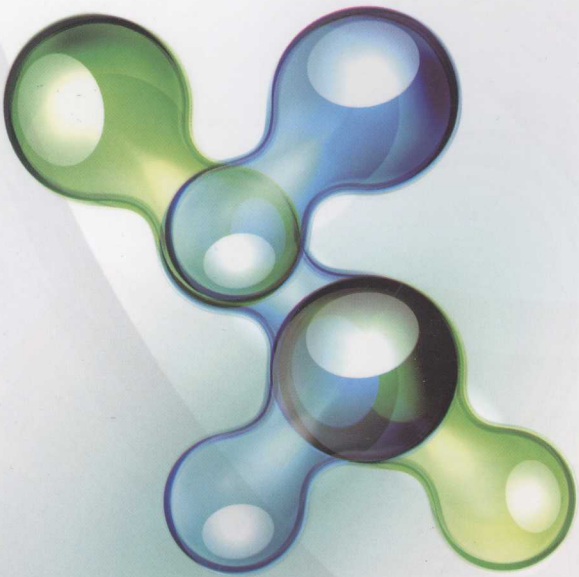
异构多核任务模型优化技术

曹倩 著



国防工业出版社
National Defense Industry Press

异构多核任务模型优化技术

曹倩 著

国防工业出版社

·北京·

内容简介

半导体工业的不断发展，促进了多核技术普遍应用，尤其是以 Cell 处理器为代表的异构多核，在专用计算方面表现出明显的优势。同时，随着实际应用的日益复杂，不确定边界的循环（如 while 循环）、递归调用（如斐波那契）等非规则、非结构化问题在分子动力学、油藏数值模拟、图形图像处理等应用中普遍存在。为了有效地提高这类应用的并行效率，2008 年 5 月发布的 OpenMP3.0 新规范引入了 task 编译指导语句，即任务模型，该任务模型使得上述非规则、非结构化应用的并行成为可能。如何利用现有的编程模型和编译优化技术降低异构多核处理器的编程难度，提高非规则、非结构化并行应用的执行效率，充分利用异构多核处理器的并行能力成为软件行业面临的一个重要问题。本书结合 Cell 异构多核架构的特点及 OpenMP 任务模型的规范，研究了 Cell 异构多核处理器下 OpenMP3.0 任务模型的实现及优化技术。

本书组织结构如下：第一篇为基础知识篇，着重介绍了多核处理器及其分类；第二篇重点介绍了 OpenMP、MPI、CUDA、OpenCL 等并行编程模型，同时对并行构件技术特别是 CCA 并行构件编程环境作了详细阐述；第三篇在分析了 Cell 异构多核处理器上支持任务模型的关键技术及国内外研究现状之后，分别阐述了 Cell 异构多核上 OpenMP3.0 的混合任务调度策略、Cell 异构多核上优化非规则访存的软件 Cache 模型以及 Cell 异构多核上优化任务粒度的自适应任务生成控制策略。同时，为了提高并行程序的复用性，提出了基于 CCA 的构件程序设计方案。

图书在版编目（CIP）数据

异构多核任务模型优化技术/曹倩著. —北京：国防工业出版社，2013.5

ISBN 978-7-118-08561-7

Ⅰ. ①异… Ⅱ. ①曹… Ⅲ. ①微处理器—并行编译程序—程序设计 Ⅳ. ①TP332②TP314

中国版本图书馆 CIP 数据核字（2013）第 084143 号

※

国防工业出版社出版发行

（北京市海淀区紫竹院南路 23 号 邮政编码 100048）

北京奥鑫印刷厂印刷

新华书店经售

*

开本 787×1092 1/16 **印张** 9 **字数** 210 千字

2013 年 5 月第 1 版第 1 次印刷 **印数** 1—3000 册 **定价** 33.00 元

（本书如有印装错误，我社负责调换）

国防书店：（010）88540777 发行邮购：（010）88540776

发行传真：（010）88540755 发行业务：（010）88540717

前　言

多核处理器是近年来半导体工艺发展的必然及主要方向，尤其是异构多核，由于其在特定应用方面表现的巨大优势，引起了国内外专家学者的广泛关注。OpenMP 并行编程模型支持增量并行，编写 OpenMP 程序可以从串行程序开始一步一步地并行化，尤其是 OpenMP 的任务模型使得非规则、非结构化并行成为可能。因此，研究异构多核处理器上 OpenMP 任务模型的意义更加明显。本书深入探讨并研究了异构多核处理器上 OpenMP 任务模型的实现及优化技术，书中涉及的优化策略可为研究并行编程模型的同行提供开阔的理论思路，而且本书结合作者在科研工作中的实际开发经验，对实际应用问题进行了深入剖析，并给出了切实可行的解决方案，为多核程序设计人员提供了丰富的代码，具有良好的参考价值。书中的一些优化方案现已成功地应用于油藏数值模拟、图像处理等实际领域中，有效地实现了理论与实践的结合。

本书给出了多核处理器的概念及多核处理器的关键技术，介绍了目前业界比较流行的多核处理器，并以 GPU 及 Cell 处理器为例，分别介绍了比较典型的同构多核及异构多核。论述了目前比较常见的并行编程模型，如 OpenMP、MPI、CUDA、OpenCL 等，并分析了这些并行编程模型的特点。同时本书还分析了目前国内外关于 Cell 异构多核处理器上支持 OpenMP 任务模型的关键技术及研究现状，提出了异构多核处理器上 OpenMP 任务模型的混合调度策略、Cell 处理器上自适应软件 Cache 模型以及 Cell 处理器上用以优化任务粒度的自适应任务生成控制策略。

本书旨在使读者了解多核处理器的基本概念，熟悉常用的多核处理器；了解常见的并行编程模型，特别是 OpenMP 并行编程模型，熟悉 OpenMP 的任务模型；能够利用 OpenMP 编程规范实现异构多核处理器的程序设计；能够根据实际应用的特点以及处理器的硬件优势达到降低 Cell 异构多核处理器编程难度的同时，提高程序员的编程效率，有效提高非规则、非结构化并行应用的执行效率。

由于作者水平有限，书中难免会存在疏漏之处，希望读者能够将问题及时反馈给我们，我们将表示衷心的感谢。感谢国防工业出版社为本书的出版所付出的心血。感谢北京工商大学对本书的出版提供的支持，以及许多老师提出了宝贵建议，在此一并致以诚挚的谢意。

目 录

第一篇 多核处理器

第二篇 并行编程模型及并行构件

第三篇 异构多核处理器上支持任务并行模型

第一篇　多核处理器

第1章　多核处理器概述

1.1　多核处理器的概念

在过去的几十年里，为了有效地提高处理器的工作性能，设计人员主要借助于增大数据宽度和提高处理器的运行频率。从20世纪70年代的几兆赫兹的4位处理器，到80年代的几十兆兹、几百兆赫兹的16位、32位处理器，再到90年代的几千兆赫兹的64位处理器，处理器性能的不断提高支撑着软件性能的逐步提高。但是，到了90年代末期，内存壁垒、功耗壁垒、频率壁垒的出现，使得单核处理器的性能提升遇到了瓶颈。主要在于以下几方面：

（1）内存壁垒。通常来讲，处理器的性能提升比内存要快得多，而内存的速度往往由于其容量大的特点而难以提高，所以程序性能在很大程度上取决于在处理器和内存之间移动数据的能力。对于那些高频率处理器，即使嵌入了内存控制器，处理器访问内存的延迟仍然接近1000个周期。尽管广泛使用的硬件高速缓存（Cache）可以在一定程度上缓解二者之间速度不匹配的问题，但是两者速度差异所带来的影响还是显而易见的。所以，即使处理器的频率能进一步提高，也会因访存速度慢而使性能大打折扣。

（2）功耗壁垒。由于处理器上集成的晶体管越来越多，处理器的功耗越来越高，当功耗高到处理器的散热系统无法将其温度控制在晶体管的正常工作范围内时，处理器就会无法工作，甚至会被烧毁。

（3）频率壁垒。为了更好地将指令执行进行重叠，传统的处理器主要通过增加流水线，然而，现代处理器的流水线级数已经达到40多级，如果仍然采用增加流水线级数来提高执行效率的话，其收益将会越来越小，如果把功耗考虑进来，甚至会出现负收益。

总之，由于上述因素的影响，在单核处理器上提高串行算法的性能变得越来越困难，因此，在单个芯片上集成多个处理器核从而构造多核处理器以充分发挥芯片内并行计算能力是处理器发展的主要的趋势之一[1,2]。

片上多处理器（ChipMultiprocessor，CMP），也就是多核[3,4]，在一个芯片内部封装了两个或多个计算核，其中每个计算核具备一些计算部件和高速缓存等。1996年，美国斯坦福大学首次提出片上多处理器的思想，并给出第一次多核结构原型；2001年，IBM推出第一个商用的多核处理器Power4；2005年，Intel和AMD的多核处理器获得了大规模应用；之后，多核处理器成为市场主流，在该过程中多核处理器经历了20年左右的发展。同时，

多核处理器的应用范围已覆盖个人计算机、多媒体计算、嵌入式设备、商用服务器和高性能计算机等众多领域。多核处理器将多个功能的核心集成在一个芯片内，整个芯片作为一个统一的结构对外提供服务。首先，多核处理器通过在一个单独的芯片内集成多个单线程处理核心或者多个同时多线程处理核心，使得集成后的处理器可同时执行的线程数或任务数是单个处理器的数倍，这极大地提升了处理器的并行处理能力；其次，多个核集成在一个芯片内，明显地缩短了核间的连线，降低了核间通信延迟，提高了通信效率，因此数据传输带宽也得到提高；再次，多核处理器可以有效地共享资源，提高片上资源的利用率，降低了功耗；最后，多核结构简单，易于优化及扩展。总之，多核处理器结构的一个重要优势在于它利用多个核心的并行处理能力，采用多个结构简单、电压和频率较低的计算核来达到与传统复杂的微处理器相同的性能，所以多核架构降低了系统设计实现的难度，减少了系统的能量消耗。因此，多核处理器在学术界和工业界赢得了青睐，成为高性能微处理器发展的主流。

1.2 多核处理器的关键技术

多核处理器结构不仅性能潜力大，集成度高，并行度高，结构简单，设计验证方便等，而且它还能继承传统单核处理器的同时多线程，宽发射指令，低功耗技术等。但与单核处理器相比，多核处理器无论在体系结构、功耗，还是在安全性设计等方面均面临着巨大的挑战。多核处理器致力于发掘计算的粗粒度并行，随着大规模集成电路技术的发展，在芯片容量足够大时，就可以将大规模并行处理机结构中的对称多处理机（Symmetrical Multi-Processing，SMP）或分布共享处理机（Distributed Shared Memory，DSM）节点集成到同一芯片内，各个处理器并行执行不同的线程或进程。在基于 SMP 结构的单芯片多处理机中，处理器之间通过片外高速缓冲存储器（Cache）或是片外的共享存储来进行通信；而基于 DSM 结构的单芯片多处理器中，处理器间通过连接分布式存储器的片内高速交叉开关网络进行通信。由于 SMP 和 DSM 已经是非常成熟的技术，CMP 结构设计比较容易，只是后端设计和芯片制造工艺的要求较高而已。因此，CMP 成为了最先应用于商用 CPU 的“未来”高性能处理器结构。虽然多核处理器使得芯片的性能成倍地增加，但很明显，原来系统级的一些问题引入到了处理器内部。

1.2.1 核心结构的选择

目前，多核处理器的核心结构主要有同构和异构两种。顾名思义，同构多核处理器是指处理器芯片内部的所有核心其结构及地位是相同的。目前的同构多核处理器大多数由通用的处理器核心组成，每个处理器核心可以独立地执行任务，与通用单核处理器结构相近。同构多核处理器可以依据其互连的层次在结构上予以区分，即核心之间可以通过共享存储器互连，也可以在 Cache 层面互连。异构多核处理器芯片内部采用多种功能不同的核心，如有的核心主要负责管理和调度，有的核心主要负责计算。目前的异构多核处理器通常将通用处理器、DSP、媒体处理器、网络处理器等多种类型的处理器核心集成在一个芯片上，其中的通用处理器往往是负责管理和调度的主核，其他的处理器作为负责计算的从核。核本身的结构，关系到整个芯片的面积、功耗和性能。怎样继承和发展传统处理

器的成果,直接影响多核的性能和实现周期。同时,根据 Amdahl 定理,程序的加速比决定于串行部分的性能。核所用的指令系统对系统的实现也是很重要的,采用多核之间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,也将是研究的内容之一。

1.2.2 片上通信技术

多核处理器中的每个核心各自执行自身的任务,但是各个核间需要进行通信以完成数据的同步与共享等,因此片上核间通信的性能是影响多核处理器整体性能的重要因素之一。通常,片上通信的方式主要有以下 3 种,即总线共享、交叉开关互连和片上网络(Network On Chip,NOC)。总线共享结构是指多核处理器上的各个核心通过共享二级、三级 Cache 结构,或者通过连接核心的总线进行通信。总线结构的优势在于较为简单,易于设计实现,当前多数双核和四核处理器基本上都采用了该结构,但总线结构的劣势在于可扩展性较差,主要适用于核数目较少的情况。比较典型的总线共享结构处理器有 Intel 的 Core(酷睿),IBM 的 Power4、Power5 等。交叉开关互连结构由交叉开关以及接口逻辑构成。与总线结构相比,交叉开关的优势在于数据通道多,访问带宽大,但交叉开关结构的不足表现为占用的片上面积较大,而且核心数增加时,性能也会有所下降,因此该结构也适用于片上核数量较少的情况。比较典型的是 AMD 公司的 Athlonx2 双核处理器。片上网络结构将互连网络用于片上系统设计,该方案借鉴了并行计算机的互联网络从而实现片上核间的通信问题。片上网络类似于并行计算机的互连,如支持包通信,可扩展,提供透明的通信服务等;但也不完全相同,如片上网络技术支持同时访问,而且有可靠性高以及可重用性高等特点。与总线结构、交叉开关结构相比,片上网络可以连接更多 IP 组件,可靠性高,可扩展性强及功耗较低,因此片上网络被认为是更加理想的大规模 CMP 互连技术。目前,片上网络设计的问题在于寻找网络开销和多核耦合程度最佳的平衡,并同时考虑网络的可扩展性。RAW 处理器就采用了片上二维网络结构,它通过集成高速网络和优化的路由算法,片上核心间的通信延迟最大不会超过 6 个周期,而且该结构可扩展性较强。上述 3 种结构虽各有优点和不足,但是也可以融合,例如在全局范围内采用片上网络而在局部区域选择总线或者交叉开关结构,以实现性能与复杂性之间的平衡。

1.2.3 多核与 I/O 结构

多核的出现对系统的 I/O 能力提出了更高的要求。近年来,标准 I/O 接口的传输率和带宽等主要性能指标不断提高,并从并行传输逐步转变为串行传输,极大地改善了以往普遍存在的 I/O 瓶颈,同时有效地改善了远程访问的延迟和带宽。当然,与单核处理器相比,多核处理器的 I/O 更为复杂,对于同构的多核环境,每一个处理器核在系统中的作用和地位是相同的,都具备独立的 I/O 操作的可能,这使得系统特别是操作系统必须给予相应的调度与管理技术支持;而对于异构的多核环境,通常主核运行完整的操作系统,从核主要负责计算,I/O 操作一般由主核统一完成,系统层面上,这与单核的情形相似。因此,研究与计算能力相匹配的新的 I/O 技术十分必要。I/O 能力的提高不仅仅是总线接口需要考虑的问题,也应该是从处理器内外综合考虑的问题。当前的研究主要包括:下一代总线架构、新 PCI - Express、DDR3 内存、全缓冲 DIMM 和 I/O 加速技术等,其中 I/O 加速技术包括借助增强特性优化 TCP/IP 堆栈,借助 I/O 加速网卡实现 CPU 平衡处理,借助直接

内存访问技术增强数据传输能力等。

1.2.4 存储结构设计

主存储器存取速度一直比中央处理器操作速度慢得多，使中央处理器的高速处理能力不能充分发挥，整个计算机系统的工作效率受到影响，因此存储系统自身的结构设计直接影响到系统的整体性能。处理器与主存储器之间的速度不匹配成为处理器结构设计中必须要面对的问题，在单核处理器中，设计中通过采用缓存结构基本上能够较好地解决该问题，保证处理器的潜能较好地发挥出来。可是，多核处理器日益普遍，多核处理器的处理能力和主存之间的速度差距变得更为严重。随着处理器内部核心数目的增多，对主存的访问需求也逐渐增加，单核时代的缓存层次和访问带宽已经无法满足多核处理器的访问需求，因此必须针对多核处理器进行相应的存储结构设计，以更好地解决存储系统的效率问题。

对于存储系统设计，目前大部分处理器采用缓存设计，也有某些处理器采用的是片上存储器结构，两种设计各有优缺点。前者的优势在于易于设计、易于开发、易于编程。缺点在于需要通过某种机制实现数据的一致性问题，并且不易于扩展。为了保证缓存数据的一致性，解决的方案主要包括总线侦听协议及基于目录的目录协议。与硬件缓存设计不同，片上存储器是将片外的存储器引入到片内，它与片外存储器一样采用统一编址的方式，所以避免了缓存不命中和数据一致性问题，但由于它采用了存储器结构，所以访问延迟比缓存要稍微大一些。当前有一些研究人员采用高速动态随机存储器实现片内存储器，有效地缩小了与缓存间的速度差距。当然，除了选择何种存储设计外，存储结构设计的问题还包括存储器的容量；在哪一级实现数据的共享和通信；在哪一级解决缓存数据的一致性问题；存储结构如何支持多线程的应用等。

分析多核处理器，我们发现 Cache 的结构设计与其数据一致性问题之间存在一定的矛盾，所以多核的 Cache 设计通常采用多级结构，比如二级 Cache、三级 Cache 等。但是一般来讲，通常会共享一级 Cache 或共享二级 Cache。对于三级 Cache，目前的多核大部分采用独有的结构。通常来讲，Cache 的结构会直接影响到系统的性能，究竟采用几级 Cache 结构以及共享几级 Cache 都是需要仔细研究和讨论的问题。同时，多级 Cache 结构还会给数据一致性带来一定的困难。

1.2.5 程序执行模型

选择合理的程序执行模型是多核处理器设计的关键问题之一，是编译器开发人员与系统实现人员之间的接口。编译器开发人员需要考虑的是如何将某种高级程序设计语言按某一程序执行模型转换成一种目标机器语言；系统实现人员则需要考虑在具体的目标机上如何有效地实现该程序执行模型。当多核处理器作为目标机器时，会产生如下问题：多核处理器架构如何支持程序执行模型？针对多核体系结构哪种程序执行模型更合适？这些程序执行模型是否具有普遍的应用价值？是否能够在商业上得到推广？不夸张地说，程序执行模型在一定程度上决定了多核处理器能否以最低的代价提供最高的性能。

1.2.6 功率和热管理

在计算机系统中，处理器所占的功耗不能忽略，特别是高性能的处理器的功耗往往占系统功耗的50%。在目前阶段，基本上处理器性能每提高1%芯片功耗就要增加3%，可以说功耗直接影响了系统的可靠性、可用性以及系统的规模。计算机系统的功耗主要分静态功耗与动态功耗两类，前者主要受与频率相关的漏电流的影响，漏电流随频率的增高而增加；后者与工作频率及工作电压的平方成正比，因此，控制频率和降低电压是降低处理器芯片功耗的重要手段。频率主要用于反映处理器峰值运算的能力，在多核环境下，处理器芯片的工作频率和工作电压一般都比单核处理器低。

1.2.7 操作系统设计

由于多核芯片内部包含多个核心，要想将任务在各个核间顺利地完成，就需要进行任务分配、调度、仲裁以及平衡负载等，任务在多核之间的合理调度是充分发挥多核处理器性能的重要表现。目前的操作系统还不能高效地支持任务在多核处理器上的运行。为了实现任务在多个核间的负载均衡，操作系统需要考虑任务的调度算法、动态任务负载均衡等。当前阶段研究比较多的有任务队列的组织，如全局任务队列结构和分布式任务队列结构。全局任务队列结构指的是系统仅维护了一个全局的任务队列供所有线程共同访问，当有任意一个线程处于空闲状态后，就可以从全局任务队列中获取一个就绪任务执行，采用全局的任务队列可以避免负载不均衡，充分利用线程的计算能力，但是该方案需要考虑线程获取任务时的冲突问题。分布式任务队列是指每个线程维护自身的一个局部的任务队列，当某个线程空闲时，它便会从自身的任务队列中获取就绪任务执行。该任务队列的组织结构优点在于不需要考虑多个线程对任务进行存取时的冲突问题，因为每个线程仅对自身的任务队列进行访问。但该方案也有一定的不足，如负载均衡不易实现。当然，除了上述的全局任务队列和分布式任务队列之外，共生调度也是较常见的一种策略，其基本思想是在同一时刻调度执行那些访问共享资源较多的任务和访问共享资源较少的任务，这就有效地避免了共享资源冲突。当前，大部分多核操作系统采用的是全局任务队列的组织方式。

1.3 多核处理器的发展趋势

多核处理器产生的直接原因是为了替代以前的单核处理器以解决微处理器的发展瓶颈，但发展多核的更客观的原因在于满足现实应用对计算性能的更高层次的需求，而且这种压力还会日益明显。即便在当前，多核开发人员已经以一个非常快的速度实现多核处理器的更新，但实际应用对处理器性能的需求仍未消失。当前，多核处理器向前发展仍然受到一定的阻碍，如功耗和应用开发等。为了实现低功耗，实现应用的高效开发，多核处理器的发展呈现出以下几种发展趋势：

（1）多核上将集成结构设计更简单，功耗更低的处理器核心。为了满足人们对高性能的日益需求，就要考虑在单个芯片上集成更多的核心，在增加核数目的同时还要考虑如何降低功耗以及降低延迟。

(2) 异构多核是多核处理器发展的一个重要方向,通过在多核处理器上运行实际的应用我们发现,综合考虑处理器的结构、功能、性能及功耗等多个因素,异构多核处理器由于在单个芯片上集成了不同功能和地位的核心,主核用来运行操作系统,起到控制、管理的功能,而从核专注于计算,这样通过各个不同核心侧重于不同的任务从而高效地运行各种不同的应用,因此异构多核的组织方式能更高效地执行应用,实现了资源的合理利用,而且对于降低功耗起到了非常重要的作用。

(3) 多核处理器上应用可重构技术。考虑到在芯片上应用 FPGA(Field Programmable Gate Arrays)技术可以有效地发挥高性能、高可靠性、高灵活性、低成本及低功耗等优势,半导体工艺上便将 FPGA 技术应用到多核设计上,让多核处理器具备可编程性及可重构性的特点,这种方案不仅有效地提高了多核处理器的通用性,而且使多核处理器具备了专用集成电路的高性能的优势。

总之,多核处理器结构的出现及普遍应用,对半导体工艺的发展具有极其深远的影响。毫无疑问,在未来相当长的一段时间内,多核芯片将在处理器市场上占有极为重要的地位。但是如何才能更有效地发掘多核处理器的潜能,更好地发挥多核处理器的性能优势,让多个任务在各个核间更合理地进行分配,从而提高多核处理器的利用率,不仅需要在硬件设计上实现合理的多核互联,还需要更有效地进行任务分配、负载均衡等。

第2章　多核处理器的分类

根据多核芯片上各个核的功能和地位是否相同，可以将多核处理器划分为同构多核（Homogeneous Multicore）处理器和异构多核（Heterogeneous Multicore）处理器。同构多核处理器在一个芯片内集成了两个或多个结构和地位相同的核，各个核的线路与主频都相同，因此设计实现相对比较简单。同构多核可以有效地提高系统的并行性，但是在一些复杂应用领域中并不能提供最优的性能。异构多核处理器在单个芯片上集成了多个结构不同的处理器核，不同结构的处理器核负责处理工作负载中不同类型的操作。这样，异构多核处理器对于某些特定应用领域比同构多核处理器能发挥出更高的性能。

2.1　同构多核处理器与编译技术

2.1.1　同构多核处理器的概念

在同构多核处理器中，每个核的主频及线路都相等。采用同构多核处理器设计，优势在于设计与工艺都较为简单。当前，同构多核处理器的主要代表有 IBM 的 Power4、Intel 的酷睿系列、AMD 的皓龙以及 Sun 的 UltraSPARC TI，这些多核处理器都是在同一个芯片内集成两个以上结构相同的处理器核。其中的 Power4[5] 是最早的商用处理器多核。目前市场上经常会看到 Intel 的双核、四核与八核处理器。AMD 将内存控制器集成到多核处理器中，并进一步考虑将图形处理单元集成到处理器芯片中，采用该设计方案，AMD 的多核处理器中出现了异构多核处理器。UltraSPARC TI 在单个芯片上集成了 8 个结构相同的 4 路多线程核心，该结构 UltraSPARC TI 处理器能够同时运行 32 个线程。

由于同构多核处理器共享了大部分底层存储资源，如系统主存储器及某级 Cache，不管是分布式存储模型，还是共享存储模型，在同构多核处理器上都可以得到较好的移植。一般来讲，同构多核处理器的规模不是很大，可以借助编译器在一定程度上实现程序并行化。例如，Intel 的编译器实现了对 OpenMP 编程模型的支持，针对同构多核实现了自动并行的功能。在同构多核处理器上实现程序的并行化时，需要考虑数据局部性问题、流水问题、负载均衡性及任务划分等。

当然，随着半导体工艺的不断发展，多核处理器芯片上势必会集成越来越多的处理器核心，开发多核处理器的并行执行效率、降低多核处理器的编程难度将是编程人员面临的一个重要问题。

2.1.2　GPU 同构多核处理器

GPU 即图形处理器，英文全称 Graphic Processing Unit，简单来讲，GPU 就是显卡的中央处理器。GPU 是相对于 CPU 的一个概念，随着现代计算机的发展以及人们对图像应用

的需求越来越高，在计算机系统中需要一个专门的图像的核心处理器，即 GPU，它是一个非常典型的同构多核处理器。GPU 通用计算应用最广泛的就是各种高性能计算领域，包括航空航天装备研制、卫星遥感数据处理、气象预报、海洋环境数值模拟、石油勘探数据处理、金融工程数据分析、新材料开发和设计、计算结构力学、生命科学计算、流体动力学、计算机视觉、数据挖掘、数值分析、科学计算、生物医药研究乃至天文演算。GPU 在计算机图形处理方面表现优异，具有极高的并行能力，因此与一般的 CPU 处理器相比，更善于处理一系列复杂的运算。在个人计算机上，GPU 往往集成于显卡或者主板。图 2.1 所示为 GPU G80 的架构图。

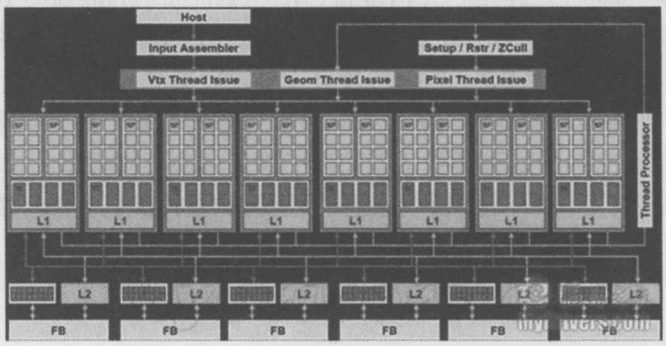


图 2.1　GPU G80 架构图

图形芯片最初用作固定功能图形管线。随着半导体工艺的不断发展，这些图形芯片的可编程性日益增加，于是英伟达（NVIDIA）公司推出了首款 GPU。1999 - 2000 年间，计算机方面的学者与电磁、医疗成像领域的科学家协同在 GPU 上进行通用计算应用程序的开发。在此过程中，他们发现 GPU 在浮点运算方面的突出表现为科学应用领域、特别是大规模数据处理领域的应用开发提供了良好的硬件平台，同时 GPU 具有价格低、速度快等明显优点，于是 GPU 被广泛应用到通用计算领域中，即 GPGPU（General - Purpose computation on GPU，图形处理器通用计算）。

2.2　异构多核处理器与编译技术

2.2.1　异构多核处理器的概念

顾名思义，异构多核处理器，是指芯片上不同的核可以根据不同的需求进行相应的设计，以增强多核系统的整体处理能力[6-8]。与同构多核处理器相比，异构多核处理器的硬件设计更为复杂，不同的核中需要运行不同的指令集，从而实现不同的运算功能。相应地，任务在异构多核处理器上执行时负载划分也会变得更为复杂，任务的执行模式也会变得复杂。典型的异构多处理器有 STI（SONY、TOSHIBA、IBM）联合开发的 Cell 处理器、斯坦福大学的 Merrimac 超级流计算机、ClearSpeed 的 CSX600 处理器等，其中 ClearSpeed 的 CSX600 处理器是作为协处理器运行的。Cell 异构多核最初是面向游戏、高清电视等多媒体领域而开发的一款处理器，它的芯片上包含一个基于 Power 架构的主处理器（Power Processing Element，PPE）和 8 个基于 SIMD 模式的协处理器（Synergistic Processor Element，