多核处理器——技术、趋势和挑战

彭晓明[1] 郭浩然[2] 庞建民[2]
(中国人民解放军空军雷达学院监视情报系 武汉 430010)[1]
(中国人民解放军信息工程大学计算机科学与技术系 郑州 450002)[2]

摘 要 多核处理器已经成为当前微处理器技术发展的重要方向。介绍了多核处理器的起源和发展现状,分析了多核处理器技术的发展趋势。重点讨论了多核处理器技术涉及的片上网络、存储结构设计、编程接口以及资源管理等关键技术;在此基础上,进一步探讨了多核处理器的发展所面临的主要挑战。

关键词 多核处理器,体系结构,片上网络,存储结构
中图法分类号 TP391 **文献标识码** A

Multi-core Processor——Technology, Tendency and Challenge

PENG Xiao-ming[1] GUO Hao-ran[2] PANG Jian-min[2]
(Department of Surveillance and Intelligence, PLA AirForce Rador College, Wuhan 430010, China)[1]
(Department of Computer Science and Technology, PLA Information and Engineering University, Zhengzhou 450002, China)[2]

Abstract Multi-core processor has become important developing direction to current micro-processor technology. The paper first introduced the origin and development status of multi-core processor, and analyed its developing tendency. Key techniques of multi-core processor including NoC, storage architecture design, programming interface and resource management are presented. And then, the main challenges to multi-core processor are discussed.

Keywords Multi-Core processor, Architecture, Network on chip, Storage structure

芯片设计、器件制造和新材料加工技术的进步使得大规模集成电路的制程工艺得到飞速发展,单个硅晶片上集成的晶体管数目遵照摩尔定律的力量持续快速增长,这使得在单个芯片内集成多个物理计算内核成为可能,这种处理器称为多核处理器。

在多核处理器出现之前,商业化处理器一直致力于单核处理器的发展,其性能已经发挥到极致,仅仅提高单核芯片的速度会产生过多热量且无法带来相应的性能改善,但应用对 CPU 资源的需求却远远超过 CPU 的发展速度,单核越来越难以满足要求,其局限性也日渐明显:以英特尔发布的采用 NetBurst 架构的奔腾 4 CPU 为例,它包括 Willamette、Northwood 和 Prescott 等 3 种采用不同核心的产品[1]。利用冗长的运算流水线,即增加每个时钟周期同时执行的运算个数,就可达到较高的主频。这 3 种处理器的最高频率分别达到了 2.0GHz、3.4GHz 和 3.8GHz。但由于流水线过长,使得单位频率效能低下,加上由于缓存的增加和漏电流控制不力造成功耗大幅度增加,3.6GHz 奔腾四芯片在性能上反而还不如早些时推出的 3.4GHz 产品。所以,Prescott 产品系列只达到 3.8GHz 就不再继续。此外,随着功率增大,散热问题也越来越成为一个无法逾越的障碍。据测算,主频每增加 1GHz,功耗将上升 25 瓦,而在芯片功耗超过 150 瓦后,现有的风冷散热系统将无法满足散热的需要。此外,单核心处理器也无法满足程序并行化的需求。

单核心处理器的局限性、编译技术、并行技术和体系结构的发展以及芯片内庞大的晶体管数量(英特尔的奔腾(Pentium)4 至尊版 840 处理器晶体管数量已经增加至 2.5 亿个),这些因素共同促使人们探寻提高处理器性能的新途径。如何有效利用这些不断增长的、数以亿计的晶体管资源,如何构建新的芯片体系结构来推动微处理器芯片性能的大幅、持续提升,成为信息技术领域(工业界与学术界)的研究热点。多核处理器技术的诞生为人们提供了新的研究方向和思路,各种多核处理器设计理念、方案及产品纷纷登台,多核处理器涉及的关键技术日益成熟和完善,应用领域也与日俱增。

本文试图从多核处理器技术的诞生、发展趋势、关键技术以及所面临的挑战等方面对该领域的研究进行综述。第 1 节从首款多核处理器的诞生引入对多核处理器发展现状的总结;第 2 节从应用的角度对多核处理器的发展趋势进行探讨;第 3 节则从学术和工程的角度入手,对多核处理器涉及的关键技术进行归纳;最后对多核处理器技术面临的挑战进行探讨。

1 多核处理器技术的发展现状

1996 年,斯坦福大学研制出世界上第一款多核处理器的

本文受国家高技术研究发展计划(863)(2009AA012201)资助。
彭晓明(1965-),男,硕士,副教授,硕士生导师,主要研究方向为计算机软件与理论、高性能计算;郭浩然(1987-),男,博士生,主要研究方向为高性能计算、虚拟现实,E-mail:haoran006@yahoo.com.cn;庞建民(1964-),男,博士,教授,博士生导师,主要研究方向为并行处理、高性能计算、机器学习。

原型系统 Hydra[2]。此后，多核处理器的相关研究与商业应用开始蓬勃发展。短短 10 年时间，双核(Daul-Core)处理器芯片成为通用微处理器市场的主流产品，2005 年 4 月，英特尔推出简单封装双核的奔腾 D 和奔腾 4 至尊版 840 处理器；AMD 在之后也发布了双核皓龙(Opteron)和速龙(Athlon) 64 X2 和处理器。但真正的“双核元年”则被认为是 2006 年 7 月 23 日，英特尔基于酷睿(Core)架构发布的处理器。同年 11 月，英特尔又推出面向服务器、工作站和高端个人电脑的至强(Xeon)5300 和酷睿双核、四核的至尊版系列处理器。与上一代台式机处理器相比，酷睿 2 双核处理器在性能方面提高 40%，功耗反而降低 40%；AMD 也推出了双核、三核以及四核的“皓龙”芯片；IBM 的 Power6 芯片也是双核设计。而不少专用微处理器甚至出现了上百核心的产品，双核、多核(Multi-Core)开始向众核(Many-Core)的方向发展。Sony 与 IBM 等公司合作研发的针对图形运算的 CELL[3] 芯片拥有 9 个核；SUN 公司的 Ultrasparc T1 有 8 个核；Clearspeed 的 CX600 达到 96 个核；CISCO 的 NPU 有 192 个核；日本 RIKEN 针对 N-body 计算的 Grape-DR 处理器达到 512 个核。图 1 为 Hydra 多核处理器，图 2 为 IBM Cell 多核处理器结构。

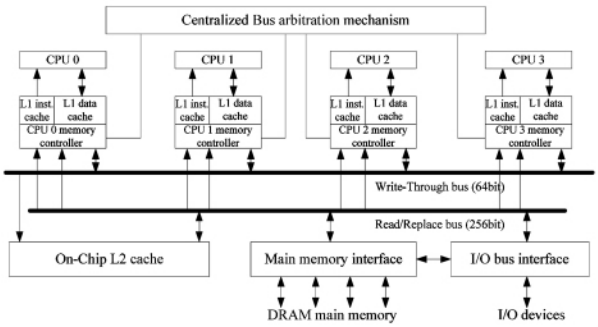

图 1　Hydra 多核处理器

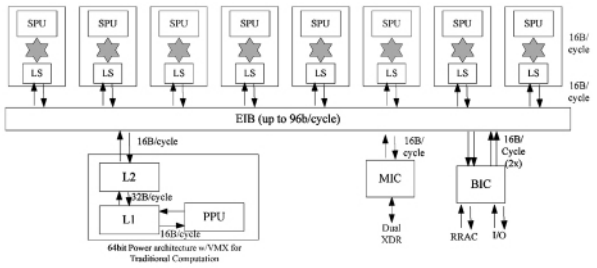

图 2　IBM Cell 多核处理器结构

多核处理器技术的研究在国内起步较晚，但发展势头迅猛。2008 年 Hot Chips 大会上，中国研究人员就展出了龙芯 3[4] 的核心架构。由中科院主导，我国已经成功生产出 65nm 制程的多核“龙芯 3”处理器，有四核和八核两个版本，时钟主频均为 1GHz。“龙芯 3”的特色在于采用分布式、可扩展的架构，使用了 MIPS64 内核，增加了超过 200 条的 X86 二进制转换和多媒体加速指令。清华大学信息科学技术国家实验室 CPU 中心汪东升教授和他的团队目前也致力于多核处理器的研究[5]，在 2009 年中国计算机大会上，他们从产业界和学术界的角度出发，针对多核系统设计中的关键问题、面向多核环境的操作系统、并行编程、并行编译、可重构等方面展开讨论，系统阐述了我国多核处理器技术的研究思路、发展方向及面临的挑战。龙芯 3 多核处理器结构如图 3 所示。

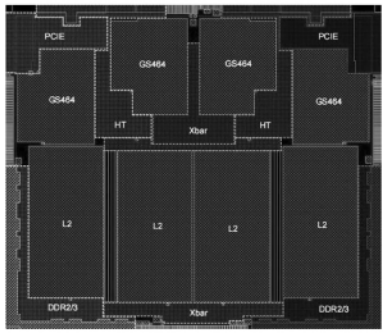

图 3　龙芯 3 多核处理器结构

2　多核处理器技术的发展趋势

2.1　多核处理器发展趋势的推动因素

多核处理器是 CPU 发展的必然趋势，是技术发展和应用需求的必然产物。多核技术使我们能通过不断地增加计算核心的数量来继续提升处理器的计算能力，从而延续摩尔定律；而根据 Pollack's Rule[6]，大量的简单核心将比少量的复杂核心带来更大的系统计算吞吐量。

一般观点认为，多核处理器技术出现的必然性基于以下 5 个事实。

1. 门延迟逐渐缩短，而全局连线延迟却不断加长

随着 VLSI 工艺技术的发展，晶体管特征尺寸不断缩小，使得晶体管门延迟不断减少，但互连线延迟却不断变大。当芯片的制造工艺达到 0.18 微米甚至更小时，线延迟已经超过门延迟，成为限制电路性能提高的主要因素。在这种情况下，CMP(单芯片多处理器)由于分布式结构中全局信号较少，与集中式结构的超标量处理器结构相比，在克服线延迟影响方面更具优势。

2. Pollack 规则的指示

按照 Pollack 规则，我们知道处理器性能的提升与其复杂性的平方根成正比。如果一个处理器的硬件逻辑提高一倍，至多能提高 40%的性能，而如果采用两个简单的处理器构成一个相同硬件规模的双核处理器，则可以获得 70%～80%的性能提升，同时在面积上也同比缩小。

3. 绿色计算(Green Computing)的推动

近年来，低能耗的绿色计算[7]理念在工业界和学术界得到很大的关注。而随着工艺技术的发展和芯片复杂性的增加，芯片的发热现象日益突出。多核处理器里单个核的速度较慢，处理器消耗较少的能量，产生较少的热量。同时，原来单核处理器里增加的晶体管可用于增加多核处理器的核。在满足性能要求的基础上，多核处理器通过关闭(或降频)一些处理器等低功耗技术，可以有效地降低能耗。

4. 芯片低设计成本的考量

处理器结构复杂性的不断提高以及人力资源成本的不断攀升，使得芯片设计的成本随时间呈线性甚至超线性的增长。多核处理器通过处理器 IP(Intellectual Property)等的复用，可以极大降低设计的成本，同时模块的验证成本也显著下降。这符合芯片低设计成本的理念。

5. 体系结构发展的必然

超标量(Superscalar)结构和超长指令字(VLIW)结构在当前的单核高性能微处理器中被广泛采用。但是它们的发展都遇到了难以逾越的障碍。Superscalar 结构使用多个功能部

件同时执行多条指令，实现指令级的并行（Instruction-Level Parallelism，ILP）。但其控制逻辑复杂，实现困难，研究表明，Superscalar 结构的 ILP 一般不超过 8。VLIW 结构使用多个相同功能部件执行一条超长的指令，但也有两大问题：编译技术支持和二进制兼容问题。

主流应用需要处理器具备同时执行更多条指令的能力，但是从单一线程中已经不太可能提取更多的并行性。为此，研究人员提出了两种新型体系结构：单芯片多处理器（CMP）与同时多线程处理器（Simultaneous Multithreading，SMT），这两种体系结构可以充分利用这些应用的指令级并行性和线程级并行性，从而显著提高了这些应用的性能。从体系结构的角度看，CMP 相对 SMT 的最大优势在于其模块化设计的简洁性，复制简单设计非常容易，指令调度也更加简单；更短的芯片连线使 CMP 比长导线集中式设计的 SMT 更容易提高芯片的运行频率，从而在一定程度上起到性能优化的效果。由于 CMP 采用了相对简单的微处理器作为处理器核心，使得 CMP 具有高主频、设计和验证周期短、控制逻辑简单、扩展性好、易于实现、功耗低、通信延迟低等优点。此外，CMP 还能充分利用不同应用的指令级并行和线程级并行，具有较高线程级并行性的应用如商业应用等可以很好地利用这种结构来提高性能。目前，单芯片多处理器（多核处理器）已经成为处理器体系结构发展的一个重要趋势。

2.2 多核处理器技术的应用展望

无论是移动与嵌入式应用、桌面应用还是服务器应用，都将采用多核的架构。

对于多任务处理、海量数据分析和高速网络数据流处理等计算密集型的高端应用而言，多核处理器的优势是显而易见的。与目前主流的双核平台相比，基于多核处理器的平台提供更多内存和 I/O，大大减少了通信瓶颈并优化了性能；内存和 I/O 共同为每一个处理器提供相关数据，从而使所有内核能够以最高速度执行计算负载。

虚拟化技术的应用和分布式计算的发展使得系统能够将应用整合到数量较少但性能更加强大的服务器上，可以控制数据中心的能源开销。运行大型数据库服务的平台，对于数据访问、数据调用和数据更改等操作的并发控制及优化有很高的要求，多核处理器的采用是个不错的助力；运行 Web 网站的服务器，大流量的数据也带来了处理性能的高要求，多核会更具优势。在工业工程技术领域、国防领域，诸如石油勘探、核爆模拟、材料合成、DNA 测序以及气象、电力系统调度等应用需要复杂的科学计算和大型图形建模，这些任务对多线程、多任务、海量数据的计算处理能力提出很高的要求，采用多核处理器的平台则能够为其提供相对理想的解决方案。

在可以预见的未来里，技术和应用的双重推动，必定使得多核处理器得到推广和普及，围绕多核处理器的产业和学术成果会得到蓬勃发展，进而引发计算机、通信、多媒体等领域的一场变革。

3 多核处理器关键技术

从结构上看，多核处理器与传统对称多处理器系统存在许多相似之处，但是仍有一些本质差别。首先，多核处理器所集成的片上网络与传统多处理器系统中互连网络的对应关键参数，如网络延迟和带宽等存在数量级上的差异，因此片上网络成为多核处理器的关键技术之一；其次，单芯片中多个处理器内核对片外内存的高速并发访问，使本来就已经成为瓶颈的内存系统面临更大的压力，如何降低或消除存储墙效应，为多个处理器内核连续提供高带宽低延迟的数据访问接口成为一个关键问题；再次，在用户层面来看，为用户提供简单易用的并行编程模型、支持系统软件来高效管理系统中的处理器内核资源也构成多核处理器研究的关键技术。本节将从多核处理器的内部结构（片上网络和存储结构）和外部软件接口（编程模型和系统软件）两大方面介绍多核处理器的关键技术。

3.1 片上网络技术

多核系统中，内核数量的增加，使得基于传统共享总线的互连结构受到扩展能力的制约，无法满足处理器内核间的高速通信需求。片上网络正成为多核处理器核间互连的必要器件。相关研究包括高效路由算法、交换技术和拓扑结构、QoS 及片上网络的功耗、资源占用的优化等。片上网络路由器主要实现了虚通道仲裁、包路由和交换等功能。图 4 是片上网络路由器的典型结构。

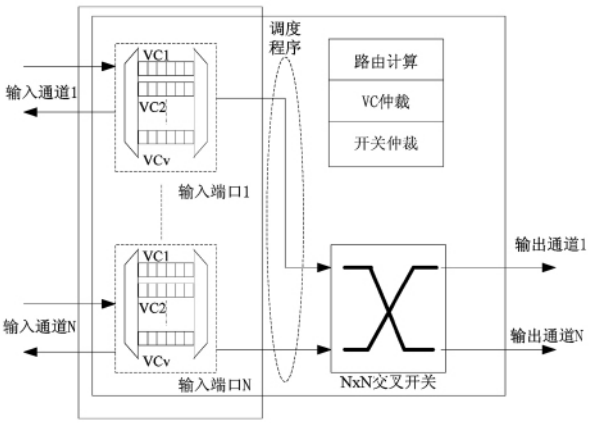

图 4 片上网络路由器的典型结构

片上网络路由算法与片外通信网络的算法不同，主要表现在两个方面：第一，片上网络具有较低的互连延迟，并且运行在较高的时钟频率下，因此片上网络很难采用复杂耗时的通用片外网络路由算法，一般多采用简单的维序路由算法；第二，片上网络通常采用包交换技术，也有一些研究开始尝试使用电路交换方法。与包交换相关的有动态调整虚拟通道数[8]、虚拟通道快递[9]和分层交换[10]等技术。

在网络拓扑结构研究方面，研究人员提出了集中式网络、层次化星型和扁平化蝴蝶等新型拓扑结构。研究还发现通过在原有拓扑结构中增加少量长距离连线，可以显著提高网络性能，且不会带来明显的功耗。IBM 发布了“穿透硅通道（Through-silicon Vias）三维芯片堆叠”技术，该技术可以大幅缩短消息传输距离。此后，关于三维互连网络拓扑结构的研究迅速发展起来。

QoS 方面的研究主要包括：基于公平的原则为不同的消息流预留虚拟通道，为不同消息分配不同的优先级，采用基于 QoS 的拥塞控制算法避免网络负载饱和等方面。片上网络功耗、资源占用的优化方面的技术包括动态电压调节、全局异步局部同步以及任务调度等。

3.2 存储层次结构设计

计算机主存访问速度远远落后于处理器性能增长的速

度。处理器和存储器之间不断扩大的速度差异导致了阻碍整个系统性能提升的“存储墙”。与单核心处理器相比，多核处理器的内存访问请求数目随着并行执行线程数目而线性增加，加剧了内存访问压力。为了解决该问题，存储层次结构设计通常需要考虑3个方面：(1)通过多级片内缓存尽可能减少处理器内核对片外主存的访问；(2)减少缓存访问延迟和缓存缺失时的响应延迟；(3)尽可能提高内存访问带宽，减少内存访问延迟；(4)使用芯片间光互连技术来替代当前的铜互连技术。

3.2.1 减少对片外主存的访问次数

通过片内缓存减少对片外内存访问的有效办法是设法提高片内缓存的命中率。片内缓存命中率与缓存组织方式和替换策略等有密切关系。缓存组织的管理包括片内缓存级数、单位缓存大小、私有或共享方式、缓存组相连度以及块大小等配置。替换策略决定哪些数据被替换出cache，cache数据的变动会影响后续数据访问的命中率。

多核处理器采用私有缓存时，缓存访问延迟较低，但是每个处理器内核可使用的缓存容量较小；采用共享缓存时，缓存访问延迟较长，但是每个处理器内核可使用全部共享缓存空间。可供每个处理器内核使用的缓存容量在某种程度上决定了缓存命中率，无论是私有缓存还是共享缓存，都无法保证既有较高的缓存命中率，又有较短的缓存访问延迟。目前，有些研究尝试在私有缓存和公共缓存中间寻找一个平衡点，旨在获得接近私有缓存的访问延迟和共享缓存的容量。

此外，许多降低核处理器片外内存访问次数方面的研究成果，比如增加读写缓冲区、写数据合并等，也都适用于多核处理器。

3.2.2 减少缓存访问延迟

减少缓存访问延迟需要考虑多核处理器缓存的内部组织结构。通常，多核处理器在片上集成两级或者三级缓存，其中最后一级缓存一般采用共享方式，其前一级或者前两级缓存采用私有方式。私有缓存容量较小，与处理器内核直接相连，访问延迟较短；而共享缓存容量较大，访问延迟较长，因此，减少缓存访问延迟主要是减少最后一级共享缓存的访问延迟。

早期的片上缓存设计采用单一均匀缓存访问延迟结构(Uniform Cache Architecture，UCA)，每个处理器内核访问所有缓存模块的延迟都一样。随着线延迟的不断增大，片上共享缓存的结构从均匀缓存访问延迟结构转变为非均匀缓存访问延迟结构(Non Uniform Cache Architecture，NUCA)[11]，处理器内核到不同缓存模块的访问延迟不再相等，而是与处理器内核到缓存模块的线长相关。

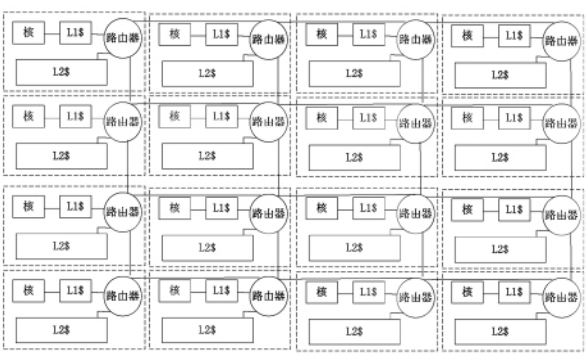

图5 Tile处理器结构示意图

目前，多核处理器设计中常采用Tile结构即一种非均匀缓存访问的延迟结构。如图5所示，处理器内核、一级缓存、二级缓存块和路由器等组成一个Tile，多个Tile通过片上网络连接成为一个多核处理器。Tile处理器的二级缓存在逻辑上共享，但物理分布到各个Tile节点中。这种凡是访问临近缓存的延迟都较短，而访问远程缓存的延迟仍然较长。

为减少非均匀缓存访问延迟结构中缓存的平均访问延迟，提高多核处理器性能，研究人员提出了一些优化技术，如D-NUCA(动态非均匀缓存访问延迟结构)、NuRAPID(Non-uniform access with Replacement and Placement Using Distance associativity)[12]、受损复原(Victim Replication)[13]、ASR[14]和R-NUCA(Reactive NUCA)[15]等。这些优化技术试图将频繁访问的数据复制或迁移到临近处理器内核的缓存块中，从而减少对这些数据的后续访问延迟。当数据被多个处理器内核连续访问时，从临近节点中获取数据的速度往往要比从宿主节点读取快。数据复制与迁移技术最初是为了优化处理器对数据的连续访问，未考虑到上述情况。基于树的目录协议[16]是对处理器内核节点进行分区管理，为每个子区域建立目录，使大多数缓存访问请求在子区域内得到响应，从而有效减少缓存访问平均访问延迟。传输中优化方法[17]是在请求消息发送到宿主节点的传输过程中，判断每个途径节点是否包含数据副本。如果节点存在副本，则读请求可直接从该节点获得响应，而将写请求直接置成无效或更新该节点。

缓存缺失时的响应延迟主要由采用的缓存一致性协议决定。多核处理器通常采用基于目录的协议，其运作方式为：首先，发起请求的处理器内核将请求发送到全局目录；然后，根据目录内容将请求转发到包含该数据有效副本的处理器内核；最后，拥有有效副本的处理器内核将数据返回到请求节点，或者使其私有副本无效。这种所谓“三次跳跃”的缓存缺失请求响应过程的延迟较长。目地集预测技术是通过预测目的节点集合(而非目录查找)来减少目录协议响应延迟。令牌协议则以广播方式发送消息，避免了目录查找过程。但是广播方式增加了网络传输的消息量。

3.2.3 提高内存访问带宽和速度

当前，提高内存带宽、降低内存访问延迟的一个手段是在处理器片内集成1个或多个内存控制器。与系统控制芯片内部的内存控制器相比，处理器内置的内存控制器和处理器内核之间的通信无需通过处理器外部总线，因而具有低延迟、稳信号及低功耗等优点。全缓冲内存模组是新一代的内存技术，其内存模块与内存控制器间采用串行接口多路级联的设计，以串行方式进行数据传输。在这种新型架构中，每个内存插槽上的缓冲区互相串联，插槽间采用点对点的连接方式，数据会在经过第一个缓冲区后传向下一个缓冲区，使得第一个缓冲区和内存控制器之间的连接阻抗能够始终保持稳定，有助于容量的增大与频率的提高。另外，全缓冲内存模组中内存模块的I/O引脚数量从DDR2模组的240Pin减少到96Pin，更适于内置在处理器芯片中。

此外，三维堆叠存储互连技术[18]为提高存储访问接口的性能提供了一条新思路，在处理器内核与封装之间安放一个薄硅片，处理器内核与内存模块通过硅片中的孔洞进行通信，这种技术可以显著提高处理器与内存间的传输带宽，同时缩短传输距离，在有效提高内存系统性能的同时，也降低了总体

功耗。

3.3 并行编程模型设计

与对称多处理机系统 SMP 类似，大多数多核处理器采用了共享存储体系结构，编程模型为共享内存模型。OpenMP 是主流的共享内存模型，几乎得到了所有商业编译器的支持，具有很好的可移植性。基于 OpenMP 编写的程序可直接移植到多核处理器上运行，而那些为单核处理器编写的串行程序则需要并行化处理之后才能在多核处理器的系统上高效运行。目前，主流的微处理器基本上都是多核处理器，应用程序从串行编程向并行编程转向所面临的最大挑战是如何为程序开发者提供简单、易用的并行编程语言和环境。

OpenMP 或线程库并行编程环境适于编写各种并行应用程序，可是若要设计和调试出正确的并行程序还是相对困难的。至于提高程序性能方面，更要求程序员能深入了解应用领域的相关背景知识，因此这些传统编程模型尚无法满足快速发展的并行编程要求。

简化并行编程的一个方法是采用自动并行处理，即通过编译器发现程序潜在的并行性，然后将此串行程序并行化。但是多数串行程序中蕴含的并发性较低，自动并行化处理的效果较差。

除了串行程序的自动并行处理外，还有一种比较有效的简化并行编程的方法是为不同领域提供不同的并行编程模型。例如，Google 提出的 MapReduce[19] 并行编程模型适用于大规模数据集的并行批量处理，图 6 是 MapReduce 模型的结构。针对多媒体及网络应用领域，研究人员也提出一些其他的编程模型。这种面向具体应用领域的编程模型会极大地简化并行编程过程。

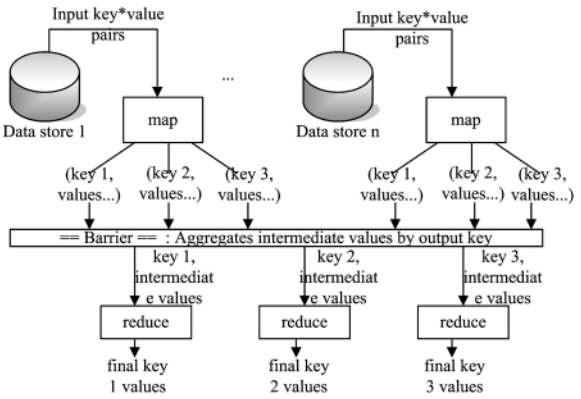

图 6 MapReduce 体系结构

3.4 资源管理技术

使用单个操作系统来管理数量众多的处理器内核并非易事，系统软件如何才能有效管理单个数量众多的处理器资源，是值得深入研究的技术。

系统虚拟化技术[20] 近年来得到了迅速发展，且应用领域极大扩展。人们普遍认为系统虚拟化技术可能是解决管理单一系统中众多处理器内核的一种相对简单而有效的方法。该技术由运行在系统/处理器最高级别权限中的相对精简的虚拟机管理器管理系统中的所有处理器资源，而运行在相对较低级别权限上的虚拟机管理器则负责用户任务调度的多个操作系统结合，由此达到对处理器资源分而治之的目的。

高效的系统虚拟机的实现需要在处理器以及指令级结构上得到有效的支持，目前英特尔多核处理器上的 VT(Virtualization Technology)技术和 AMD 多核处理器中的 Pacifica 技术都直接反映了这种思路。当前虚拟机技术已经被广泛认为是云计算领域中的关键技术，随着多核技术的进一步发展和操作系统的突破性创新，多核处理器可以为系统提供更强的虚拟化支持，为平台的资源管理提供更好的支撑。

4 多核处理器技术面临的挑战

多核处理器虽然拥有芯片集成度高、指令并行度高、器件设计验证方便等诸多优势，而且还继承了传统单处理器研究中的某些成果，例如同时多线程、宽发射指令、降压低功耗技术等，但它毕竟是一种新的结构，在多核结构设计和应用开发中出现了很多新问题，这些问题成为制约多核处理器技术进步的重要因素，给多核处理器的未来发展提出了挑战。

4.1 核心结构的选择难题

目前多核处理器的核心结构主要有同构和异构两种。同构结构采用对称设计，原理简单，硬件上较易实现。当前主流的双核和四核处理器基本上都采用同构结构。同构设计的问题在于：随着核心数量的不断增多，如何保持各个核心的数据一致；如何满足核心的存储访问和 I/O 访问需求；如何选择一个各方面性能均衡、面积较小以及功耗较低的处理器；如何平衡若干处理器的负载和任务协调等。

与同构结构相比，异构的优势是通过组织不同特点的核心来优化处理器内部结构，实现处理器性能的最佳化，而且能有效地降低功耗。但是异构结构也存在着一些难点。首先，搭配哪几种不同的核，核心间任务如何分工以及如何实现；其次，结构是否具有良好的扩展性，还是受到核心数量的限制；再者，处理器指令系统设计和实现也是问题。因为不同核所用的指令系统对系统的实现也是很重要的，那么采用这些不同的核，是采用相同的指令系统还是不同的指令系统，能否运行操作系统等，也是需要考虑的内容。同构多核和异构多核如图 7 所示。

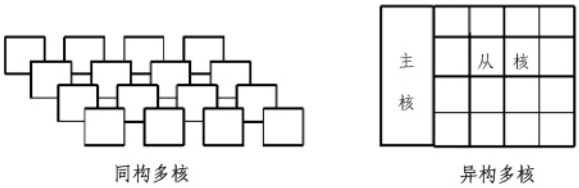

图 7 同构多核和异构多核

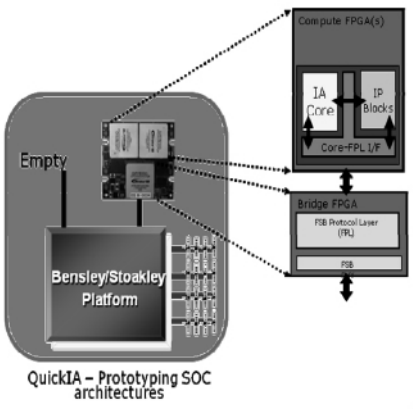

图 8 Quick IA 原型系统

图 8 是英特尔提出的异构多核结构原型系统 QuickIA，用于验证新的处理器体系结构、SoC 原型验证、测试新的指令、异构系统体系结构研究、可重构计算研究以及周期精确的仿真等。QuickIA 是 CPU 核心和 SoC 核心协同工作于单芯片内的异构多核结构。它使用 FPGA 来实现 IA 核心和 IP 模块，如图 8 所示。

4.2 功耗问题

传统单处理器的一个瓶颈就是随着频率的提升，功耗越来越高，最终使得芯片无法正常运行。在早期的多核处理器设计中，主要通过降低核心频率来降低处理器的功耗，但是这样限制了核心的运算性能，并没有从根本上实现高性能、低功耗的目的。功耗过高不仅导致能源消耗，而且热堆积和过高的功耗密度也会对系统稳定性造成影响。现在一个芯片上可以集成 10 亿以上的晶体管，如此众多的片上资源，如何控制它的功耗，保持较高性能，成为了一个重要的问题。在多核处理器产生以前，低功耗技术主要有降低动态消耗和降低静态消耗技术两方面[21,22]。动态消耗包括处理器内部各元件正常工作时所消耗的电能，例如电容性的充放电、切换频率、逻辑门的状态转换等。降低动态消耗一直以来都是人们研究的重点，而且技术比较成熟。降低动态消耗技术现在主要有多元功能电压技术、动态电压调节、时钟屏蔽技术等。静态消耗技术是指来自漏电流的功率消耗，特点是即使元件处在空闲状态也会消耗电能，具体包括亚阈值漏电流和门漏电流。降低静态消耗技术的主要技术有通道长度调整、寄存器锁存技术、能量选通技术等。

上面两方面的技术主要在电路层次上进行低功耗设计和技术开发。在多核处理器出现以前，这些技术便已出现在单核处理器上。随着多核处理器的产生，由于多核处理器在结构和实现上有了新的特点，研究人员又在新的方面发现了降低功耗的方法，例如异构结构设计、动态线程分派与转移技术等。异构的结构设计就是利用异构结构对片上资源的最佳化配置，处理器的执行效率提升，使得处理器不仅具有高性能也降低了功耗。动态线程分派与转移技术是利用多核心处理能力，将某个核心上的过多负载转移到负载小的核心上，从而使处理器在不降低处理性能的情况下，降低处理器功耗。

研究人员还通过对操作系统的设计和优化来降低多核处理器的运行功耗。例如当任务较少时，操作系统会关闭一个核心或降低处理器频率，并降低封锁转速来使整个系统降低消耗。因此，低功耗设计包含了电路级、结构级、算法级和操作系统级等多个方面的内容，是一个需要从多方面进行综合考虑的问题。

4.3 操作系统设计

由于多核内部有多个核心，因此就存在任务分配、调度、仲裁以及平衡负载等问题，多核之间的任务调度是充分利用多处理器性能的关键。现有的操作系统还无法有效地支持多核处理器的任务运行。为满足实时处理的要求，均衡各处理器负载，任务调度机制需要研究的内容有设计和优化分布式实时任务调度算法、动态任务迁移技术等。

当前关于多核的任务调度算法主要有全局队列调度、局部队列调度和共生队列调度算法。全局队列调度是操作系统维护一个全局的任务等待队列，当系统中有一个 CPU 核心空闲时，操作系统便从全局任务等待队列中选取就绪任务并开始在此核心上执行，它的优点是 CPU 核心利用率较高。局部队列调度是指操作系统为每个 CPU 内核维护一个局部的任务等待队列，当系统中有一个 CPU 内核空闲时，便从该核心的任务等待队列中选取恰当的任务执行，局部队列调度的优点是任务基本上无需在多个 CPU 核心间切换，有利于提高 CPU 核心局部缓存命中率，缺点是 CPU 利用率太低。共生调度方法的基本思想是将访问共享资源较多的任务和访问共享资源较少的任务调度到同一时刻执行，从而最大程度上减少资源冲突。目前，多数多核的操作系统采用了基于全局队列的任务调度算法。

4.4 平衡设计问题

平衡设计是指在芯片的复杂度、内部结构、性能、功耗、扩展性、部件成本等各个方面做一定的权衡，即不能为了单纯地获得某一方面的性能而导致其它方面的问题，在设计过程中要从整体结构的角度去权衡各个具体的结构问题。

在多核处理器设计工程中，项目人员需要坚持平衡设计的原则。因为往往在减少一个方面问题的同时就增加了另一个方面的问题，所以在设计过程中要仔细权衡对某些问题的解决方法，尽量采用简单、易于实现、成本低廉而且对整体性能影响不大的设计。微处理结构设计的重点不在于其中某一个细节采用什么复杂或性能表现较好的设计，而是在于整体的设计目标。也即是说，要得到一个在通常情况下逻辑结构简单且对大多数应用程序有良好性能的微处理器结构，在适当的时候为了整体目标就不得不牺牲某一方面。当然在具体的设计中，不能只是简单地选择，应该是建立在科学的实验和模拟分析基础上的选择或平衡。因此在多核处理器设计中，要以科学分析的数据结果为基础，坚持合理平衡的设计原则。

除了上述挑战，多核处理器技术的发展过程中还涉及到存储结构设计、片上网络、应用软件设计开发等问题所带来的挑战，第 3 节在介绍多核处理器关键技术的时候已经对这些问题进行了阐述，此处不再赘述。

结束语 随着多核、异构成为研究热点，处理器技术正进入新的里程碑，创新型的处理器体系结构初露端倪。多核处理器结构通过采用简化单核结构、增加核心数目和片上部件等结构设计方法提高了处理器性能，适应了工艺发展和应用的需求，逐步成为了应用的主流。

多核处理器技术的进一步发展需要解决存储墙、片上网络、操作系统、编程模型和低功耗等几个重要问题，而这些问题的解决将是一个需要持续深入探索、综合权衡的过程，它要求研究人员根据设计目标和应用需求去采用适当的技术和解决方案。因此，深入理解多核处理器涉及的各项关键技术是多核处理器结构设计和实现的基础，明晰多核处理器所面临的技术挑战，理清研究思路，抓住研究重点，又将更进一步推动多核技术的发展。

目前，多核正向着众核方向发展，并呈现多核心、低功耗、异构和可重构等几个方面的特性，多核处理器的未来值得期待。

参 考 文 献

[1] 多核处理器——百度百科[EB/OL]. http://baike.baidu.com/

view/2797908.htm,2009
[2] Olukotun K,Nayfeh B,Hammond L. The case for a single chip multiprocessor[J]. ACM SIGPLAN notice,1996,31(2)
[3] CELL 处理器——百度百科[EB/OL]. http://baike.baidu.com/view/1347737.html,2009
[4] 王焕东,高翔. 龙芯 3 号互联系统的设计与实现[J]. 计算机研究与发展,2008
[5] 王海霞,汪东升. 多核/众核处理器的关键技术[J]. 中国计算机学会通讯,2009,5
[6] Gelsinger P. Microprocessors for the new millenium,challenges,opportunities,and new frontiers[C]// Proc. of the SolidState Circuit Conference. IEEE Press,2001
[7] Wang D. Meeting Green Computing Challenges [C]// Proceedings of HDP'07. 2007
[9] Nicopoulos C A,Park D,Kim J,et al. ViChaR:A Dynamic Virtual Channel Regulator for Network-on-Chip Routers[C]// Proceedings of the 39th Annual IEEE/ACM International Aymp on Microarchitecture. 2006:333-346
[10] Kumar A, Peh L, Kundu P. Express virtual channels: Towards the ideal Interconnection Fabric[C]// Proceedings of Int. Symp Comput. Architecture(ISCA07). June 2007:150-161
[11] Lu Z, Liu M, Jantsch A. Layered Switching for Networks on Chip[C]// Proceedings of Design Automatic Conference. June 2007:122-127
[12] Chishti Z,Powell M D,Vijaykumar T N. Distance associativity for high-performance energy-efficient non-uniform cache architectures[C]// Proceedings on 36th Annual IEEE/ACM International Symposium on. 2003:55-66
[13] Zhang M,Victim A K. Replication:maximizing capacity while hiding wire delay in tiled chip multiprocessors[C]// Proceedings on 32nd International Symposium on. June 2005:336-345
[14] Beckmann B M,Marty M R,Wood D A. ASR:Adaptive Selective Replication for CMP Caches[C]// 39th Annual IEEE/ACM International Symposium. Dec. 2006:443-454
[15] Hardavellas N, Ferdman M. Reactive NUCA: Near-Optimal Block Placement and Replication in Distributed Caches[C]// Proceedings of the International Symposium on Computer Architecture. June 2009
[16] Nilsson H,Stenstrom P. The Scalable Tree Protocol-a cache coherence approach for large-scale multiprocessors[C]// Proceedings of the Fourth IEEE Symposium. Dec. 1992:498-506
[17] Eisley N,Peh L-S,Li Shang. In-Network Cache Coherence[C]// Proceedings of 39th International Symposium on Microarchitecture. Florida,2006
[18] Awasthi M, Venkatesan V. Understanding the Impact of 3D Stacked Layouts on ILP[C]// The Journal of Instruction-Level Parallelism. 2007,9
[19] Dean J,Ghemawat S. MapReduce:Simplified Data Processing on Large Clusters[C]// 6th Symposium on Operating System Design and Implementation. California,December 2004
[20] Humphreys J. System Virtualization Profile [M]. Distributed Management Task Force,Inc.,2007
[21] 郝松,都志辉,王曼,等. 多核处理器降低功耗技术综述[J]. 计算机科学,2007,34(11):259-263
[22] 张骏,樊晓桠,刘松鹤. 多核多线程处理器的低功耗设计技术研究[J]. 计算机科学,2007,34(10):301-305

(上接第 294 页)

扩充。进一步的研究工作主要包括 3 个方面:1. 单通道识别技术的进一步研究。2. 多通道交互中的身份识别研究。困扰多人多通道协作交互的一个重要障碍就是身份的识别。这方面目前已有的解决办法都存在着对用户的限制条件,如何实现身份的自然识别是这一步研究的重点。3. 上下文的研究。本文中的上下文只是简单地对用户反馈进行了记录,类似于现在的输入法。但实际上在多通道交互系统中存在着其它更多的上下文信息,如何利用好这些上下文信息,是多通道研究的另一个重要方面。

参 考 文 献

[1] 美国国防高级研究计划. 美军未来指挥所(Command Post of the Future—CPoF)科研计划[R]. 1998
[2] 吴振锋,赵克俭. 未来指挥所发展展望[J]. 火力与指挥控制,2005,30(4):3-6
[3] 董士海. 人机交互的进展及面临的挑战[J]. 计算机辅助设计与图形学学报,2004,16(1):1-13
[4] 董士海,王坚,戴国忠. 人机交互和多通道用户界面[M]. 北京:科学出版社,1999:101-120
[5] Oviatt S,Cohen P,Wu L Z,et al. Designing the user interface for multi-modal speech and gesture applications[C]// Carroll J,ed. State-of-the-Art systems and research directions for 2000 and beyond. Human-Computer Interaction in the New Millennium. Boston:Addison-Wesley,2000:263-332
[6] Shafer S,et al. The new easy living project at Microsoft research [C]// Proceedings of the 1998 DARPA/NIST Smart Spaces Workshop. Gaithersburg,MD,1998:127-130
[7] 叶挺. 基于任务分析的指挥空间多通道交互方法研究[D]. 长沙:国防科学技术大学,2009
[8] 田丰,牟书,戴国忠,等. Post-WIMP 环境下笔式交互范式的研究[J]. 计算机学报,2004,27(7):977-984
[9] 王鹏,黄广连,等. 一种红外多点触摸式双手交互技术[J]. 小型微型计算机系统,2009,30(7):1467-1472
[10] 王鹏. 未来指挥所双手触摸式自然交互技术研究[D]. 长沙:国防科学技术大学,2007
[11] Paternò F, Mancini C, Meniconi S. ConcurTaskTrees: A Diagram-matic Notation for Specifying Task Models[C]// Proceedings of International Conference on Human-computer Interaction. Sydney,Australia:ACM Press,1997:362-369
[12] Pribeanu C. Task Modeling for User Interface Design: A Layered Approach[J]. International Journal of Information Technology,2006,3(2):86-90