基于并行PC结构的操作系统模型研究*)

The Operating System Model for Parallel PC Architecture

李之棠　黄辉龙　李　非

（华中理工大学计算机学院　武汉430074）

Abstract　Parallel PC is a novel computer architecture, in which the data-exchanging network becomes the kernel of whole system, and the distributed control theme is adopted to realize the autonomy of all the I/O devices. This paper proposes a new operating system model for the parallel PC architecture, which has a multi-kernel structure, and supports QoS (Quality of Service). In this paper, the overall structure and some important problems, such as the concept of abstract device, the communication and control procedures between the CPU-Memory object and the device objects, and the QoS in the operating system, etc, are discussed thoroughly, and some design schemes are proposed correspondingly.

Keywords　Parallel PC, Operating system, Multi-kernel structure, Abstract device, Quality of Service

1. 引言

并行PC是针对未来高端个人机、工作站设计的，面向实时多媒体应用的新型计算机体系结构[1]。传统计算机采用总线结构，以主存为中心，采用集中式控制策略，其缺陷在于总线结构形成通信瓶颈，集中式控制造成控制瓶颈，成为限制计算机性能进一步提高的障碍。并行PC是针对传统计算机结构的缺陷，综合考虑将来的应用需求而提出的一种以高速交换网络为中心、采用分散控制策略的新型体系结构。这种新结构能够消除系统的通信瓶颈和控制瓶颈，使系统性能得到进一步的优化。

并行PC在体系结构上与传统计算机有很大差别，从而必须设计新型的操作系统以适应这种结构。本文针对并行PC的体系结构提出了一种多核心结构，提供QoS服务的操作系统模型，并对其中关键性的问题，如抽象设备的概念、抽象设备对象的通信及控制方式、操作系统的QoS模型等，进行了讨论并提出了相应设计方案。

2. 并行PC系统结构

并行PC的系统结构模型如图1所示。该结构以高速交换网络为中心，计算机内各部件通过数据交换网络进行通信，彼此之间使用协议完成数据交换和设备控制。

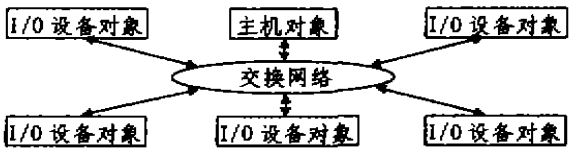

图1　并行PC系统结构模型

在并行PC系统通信事务中，各个通信端对象采用客户/服务器模式(Client/Server)进行交互。

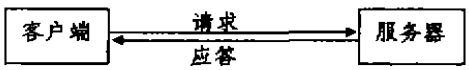

图2　并行PC内部客户/服务器交互模式

在这种交互模式下，大部分设备管理任务转移到该设备所属硬件对象的处理单元上执行，从而减轻了主CPU的负担，使其能更有效地运行操作系统核心和应用程序。此外，由于各设备对象具有功能划分明确、处理隐藏的特点，系统内的通信量将比传统结构减小很多。同时，由于数据交换网的带宽高，支持多路并发通信，因而不存在通信瓶颈，使系统总体性能大为提高。另一方面，这种通信C/S模式，在硬件基础上提供了对现代微内核、Client/Server结构的操作系统模

*)基金项目：国家自然科学基金资助项目(69873016和69972017)。李之棠　博士、教授，研究方向为光互连并行计算机系统结构和计算机网络安全。黄辉龙　硕士研究生，研究方向为计算机系统结构。李　非　博士研究生，研究方向为光互连计算机系统结构。

型[2]的支持。

3. 并行 PC 操作系统模型

并行 PC 与传统计算机在系统结构上的显著差异要求我们根据并行 PC 自身的特点，确定新的操作系统结构模型，以充分发挥并行 PC 的硬件性能。

3.1 并行 PC 操作系统的总体结构

并行 PC 与分布式计算机系统有一定的相似之处，其操作系统设计可以适当借鉴分布式操作系统的设计方法，但它们之间毕竟有很大的区别：分布式系统中的每一个节点都是一个完整的计算机，其上运行的操作系统也都是完备的操作系统，只不过在计算任务上可以进行分配和协作，并向用户提供统一的操作界面，掩盖其多机系统的实质[1]。而在并行 PC 中，交换网络所连接的各个节点虽然都具有处理和存储能力，但从总体功能上看，各个设备节点主要完成所连接设备的管理功能，并向主机应用程序提供相应的服务，系统内的计算任务仍由主机完成。从操作系统设计的角度来看，仅在主机节点上运行一个完整的操作系统，而在各设备节点上运行的管理程序虽然要完成一定的存储器管理、设备控制、协议处理以及进程调度等功能，但它们不是完整的操作系统，只是一个尽量简化了的，面向专用控制的操作系统雏形，可称之为设备操作系统。

针对并行 PC 系统结构的特点，同时也参照目前已有的单机及分布式操作系统设计方案，我们提出了基于并行 PC 的多内核结构的操作系统模型，其结构简图如图3所示。

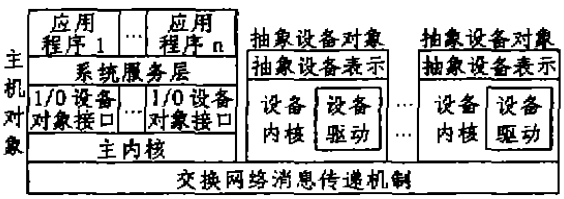

图3 基于并行 PC 的多内核结构的操作系统模型简图

该操作系统采用了多内核结构。在主机对象处是操作系统的主内核，其上依次为设备对象调用接口和系统服务层，最上层是应用程序。主内核需完成以下功能：1)进程调度；2)存储器管理；3)通信接口及协议处理；4)主机资源管理。设备对象接口是各个设备对象在主机对象内的调用接口，通过该接口，主机可以调用相应设备对象的服务，也可以向该设备对象输出操作脚本。设备对象接口在设备对象安装时植入主机操作系统。设备对象接口之上是系统服务层，该层实现传统操作系统所提供的各种系统服务，如内存管理，文件管理等等。系统服务层向用户屏蔽了操作系统内部分布式多核心的特点，用户可见的只是主机对象及其提供的系统服务，所有设备对象的控制、交互与协作均由操作系统内部完成，这种结构可以使并行 PC 在应用接口上与传统的单机操作系统，如 UNIX，WINDOWS，LINUX 等保持兼容。

设备对象是与通讯端对象相对应的软件实体，它具有很高的自治性，拥有独立的设备操作系统，运行在局部处理器和存储器之上。设备操作系统是针对设备而设计的一个精简、高效的管理系统，在功能上类似于传统操作系统中的设备驱动程序，但提供的功能远比传统的设备驱动程序强大。它支持多用户并发访问、QoS 保证、执行操作脚本等多种新的特征。

设备对象的实现并不是任意的，并行 PC 的操作系统模型按设备类别对设备对象的功能及接口加以规范，从而引入抽象设备概念，下一节将具体阐述该思想。

3.2 抽象设备

在并行 PC 中，抽象设备是一组标准，它对设备对象的功能及接口进行抽象，并标准化。并行 PC 中的设备对象被称为抽象设备对象，是符合抽象设备标准的实例。抽象设备采用面向对象的方法对计算机系统的外围设备控制软件进行封装，成为一个自治、独立的对象，并对外提供标准化操作接口。由于有独立的 CPU 和存储器作为抽象设备的硬件支持，与传统的设备驱动软件相比，它的功能更强，效率更高，灵活性更好。

此外，抽象设备并不等同于 I2O 设备（智能 I/O 设备），I2O 设备的主导思想也是让 I/O 设备拥有一定的智能，并制定交互接口，但其调用是设备控制级的，而非系统服务级的。I2O 设备采用存储器映射方式进行通信，而且其功能远没有抽象设备复杂和灵活。

并行 PC 中采用抽象设备的优点主要表现在：

1)设备的控制程序被封装为独立的实体对象，控制接口简化，从而使整个系统的结构更加规整，便于设计和更改。

2)抽象设备使外围设备控制接口标准化，能够很好解决兼容性问题。在传统的系统中，不同厂家外围设备的控制接口都不一样，而且由于多种操作系统的存在，需要为每一个单独设计驱动程序，使设计任务繁重复杂。并行 PC 的抽象设备则根据功能，将计算机的外围设备分为若干基本类别。每一个基本类都规定其基本调用接口。操作系统依据该调用接口操纵设备对象，不关心对象内部的实现和具体类型。这样，一方面使操作系统的操作简化，另一方面设备对象的设计也简化了，只需针对标准协议接口设计其相应软件实现。

3)并行 PC 的抽象设备对象支持多个并发服务。在传统计算机系统中,外围设备都看成是独占式的,只能响应一个用户请求。而并行 PC 的抽象设备则可以同时响应多个用户请求、对应每一个请求生成其服务实例,并发执行相应的 I/O 操作,从而改善每个进程的 I/O 响应时间。

4)抽象设备支持继承和派生,从而使软件易于升级并保持向下兼容,使抽象设备具有很好的可扩展性。

目前,在并行 PC 的设计中,主要考虑以下几类抽象设备:①磁盘抽象对象类。包括硬盘、软盘、CDROM、磁盘阵列等随机块存储设备。②标准终端输入设备类。包括键盘、鼠标等标准输入设备。③网络抽象设备。包括网络适配卡,不仅完成网络设备驱动功能,而且将通信协议栈直至网络层的部分置于该对象上实现,从而进一步减轻主 CPU 负担。④音频处理抽象设备。包含传统的声卡,完成音频数据流的处理,包括音频压缩/解压、MIDI 处理,音频播放等功能。⑤视频处理抽象设备。其基本功能应完成显示设备的输出控制,此外,可选择的扩展功能还包括各种较复杂的图像、图形处理功能,如 JPEG、MPEG 压缩/解压缩,二维、三维图形计算,等等。

3.3 主机操作系统与抽象设备对象的交互方式

在并行 PC 中,主机操作系统与各个设备对象之间采用消息传递方式进行通信。应用程序向操作系统提出的有关 I/O 的系统调用(如读/写文件),系统服务层经过处理、根据设备对象提供的操作接口,形成相应的操作命令及数据,提交主机操作系统内核,内核对这些命令及数据加以封装、通过简单的通信协议处理,发送到系统内数据交换网上、最后由相应的设备对象接收。在设备对象处,设备操作系统内核接收到该数据包后,首先执行通信协议处理,解封装后获取发来的命令和数据,然后将这些命令和数据发向抽象设备服务器,由其调用相应的设备驱动程序完成相应的操作。

主机操作系统与设备对象的交互方式采用客户机/服务器模式,主要有两种访问控制方法,一种是基于抽象设备命令接口的远端过程调用方式,另一种是远端脚本执行方式。下面对这两种方法分别加以说明。

1)基于抽象设备的远端过程调用方式　远端过程调用(RPC)最初由施乐公司的 Birrel 等人提出,后由 Sun 微系统公司制订为一项互连网标准[5]。RPC 目前已广泛应用于互连网络、分布式系统等。许多典型的分布式操作系统,如 Amoeba、Mach、Chorus、DCE 等[3],均采用 RPC 作为分布式程序的通信机制。

在并行 PC 中,由于各个抽象设备对象已向主机对象公布了其控制接口,其中包括可供调用的方法声明,因此,采用远端并行 PC 中的远端过程调用实现对抽象设备对象的控制是一种十分自然的方式。并行 PC 中的 RPC 调用过程如下图所示:

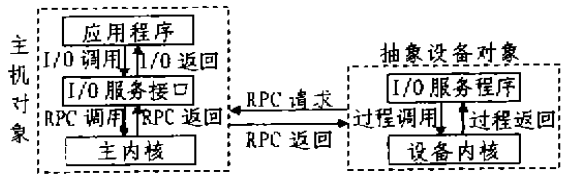

图4　并行 PC 内部 RPC 调用过程

主机对象中 I/O 设备控制部分接收到应用程序的 I/O 调用命令后,将命令和参数等封装为一个 RPC 包,并发向抽象设备对象。在抽象设备对象一方,抽象设备服务器根据该命令及参数调用相应的过程或函数,并返回结果。

2)远端脚本执行方式

在并行 PC 中,除了以命令原语的方式执行对抽象设备控制外,还可以采用生成远端执行脚本的方式完成对设备对象的操纵。设计这种远端脚本执行方式的目的在于使面向 I/O 的计算尽可能在智能 I/O 设备上执行,从而减少不必要的主机处理,降低系统内的通信量。这种使 I/O 计算尽可能靠近 I/O 设备的思想在主动磁盘(Active Disk)[6]的设计中已被提出,但是在主动磁盘的设计中,采用的是下载磁盘小程序(Disklet)并执行的方式。考虑到在并行 PC 中,主机系统和抽象设备是异种环境,难以完成针对设备对象的可执行程序编译,因此,本文提出远程脚本执行方式。并行 PC 抽象设备的脚本语言针对每一类抽象设备分别设计,语言规范简单,只包括简单数据类型和数据结构,支持基本的程序结构,以及面向设备的操纵命令。

与主动磁盘的 Disklet 相比较,远程执行脚本虽然有功能较弱,执行效率较低等缺点,但它保证了抽象设备对象内部结构对主机操作系统的透明性,以及抽象设备对象接口的规范性。此外,与主动磁盘的 Disklet 相比较,远程脚本执行方式还可大大简化操作系统的设计。

并行 PC 中采用远程执行脚本实现 I/O 处理的基本过程如下图所示。

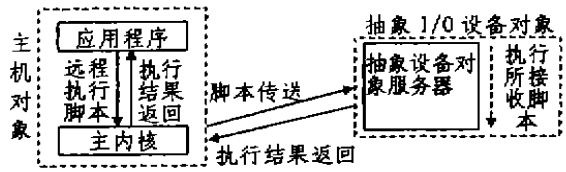

图5　并行 PC 中的远程脚本执行方式

3.4 并行 PC 操作系统中的 QoS 模型

与传统计算机相比，并行 PC 更着重于对实时多媒体应用的支持，并且其硬件结构也非常适合并发实时多媒体应用。为了能充分利用其硬件结构的高性能，还需要在软件结构上，尤其是在操作系统的设计中加入对实时多媒体应用的支持，其中最主要的是实现对应用程序提供服务质量 QoS 保证。

操作系统中的 QoS 是指操作系统在对用户所提供的各项服务中，包括处理机时间分配、内存分配、I/O 操作等，提供相应的服务质量保证，从而确保各种实时、多媒体应用程序在运行时可及时获得所需的资源。在支持 QoS 的操作系统中，用户在申请系统服务时，同时给出其 QoS 要求，操作系统根据系统内资源使用情况与用户进行协商，最终形成 QoS 协定，并根据该协定为用户保留相关资源。

国际上对操作系统中的 QoS 服务问题的研究始于九十年代初，并已提出过多种 QoS 模型。传统操作系统 QoS 模型基于已有的操作系统改造而成，一般采用下图所示结构[7]。

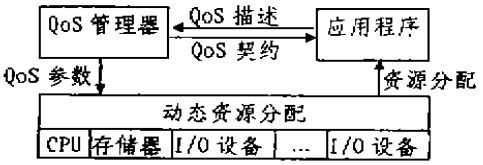

图6　传统操作系统中的 QoS 模型

该模型的核心部分是 QoS 管理器和动态资源管理器。QoS 管理器接收从应用程序发来的 QoS 描述，并将其转换为相应的 QoS 参数，同时确定系统是否能够满足该 QoS 要求。如果不能满足，则向应用程序发否定消息，如果能够满足，则记录该 QoS 并与应用程序达成一个契约(contract)，同时将相应的 QoS 参数传给动态资源管理器，由其完成具体的资源预留与分配[7]。

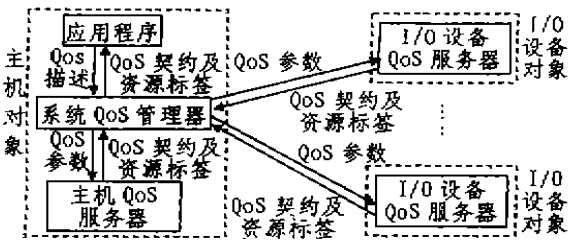

图7　并行 PC 操作系统的 QoS 模型

上述模型有两个基本假定，一是主机上的 QoS 管理器能够了解并记录系统所有资源的使用情况；二是所有的资源，包括各个 I/O 设备均由主机系统统一管理。然而在并行 PC 计算机系统中，由于采用了"抽象设备"的策略，各个 I/O 设备已经形成了自治的设备对象，因而主机系统并不能直接完成 I/O 设备资源的管理，显然针对这些资源的 QoS 服务并不能在主机处实现，因而并行 PC 不能使用传统 QoS 模式。针对并行 PC 的系统结构，本文提出一种基于代理的操作系统 QoS 模型(图7)。

在这种模型中，系统 QoS 管理器实际上是一个 QoS 代理。与传统模型中的 QoS 管理器一样，它从应用程序接收 QoS 描述，但并不进行决策及资源分配等处理，而是将该 QoS 描述分解后转发给相应的设备对象。在各个设备对象处，存在各自的 QoS 服务器，处理相应的 QoS 请求。要指出的是，由于各个设备对象的 QoS 管理及资源分配都十分简单，而且是专用的，因而其 QoS 管理器和动态资源管理器可合为一体，即 QoS 服务器。各 QoS 服务器完成 QoS 处理后，将结果(QoS 契约及所分配的资源标签)返回系统 QoS 管理器，由其汇总后进一步返回应用程序。

各个抽象设备对象的 QoS 接口事实上也是该抽象设备控制接口的一部分，并应当与抽象设备的其它控制接口一起实现标准化，其 QoS 服务器实际上也可嵌入抽象设备服务器中实现。

总结　本文针对并行 PC 的系统结构提出了一种新型的操作系统模型。这种操作系统采用多核心结构，其设备管理子系统采用了面向对象方法进行封装，成为自治的抽象设备对象；主机与设备间通过 RPC、远端执行脚本等方式进行通信和控制。此外，还加入了 QoS 服务特征以增强对实时应用的支持。

该操作系统模型针对并行 PC 的 I/O 设备智能化，内部数据交换网络化的特点设计，可以充分发挥并行 PC 的硬件性能，较好地满足各种智能化、实时多媒体应用的需求。

参 考 文 献

1　Li Zhitang, Huang Huilong. The Architecture Design of Parallel PC for Switching Network Centric. In: Proc. of 3rd Workshop on Advanced Parallel Processing Technologies. Changsha, China, 1999

2　Tanenbaum A S, Woodhull A S. Operating Systems: Design and Implementation. 2nd Ed. Prentice Hall Inc, 1997

3　Tanenbaum A S. Distributed Operating Systems. Prentice Hall Inc, 1995

4　Thompson T. I2O Beats I/O Bottlenecks. BYTE, 1997, 22(8): 85～89

5　RFC 1050, RPC: Remote Procedure Call Protocol Specification Version 2, Sun Microsystems, Inc, June 1988

6　Acharya A, Uysal M, Saltz J. Active Disks: [Technical Report TRCS98-06]. University of California, Santa Barbara, Department of Computer Science

7　Hyden E A. Operating System Support for Quality of Service: [Ph. D thesis] University of Cambridge, Computer Laboratory, March 1994