉动和促进了计算机和信息技术的发展, 而计算机、网络以及其他信息技术的创新和发展又促使新型应用的产生与普及。两者相互影响、相互促进, 不断发展变化, 共同促进了计算模式的发展与演化。

在计算机发展的早期, 由于受硬件制造技术的限制, 因此当时的计算机不但体积庞大而且价格非常昂贵, 许多机构都只能拥有一台或几台计算机, 也就是大型机。为了共享和使用这种计算机, 人们需要通过哑终端(dumb terminal)和调制解调器(modem)连接到中央主机并通过分时系统来共享中央主机的处理能力。在这一时期, 人们对计算机的应用需求仅限于科学计算与军事应用<sup>[1]</sup>。

微处理器以及微型计算机如PC(personal computer)的出现极大地推动了计算机的应用和发展, 以文字处理为代表的计算机应用开始渗透到社会

的各个领域。后来网络技术的发展又使得这些计算机通过网络连接并相互通信。人们对计算机的使用不仅限于文字处理和企业信息管理, 而是利用计算机开始相互沟通、获取信息与共享信息和资源。

移动设备和数字家电等的出现使得计算机进一步小型化、微型化, 种类和形态变得日益丰富多彩。泛在网络(ubiquitous network)和普适计算(pervasive computing)<sup>[2,3]</sup>时代已经来临。在这种计算模式下, 计算将变得不可见(透明), 用户将不再要求资源, 而只关心服务, 特别是个性化服务。例如, 如何将手机拍摄的高清晰照片及时存储和处理? 如何像选择电视台那样选择自己所需的操作系统? 如何减少软件安装、维护和使用的复杂度? 如何增强安全性和防护病毒等。

用户希望从任何一台计算设备上都能获取所需要的服务, 或者能够从任何计算设备上都能获得同一服务, 而且无需关心这些服务的来源、安装以及维护管理等问题。我们将这种以人或服务为中心的计算需求称为服务共享。

虽然泛在网络和普适计算时代已经来临, 然

**[收稿日期]** 2008-11-30; **[修回日期]** 2008-12-20

**[基金项目]** 国家“八六三”资助项目(2004AA114062, 2005AA114160)

**[作者简介]** 张尧学(1956-), 男, 湖南澧县人, 中国工程院院士, 主要研究方向为计算机网络、操作系统和普适计算等; E-mail: zyx@moe.edu.cn

而,如何在这一时代实现以人为本的服务共享,仍是一个悬而未决的问题。国内外都在为此进行研究。例如,虚拟机计算<sup>[4,5]</sup>是VMWare,HP和Intel等公司提出的一种能够在—个硬件平台上提供多种操作系统和应用服务的努力。但是虚拟机计算要求非常高性能的计算平台和高带宽的网络才能使用,因此不适合在小设备和低带宽网络环境下实现。再如,近年来出现的网格计算<sup>[6]</sup>也主要是从资源共享的角度将多台计算机的资源整合起来供用户使用,但是它在部署、开发和使用方面也还比较复杂,难于推广到普通用户,如个人和企业的使用。

此外,其他一些工作也都还存在不足之处。例如,Google,IBM等公司最近提出了云计算<sup>[7]</sup>,它通过将应用以及数据放在“云端”并通过浏览器为用户提供各种搜索、地图或其他服务。但是,云计算主要是解决了服务和计算分离的问题,而没有能够解决存储和计算分离以及服务的任意选择等问题,从而还是不能实现完整意义的服务共享。

透明计算从单机思维转换为以系统为基础的网络思维,将过去一台台独立计算机之间通过网络的联合,变成在网络环境下终端、网络和服务器等系统的联合。前端面向用户,提供服务;操作系统和应用服务的存储和支持等工作放在后端的服务器上来完成。这样使得存储和计算在时间和空间上分离。

## 2 透明计算

### 2.1 透明计算的概念

透明计算有以下3方面的含义与特点:

1) 存储与计算分离。存储与计算分离是指操作系统和应用程序的存储和执行分离<sup>[8]</sup>。一般来说,传统计算模式下,计算机的操作系统和应用程序的存储和执行(计算)都在同一台计算机中进行,如图1(a)所示。在透明计算模式下,操作系统和应用程序的存储是在服务器的存储器中,执行则是在面向用户的前端(终端)进行,如图1(b)所示。

为了实现将操作系统存储在后端服务器上,并将其调度到前端执行,需要一个分布式的管理平台来进行相关的调度与管理工作。为了方便后面的说明,引入以下几个定义。

定义1:运行在终端上的传统操作系统,如Windows 2000/XP, Linux, Unix, 以及Symbian等被称为实例操作系统(instance OS, IOS)。

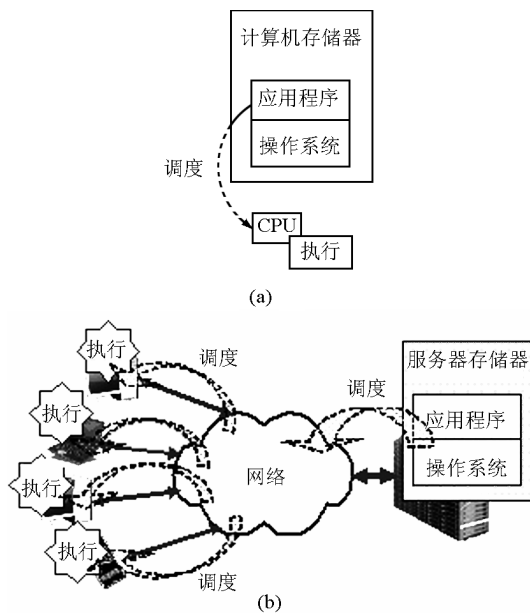


图1 存储与计算分离

Fig. 1 Decoupling of computing and storage

定义2:将实现支持操作系统后端存储,前端调度运行的管理平台称为Meta OS (MOS)。MOS管理其他IOS,以及透明终端和透明服务器上的指定资源和服务。

定义3:由位于前端的终端,位于后端的服务器,以及中间进行传输的网络组成的透明计算系统中的终端被定义为透明终端(transparent client, TC),后端服务器被定义为透明服务器(transparent server, TS),而将连接它们的网络定义为透明传输网络(transparent delivery network, TDN)。

因此,存储和计算分离主要是指在MOS的支持下,IOS(包括其上的应用程序)的存储是在透明服务器上,而其执行则是通过透明传输网络被调度到透明终端上进行。

2) 操作系统分布式装载、动态调度执行。操作系统的分布式装载是指MOS的一部分被预先装载在TC上,而另外一部分则被装载在TS上。在传统计算模式下,计算机执行的操作系统全部都被装载在本地的存储器(例如硬盘)上,如图2(a)所示。在透明计算模式下,MOS( $\{M_1, M_2, M_3, M_4, M_5, \dots, M_n\}$ )的一部分,例如图2(b)中 $\{M_1\}$ ,  $\{M_2\}$ ,  $\{M_3\}$ ,  $\{M_4\}$ 分别被装载在不同的透明终端上。MOS的绝大部分,如图2(b)中的 $\{M_4, M_5, \dots, M_n\}$ 则被装载在透明服务器上。装载在透明终端上的MOS部分,主要是完成透明终端的启动,将透明终端的启动中

断发送到透明服务器上。装载在透明服务器上的 MOS,可以在需要的时候被调度到透明终端执行,这就是分布式动态调度执行的含义。

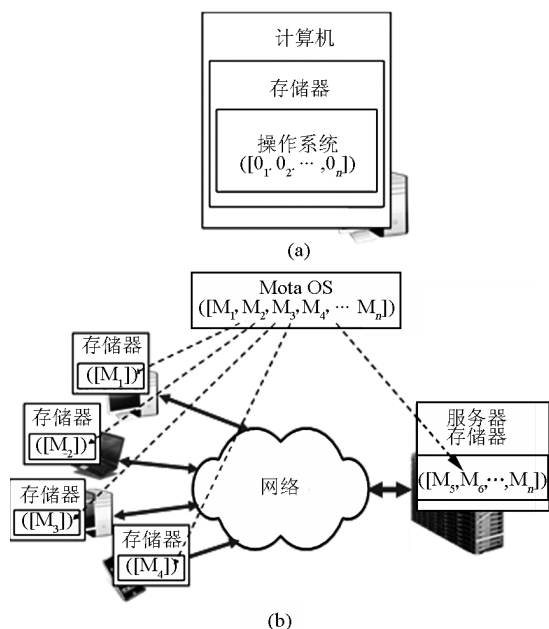


图2 透明计算中 Meta OS 的分布式装载  
Fig. 2 Distributed storage of Meta OS in transparent computing

操作系统的分布式动态调度执行的含义是指装载在透明服务器上的 MOS 或 IOS 代码或数据\*,可以在透明终端需要的时候才被动态调度到透明终端的内存中执行。由于不是一次全部将操作系统代码下载完毕后才执行,这样就可以在小终端上,运行较大的操作系统或应用程序。

分布式动态调度执行的具体过程如图3所示。当前端的终端设备在运行的时候,如果发现所需的 MOS 或 IOS 的指令或数据块在自己的内存中不存在的时候,则向透明服务器发出请求中断,如图3(a)中的①所示。透明服务器在监听并接收到透明终端的请求后,将从自己的内存中或硬盘上读取相应的指令或数据块,通过透明传输网络发送给透明终端,如图3(b)中的②所示。在透明终端接收到透明服务器传送过来的操作系统指令块时,将它们加载到透明终端的内存中执行,如图3(b)中的左侧部分所示。图3(a)中的  $\{M_1\}, \{M_2\}, \{M_3\}, \{M_4\}$  分别变成了图3(b)中的  $\{M_1, M_i\}, \{M_2, M_j\}, \{M_3, M_k\}, \{M_4, M_l\}$ 。由于每次都只取所需要的指令或者数据块,因此上述过程会在透明终端和服务器的间随着计算的进行而循环反复,形成基于“块流”的

动态调度。

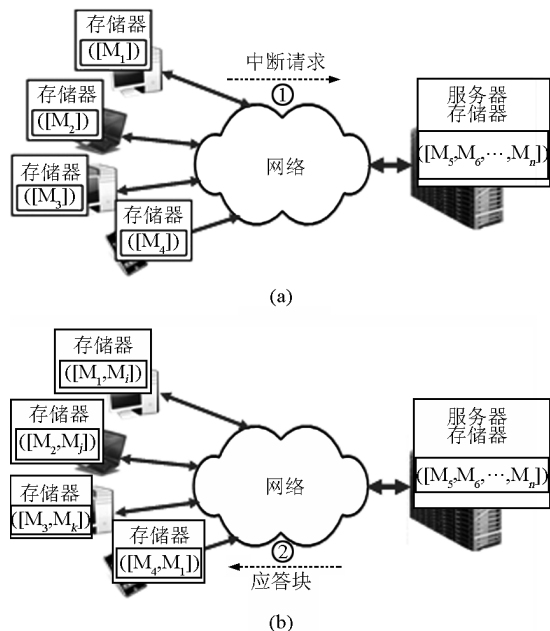


图3 透明计算中操作系统的分布式调度运行  
Fig. 3 Distributed scheduling of OSes in transparent computing

3) 计算机主板与多操作系统的接口标准化。在传统计算模式中,操作系统和主板之间的关系基本上是一个一一对应的关系,如图4(a)所示,BIOS,硬件平台(x86为例)和操作系统(以Windows为例)是一个一一对应的绑定关系。在透明计算模式中,用户可以通过透明终端自由选择并动态运行各种实例操作系统,因此操作系统与计算机主板之间不再存在一种紧密的绑定关系。要实现不同的操作系统能够在同一主板上运行,就需要实现各种操作系统与主板之间的无缝连接。这就涉及到底层接口的标准化问题,如图4所示。如果实现了标准化的接口,例如,图4中的EFI(extensible firmware interface)<sup>[9]</sup>接口,在不同架构的计算机主板(例如x86,MIPS,以及数字电视板)上就都可以运行多种实例操作系统,如Windows, Linux, Symbian等。这样不同设备终端,无论是PC,还是手机,甚至是新研制的终端设备,只要遵循标准接口,接上网络就很容易从透明服务器上动态调度不同的实例操作系统及其上的应用到本地内存中执行。

## 2.2 透明计算的长处

根据上面关于透明计算的讨论,可以得出透明

\* IOS 上应用程序的代码和数据也是动态调度执行的。

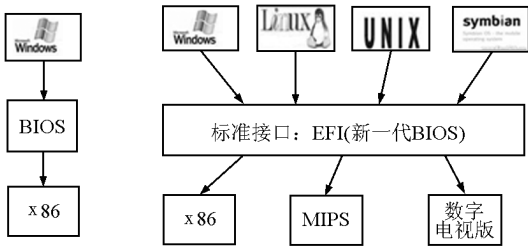


图4 计算机主板与操作系统之间的接口标准化

Fig. 4 Interface standardization between computer motherboard and OSes

计算的长处如下:

1) 由于在透明计算中操作系统和应用软件等都存储在服务器上,因此终端上可以不用存放任何指令和数据,而且在必要的时候还可以利用服务器的计算能力,因此终端就可以变得很小。利用很小的 CPU 就可以完成多样的功能,从而能够降低硬件成本。此外,由于操作系统和应用软件的维护都集中在服务器端完成,终端用户无需任何维护,因此对终端用户而言,维护很方便,成本也很低。

2) 由于小终端只在需要时才从服务器上调度不同操作系统和应用代码到其内存中执行,因此这样完成的实际上是一个虚拟计算。服务器上面的操作系统、应用和服务都可以供多个用户共享。因此在透明计算模式下,可以按照用户的运行来收费,而无须像过去那样,无论用户是否需要,都要购买包含所有功能的一整套软件和拷贝,如 Windows 和 Office 套件。因此,透明计算不但使用方便,而且也能够有效降低软件成本。

3) 在透明计算模式下,一套软件,包括操作系统和应用程序,可以由多个用户共享,用户可以自由选择和运行,因此实现了软件即服务(SaaS, software-as-a-service)。用户只需要购买使用权,按照实际购买和使用的情况来付费。

4) 在透明计算模式下管理工作是集中进行的。所有的操作系统和应用对终端用户来说是透明的,终端用户只需考虑如何使用。管理工作,包括安全性和可靠性的问题,都由专业的技术人员集中来进行处理。这可以提高整个系统的安全性和可靠性。此外,透明计算特有的结构,还可以增加新的安全机制,如基于网络数据流的监控等。

### 3 透明计算的关键技术与实现

透明计算中的关键技术主要涉及到透明终端硬件、硬件与操作系统之间的接口,以及操作系统 3

个层次。这 3 个方面的关键技术主要是: Meta OS 技术,操作系统与硬件主板之间的无缝连接技术,透明终端技术。下面分别加以介绍。

#### 3.1 Meta OS

##### 3.1.1 Meta OS 的概念

Meta OS 就是管理和调度其他 IOS 并对透明终端和透明服务器上指定软硬件资源进行管理的超级操作系统。它具有如下 3 个特点:

1) Meta OS 在空间上是分布式的,即 Meta OS 的各个部分分别装载在不同的透明终端和透明服务器上。

2) Meta OS 各部分之间靠网络协议协调工作。在 Meta OS 各部分之间,可以交换任何信息,包括数据信息,以及指令或中断信号信息等。

3) 从体系结构上来说,Meta OS 的运行层次在 IOS 之下。

MOS 与 IOS 的关系如图 5 所示。图中的  $TS_1, TS_2, TS_n$  表示不同的透明服务器;  $IOS_1, IOS_i, IOS_n$  表示不同的 IOS; TC 则表示透明终端。从图 5 中可以看出, MOS 的一部分装载在透明终端上,而另外一些部分则装载在透明服务器上。需要注意的是,图 5 中虽然只画出了一台透明终端来进行示意性说明,在实际系统中会存在多个透明终端。由于 MOS 的各个部分被装载在不同的机器上,因此它们之间需要通过网络通信来协调各个任务之间的运行以及相应的指令和数据交换。存储在透明服务器上的 IOS,将在用户选择后动态调度到透明终端上执行。在透明终端上, MOS 在 IOS 之下运行,为 IOS 提供 IOS 指令和数据的传输工作。图 6 表示了 MOS 在透明计算系统中的层次结构。为了简单起见,只画了一台透明服务器和几台透明终端的情形。

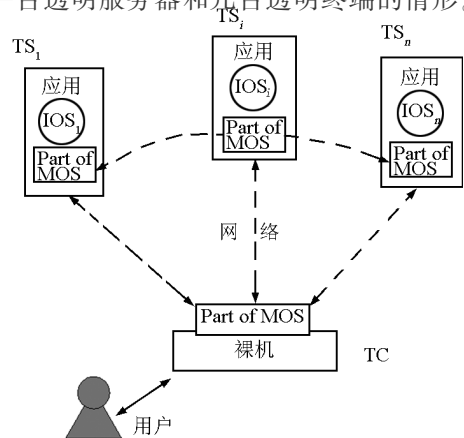


图5 MOS 与 IOS

Fig. 5 MOS and IOS

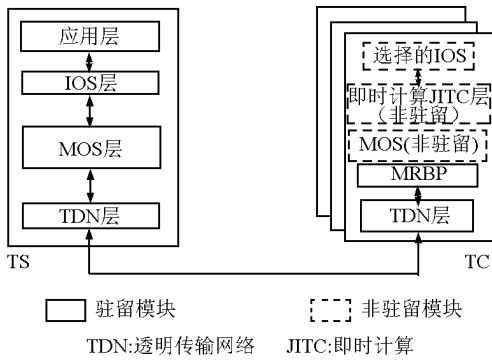


图6 MOS在透明计算中的层次结构

Fig. 6 Layout of MOS in the transparent computing system

在图6中,最下面是一个透明传输网络TDN(transparent delivery network),它是满足一定带宽和时延要求的网络。在透明服务器(TS)上,除了TDN层之外,从下到上分别为MOS层,IOS层,以及应用层。这些层的代码都驻留(提前装载)在透明服务器上,在图6中用实线框表示。在透明终端(TC)上,在TDN层之上,是驻留的MOS代码部分,即图6中的MRBP(multi-OS remote booting protocol)。它是Meta OS实现时的一个组件,它的主要功能是完成透明终端的启动并负责让用户选择需要运行的IOS。在MRBP之上,运行的是非驻留的MOS以及即时计算(just-in-time computing, JITC)层(此处非驻留的含义是这些层的代码不是事先装载在透明终端上的,而是从透明服务器上调度过来进行执行的)。TC上非驻留的MOS层的功能主要是维护与TS上MOS层的网络连接,并协同管理透明终端上的软硬件资源。非驻留的JITC层的主要功能是通过MOS为IOS的运行提供指令和数据的即时传送。在TC的JITC层之上,是用户选择运行的IOS。无论是在TC上运行的非驻留的MOS层、JITC层还是IOS层的代码实际上都驻留在TS上,而只在需要的时候才被动态调度到TC上的内存中执行,因此在图6中用虚线框表示。

### 3.1.2 Meta OS的组成

由于Meta OS是一个分布式装载、动态调度执行的操作系统,因此,在透明终端和透明服务器上,都会装载和运行Meta OS的不同部分。为了方便描述,将Meta OS按照运行的位置,划分为Meta OS客户端(运行在透明终端上)和Meta OS服务器端(运行在透明服务器上)。

Meta OS有多种实现方式,在此将介绍一种基于4VP<sup>+</sup>(4 virtualization layer, 2 protocols)<sup>[10]</sup>的实现。

在这种实现方式下,Meta OS的功能被划分为两个协议和4个虚拟层,如图7所示。需要注意的是,由于Meta OS是一个分布式的操作系统,因此每个协议和每个虚拟层都会具有客户端和服务端。而且客户端和服务端是一种多对多的关系。因此图7只是一个从总体上看Meta OS组成的示例性说明。

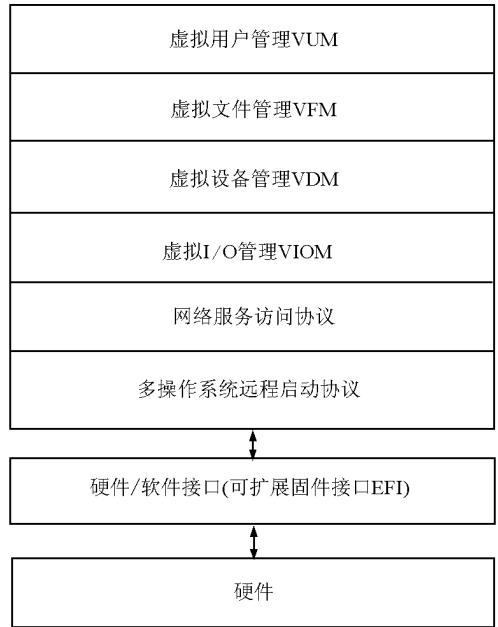


图7 4VP<sup>+</sup>的组成

Fig. 7 The composition of 4VP<sup>+</sup>

从图7中可以看出,两个协议分别为多操作系统远程启动协议MRBP(multi-OS remote booting Protocol)和网络服务访问协议NSAP(network service access protocol)。MRBP的主要功能是捕获透明终端的初始启动中断信号,发现透明服务器上的实例操作系统列表,下载NSAP协议代码并通过它获取实例操作系统指令到透明终端内存中执行,完成实例操作系统的启动。因此,MRBP客户端的一部分必须提前驻留在透明终端的主板BIOS或独立的芯片中。由于MRBP工作时,实例操作系统并未启动,因此MRBP客户端不能像传统协议一样工作在操作系统环境中,因此MRBP客户端必须实现自己的独立于实例操作系统的网络通信功能,包括网络设备驱动和相关底层协议栈。

NSAP协议主要的功能是在透明终端和透明服务器间传递Meta OS本身或者用户所选的实例操作系统的指令、数据以及信号等。通过MRBP和NSAP两个协议,就在透明终端和透明服务器间建立起了独立于实例操作系统的网络通信连接,并

为实例操作系统的可选择自由调度运行提供了指令和数据传输的基础。

4个虚拟层主要是共同协作,完成对透明终端和透明服务器上软硬件资源的管理和调度功能。4个虚拟层分别是:虚拟I/O管理VIOM(virtual I/O management),虚拟设备管理VDM(virtual disk management),虚拟文件管理VFM(virtual file management),以及虚拟用户管理VUM(virtual user management)。下面仅简单介绍每个虚拟层的功能。

1)VIOM。主要的功能是接收透明终端上中断请求和用户进程的I/O请求;分析中断原因,并唤醒相应处理程序进行处理,同时进行相关设备分配和缓冲处理,启动I/O操作。与在普通单机环境下不同的是,VIOM必须处理不同种类的中断和I/O请求,而且需要在透明终端和透明服务器之间管理多级队列并进行相应的同步互斥处理。

2)VDM。VDM主要有3个方面的功能:第一个主要是完成虚拟设备(virtual device,VD)的分配与回收,以及透明终端上的虚拟设备驱动;其次,是完成透明终端上虚拟设备到服务器端虚拟设备镜像之间的映射及管理;再次,是完成透明终端与服务

3)VFM。VFM的功能是完成文件空间以及文件目录的分配与管理,以及文件访问的重定向工作。此外,VFM还需要保证文件访问的一致性,以及提供文件的访问、控制与搜索功能。

4)VUM。VUM主要提供对用户参数配置的管理,以及透明终端地址的分配。此外,需要完成系统中关于用户的增加和减少等日常管理工作。最后,VUM还需要提供对用户使用过程的管理等。

### 3.1.3 Meta OS 的实现

图8表示了基于4VP<sup>+</sup>的一个简单的实现框架。为了简单起见,图8只画出了一(TS,透明服务器)对一(TC,透明终端)的情况。在透明终端侧,所有模块都运行在透明终端的内存中。为了让用户选择实例操作系统,并从透明服务器上调度Meta OS其他模块到透明终端内存中执行,需要首先利用MRBP进行透明终端的启动。

在透明终端上,所有模块都是从透明服务器上动态调度到其内存中运行,包括相应的虚拟I/O管理模块、虚拟文件管理模块。它们通过与透明终端上运行的虚拟设备驱动,文件重定向器的交互,来完成相应的调度和管理功能。所有透明终端内存中的模块与透明服务器的交互,均通过NSAP来进行。

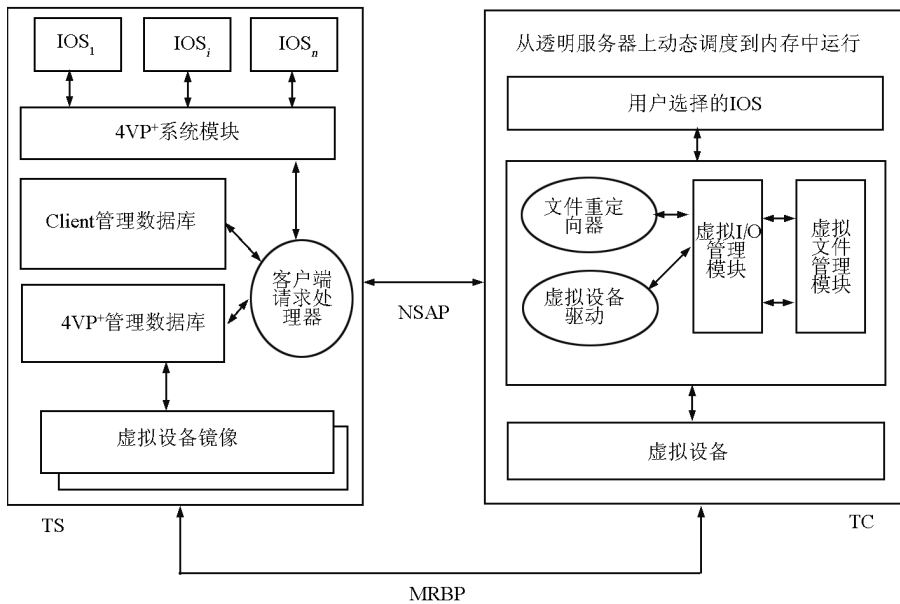


图8 Meta OS的实现结构

Fig. 8 Implementation structure of Meta OS

在透明服务器上,主要是存储各种实例操作系统,并通过客户端请求处理器来监听、接收以及处理来自透明终端上的各种请求。客户端请求处理

器将协同其他模块,如4VP<sup>+</sup>系统模块,以及各种管理数据库等一起来完成透明终端各种请求的处理工作,并将处理结果通过NSAP发送给透明终端。

上面只是介绍了一种简单的实现框架。我们从2000年开始,已经在多个自制的透明终端平台上实现了Meta OS,并能够支持Windows 2000/XP,以及多种Linux等实例操作系统,并在政府、金融、军事、企业以及农村信息化中得到了广泛的应用,取得了较好的社会和经济效益。

### 3.2 操作系统与硬件主板之间的无缝连接

透明计算的另一个关键技术就是操作系统与硬件主板之间的无缝连接。无缝连接的含义就是通过标准化的接口技术,能够使得各种实例操作系统都能够在透明终端上稳定运行。这里至少需要考虑如下3个问题。

1) 可扩展性问题。可扩展性问题是如何实现对新的软硬件平台的兼容,即很容易能够扩展到多种终端和多种软件平台上。由于在普适计算时代,新的设备和软件平台可能会不断增多,因此如何实现可扩展性,也就成了一个必须考虑和解决的问题。

2) 效率问题。效率问题是当终端的处理能力,例如,CPU比较小时,如何能够定义合适的接口,而使得它能够支持多种操作系统的高效运行。不但能够让小设备运行较大的操作系统,还能使用户感觉不到任何效率的降低。

3) 安全性问题。由于存储和运算的分离,因此原来在单机内部总线上流动的指令和数据流变成了在透明计算系统中透明网络上流动的指令和数据流。这里所带来的安全性问题是两个方面的。首先,由于原来在单机中流动的指令和数据流难以监控和管理,这导致传统安全策略和模型中监视点的设置在实际中很难有效实现。而在透明计算模式下,能够在透明网络或透明服务器上对流经的指令和数据流都能进行有效的监视和控制,而不会受到透明终端上运行的各种代码的干扰和威胁。其次,指令和数据流在网络上的流动,也给透明计算的安全带来了一些潜在的威胁。因此需要通盘考虑透明计算中因为这种操作系统和主板之间接口的开放而导致的安全性问题。

### 3.3 透明终端

透明计算中的另一个关键技术就是透明终端技术。如何研制低成本、低功耗,品种繁多,大小、功能各异的透明终端,是透明计算能够得到广泛应用的前提和基础。

基本上现有的所有计算设备都可以应用到透明计算模式中,包括移动上网设备MID(mobile in-

ternet device),轻量级计算机终端LID(light-weight internet device),以及数字家电设备DHD(digital home device)等。

在透明计算模式下,通过一个透明终端,就可以访问多种服务,实现服务共享,因此可以简化硬件设备的数量和复杂度,提高服务质量。

## 4 透明计算的影响与展望

以下将对透明计算在学术和产业方面的影响进行分析。

### 4.1 对学术的影响

透明计算将过去以单机为基础的存储运算扩展为以网络为基础的存储运算,提出并成功实现了存储与运算在时空上的分离,这将给计算带来许多概念上和结构上的变化。

这里面还有许多问题值得深入研究。以I/O管理为例,原来在单机时候可能是一个或几个队列,而在透明计算模型里,可能从一个队列变成了上百个甚至更多的队列。其次,由于存储和运算的分离,指令和数据流都要从网络上经过,因此授权模型也会发生变化。此外,如何实现从资源共享跨越到服务共享,这之间也还有许多工作要作。因此,从学术上来说,如何完善透明计算的理论和模型,形成一个用户不可见,能够实现服务共享的计算,还需要进一步的研究和完善。

### 4.2 对产业的影响

从产业上来说,透明计算能否催生新的产业链?最早的科学计算带来了程序的开放,带来了大型机的发展;文字处理和办公自动化,带来了DOS、Windows和PC的发展;信息共享产生了互联网,带动了Google,百度,亚马逊等公司的发展。那么,在服务共享时代,也会诞生新的公司,带动相关产业和产业链的发展。

从透明计算来说,首先是前端系统,可以做出很多新的终端设备,如MID,数字家电等。其次,可以应用到许多新的领域,如移动互联网、数字家庭等。这种新市场是一个巨大的潜在市场。再次,对于透明服务器以及提供的服务而言,也可以产生新的需求和市场。因此,在合适的条件下,透明计算会对信息产业的发展产生新的巨大影响,将拉动信息产业链、甚至产业群的发展。

## 5 结语

分析了计算模式从资源共享到服务共享的发

展变化趋势,并针对该趋势提出了一种能够有效解决服务共享问题的透明计算模式。通过介绍透明计算的概念,关键技术以及实现,我们可以看出透明计算是在用户界面、系统部署、维护 and 安全性等方面,具有传统计算模式所不可比拟的优势,能够从根本上解决服务共享面临的诸多问题。因此,在新的历史条件下,透明计算具有巨大的机会和发展潜力。但是如何发挥透明计算的优势,使其能够发挥真正的作用,促进计算技术的发展,并带动相关信息产业链的发展,也是一个具有挑战性的问题。

#### 参考文献

- [1] Ceruzzi P E. A History of Modern Computing 2nd Edition [M]. Cambridge: MIT Press, 2003
- [2] Dertouzos M. The future of computing [J]. Scientific American, 1999, 282(3): 52 - 63
- [3] Weiser M. The computer for twenty - first century [J]. Scientific American, 1991, 265(3): 94 - 104
- [4] Graupner S, Pruyne J, Singhal S. Making the utility data center a power station for the enterprise grid [R]. Technical Report HPL - 2003 - 53, Hewlett Packard Laboratories, 2003
- [5] Sugerma J, Venkitachalam G, Lim B H. Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor [A]. 2001 USENIX Annual Technical Conference [C]. Boston, Massachusetts, USA, 2001, 4(25 - 30): 1 - 14
- [6] Foster I, Kesselman C. The Grid: Blueprint for a New Computing Infrastructure [M]. San Francisco: Morgan Kaufmann, 2004
- [7] Meiss A. Computing in the clouds [J]. netWorker, 2007, 11(4): 16 - 25
- [8] Zhang Yaoxue, Zhou Yuezhi. Transparent Computing: A New Paradigm for Pervasive Computing [A]. 3rd International Conference on Ubiquitous Intelligence and Computing (UIC06) [C]. LNCS 4159, 2006, 9: 1 - 11
- [9] Zimmer V, Rothman M, Hale R. Beyond BIOS: Implementing the Unified Extensible Firmware Interface with Intel's Framework [M]. Santa Clara: Intel Press, 2006
- [10] Zhang Yaoxue, Zhou Yuezhi. 4VP<sup>+</sup>: A Novel Meta OS Approach for Streaming Programs in Ubiquitous Computing [A]. IEEE 21<sup>st</sup> International Conference on Advanced Information Networking and Applications (AINA 2007) [C]. Niagara Falls, Canada, 2007, 5(21 - 23): 394 - 403

## From resource sharing to service sharing: opportunities and challenges of transparent computing

Zhang Yaoxue, Zhou Yuezhi

(Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Beijing 100084, China)

[**Abstract**] With the rapid development of computer and network technologies, the computing paradigm has evolved from the original mainframe computing to desktop (PC) computing, and then to network computing, and even pervasive computing. However, the existing computing paradigms all follow the thinking model of single machine or joint-based single machine and thus are characterized as resource sharing. In the new era of network or pervasive computing, users are mainly concerned about services and service sharing, and accordingly the computing will also shift from being characterized as resource sharing to service sharing. This paper proposes a new computing paradigm, transparent computing, which aims for service sharing and presents the concept, architecture and key technologies of transparent computing. Moreover, it also gives the prospect of the impact and future development of transparent computing.

[**Key words**] pervasive computing; resource sharing; service sharing; transparent computing