

多处理机体系结构虽然复杂多样,但经过二十多年的研究与发展,也逐步形成了应用范围广、体系结构相对稳定的多处理机。本章讨论了:多核处理器、机群多处理机、大规模并行多处理机的组织结构与性能特点及其分类,多核处理器的发展缘由及其多线程超线程技术,机群多处理机软件的组织结构及其单一系统映像,大规模并行多处理机软件的组织策略。介绍了机群多处理机的关键技术、MPP 的支持技术、各种典型多处理机的体系结构及其节点与互连网络的组成结构。

3.1 高性能微处理器及其多线程

3.1.1 多核与多核处理器

1. 多核处理器及其与多处理机的差异

多核处理器指在一个处理器中集成多个(两个或两个以上)运算核心或计算引擎,通过任务划分来充分利用多个运算核心,并行处理多个任务。可见,核心可以认为是增强型运算单元,核心与核心之间联系非常紧密。另外,也可以认为处理器核心是一个相对简单的单线程微处理器或比较简单的多线程微处理器,多个微处理器就可以并行地执行程序代码,实现线程级并行。所以,多核处理器又可以称为单芯片多处理器(chip multiprocessors, CMP)。最早的多核是双核,由 IBM、HP、SUN 等支持 RISC 架构的高端服务器厂商提出。多核处理器性能发挥的基础是多线程技术,如果运行单程序、处理单任务,在同频率时,多核处理器与单核处理器的实际性能是一样的。

多核处理器与多处理机是不同的概念,多核处理器是一个处理器中包含多个共享多级 Cache 的运算核心,一枚处理器芯片内部通过高速总线连接,使多个运算核心协同工作;多处理机是一台处理机中包含多个处理机,处理机通过主板线路连接,使多个处理机交互作用。

2. 多核处理器的性能特点

由于多核处理器采用相对简单的微处理器作为处理器核心,使得多核处理器主要具有以下几个显著的特点。

(1) 控制简单容易实现。相对超标量结构与超长指令字结构,多核结构控制逻辑的复杂性明显低得多,相应的多核处理器的硬件实现必然简单得多。

(2) 高主频。由于多核结构的控制逻辑相对简单,包含极少的全局信号,因此线时延对其影响比较小;在同等工艺条件下,多核处理器的硬件实现能够获得比超标量微处理器和超

长指令字微处理器更高的工作频率。

(3) 低通信时延。由于多个简单处理器集成在一枚芯片上,且共享多级 Cache 或主存,多线程之间的通信时延会明显降低,当然这也对存储层次提出了更高要求。

(4) 低功耗。通过动态调节电压/频率、负载优化分布等,可有效降低多核处理器功耗。

(5) 设计与验证周期短。微处理器厂商一般采用现有的成熟单核处理器作为处理器核心,从而可缩短设计和验证周期,节省研发成本,也便于扩展。

3. 多核处理器的分类

多核处理器可以按共享存储层次和处理器核是否相同来分类。

按共享存储层次来分,可以把多核处理器分为 3 种:共享一级 Cache、共享二级 Cache、共享主存,这种分类侧重于存储层次的组织 and 处理器核的连接。在共享一级 Cache 的多核处理器结构中,一级 Cache 由多个处理器核所共享,也就是处理器核在一级 Cache 这个层次上相连接。在共享二级 Cache 的多核处理器结构中,每个处理器核拥有独立的一级 Cache,二级 Cache 由多个处理器核所共享,即处理器核在二级 Cache 这个层次上相连。在共享主存的多核处理器结构中,每个处理器核不仅拥有独立的一级 Cache,还拥有独立的二级 Cache,或存储系统中不设 Cache,主存由多个处理器核所共享,即处理器核在主存这个层次上相连接。若应用程序的通信量大时,共享一级 Cache 多核处理器结构的性能优于另两种多核处理器结构;若应用程序的通信量小时,共享一级 Cache 多核处理器结构的性能与另两种差不多。但共享一级 Cache 多核处理器结构的设计较为复杂,处理器核之间的耦合度比较高,在处理器数目增加时可扩展性较差。

按处理器核是否相同来分,可以把多核处理器分为两种:同构的和异构的,这种分类侧重于处理器核的结构与功能。同构多核处理器一般由通用处理器核组成,多个处理器核执行相同或类似的任务,如 Intel 公司、AMD 公司推出的面向 PC 的双核、4 核、8 核处理器都属于同构多核处理器。异构多核处理器除含有通用处理器核作为控制、通用计算之外,还集成 DSP、ASIC、媒体处理器、VLIW 处理器等针对特定的应用来提高计算的性能。

3.1.2 多核处理器产生的原因

1. 单核处理器的局限性

从 2008 年开始,单核处理器基本停产,取而代之的是多核处理器。单核处理器的局限性主要体现在以下几个方面。

(1) 实现指令级的并行度有限。理想处理器指消除了所有指令级的并行约束,具有无限发射指令的能力的处理器。当然,这种处理器是不可能出现的,也是难以实现的。指令级并行约束是指令之间的相关性,相关性分析单靠静态编译分析是远远不够的,更多需要硬件动态分析。用于存储被检测指令集合的窗口大小受限于存储容量、能够承受的比较次数、寄存器的数目和有限的发射速率等,而窗口大小又直接限制了在给定时钟周期内处理指令的数量。另外,一个时钟周期发射指令数量、功能单元及其时延时间与队列长度、寄存器文件端口、对转移发射的限制、对存储器并行访问的限制和对指令提交的限制等都是影响指令级并行的因素。

(2) 处理器主频与主存、I/O 访问速度的发展极不平衡。2007 年以前,处理器的主频每

2 年翻一番,而主存的访问速度每 6 年提高一倍、I/O 的访问速度每 8 年提高一倍,这种不平衡已经成为单核处理器性能提高的瓶颈。单纯依靠提高处理器主频来提升整机的性能已经不可行,反而会造成效率降低,因为大部分时间 CPU 都在等待主存或 I/O 访问的返回才能继续下一步的工作。

(3) 频率已近极限、功耗大。单核处理器功耗有静态功耗与动态功耗之分,静态功耗随晶体管的数量成比例增大,动态功耗与晶体管的切换次数与切换速率的积成正比。处理器的频率越高,晶体管的切换次数与速率越大,动态功耗的比例越大。对于多发射处理器,相关分析的逻辑开销增长比发射率的增长要快,发射速率峰值的增大同晶体管的切换次数成正比。为了得到更高的性能,必然持续维持高逻辑开销和高发射速率,从而带来高动态功耗。一般认为,当处理器主频提高到 4 GHz 时,几乎接近目前集成电路制造工艺的极限,也限制了超标量的宽度和超流水的深度。

(4) 性价比变低。高频处理器使得设计和验证所花费的时间变得更长,设计对工艺要求非常高、成品率较低、生产难度大,由此导致成本高,往往性价比比较低。

2. 多核处理器产生的基础

单核处理器的局限性是多核处理器产生的前提条件,技术发展和应用需求则是多核处理器产生的基础和必然要求。

(1) 门时延逐渐缩短,线时延不断增长。随着 VLSI 工艺技术的发展,晶体管特征尺寸不断缩小,使得晶体管门时延不断减少,但互连线时延却不断变大。当芯片的制造工艺达到 $0.18\mu\text{m}$ 甚至更小时,全局连线时延已经超过门时延,成为限制电路性能提高的主要因素。在这种情况下,由于多核处理器是分布式结构,全局信号较少,与集中式结构的超标量处理器结构相比,克服线时延影响更具优势。

(2) 符合 Pollack 规则。按照 Pollack 规则,处理器性能的提高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,至多能提高性能 40%;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,同时面积也同比缩小。

(3) 有效降低功耗。随着工艺技术的发展 and 芯片复杂性的增加,芯片的发热现象日益突出。多核处理器中单个核的速度较慢,处理器消耗的能量较少,产生的热量也较少。同时,原来单核处理器中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)一些处理器核等低功耗技术,可以有效地降低功耗。

(4) 有效降低设计成本。随着处理器结构复杂性的不断提高和人力成本的不断攀升,处理器设计成本随时间呈线性甚至超线性地增长。多核处理器通过处理器的复用,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。

(5) 体系结构发展的必然。超标量结构和超长指令字结构已广泛应用于高性能微处理器,但它们都遇到了难以逾越的障碍。超标量结构使用多个功能部件并行处理多条指令,实现指令级并行,但控制逻辑复杂、实现困难,研究表明:超标量结构的并行度一般不超过 4。超长指令字 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指令代码的兼容。而多核处理器可以充分利用应用程序的指令级与线程级并行性,从而可显著提高性能。

3.1.3 多线程与超线程

1. 线程及其与进程的区别

在一些应用程序中,蕴藏大量任务或作业等高级别并行,而开发指令级并行的方法对任务或作业的并行无能为力。例如,在联机事务处理系统中,查询操作和更新操作常常互不相关,可以并行处理。因此,便在指令并行与任务或作业并行之间,提出线程并行(thread level parallelism, TLP)。线程指可以独立执行的顺序控制的程序段,它拥有自己的指令与数据,且具备执行的所有条件(指令、数据、状态等)及其资源(PC、寄存器组、堆栈等)。一个线程可以是并程序的一部分,也可以是一个独立的程序。

线程与进程不同,进程是整个程序或部分程序的动态执行实例,线程是进程内部的一个执行单元,线程比进程“轻巧”得多。一个进程包含一个到多个线程,不同进程有不同数据空间,而多个线程可以共享数据空间。线程切换时仅需要保存和设置少数寄存器内容,开销很小,仅需要几个时钟周期甚至仅需要一个时钟周期;而进程切换一般需要成百上千个时钟周期。

2. 多线程及其分类

多线程指同一程序中的多个线程并行运行,以处理不同的任务,在单核处理器中实现多线程的技术称为多线程技术。在指令级并行过程中,由于存在相关或停顿,在数据路径上经常会有一些空闲的功能单元。如果通过巧妙安排,使多个线程以重叠方式共享处理器中的功能单元,从而利用由指令级并行造成空闲的功能单元。为了支持多个线程共享功能单元,处理器必须为每个线程保持指令状态。例如,每条指令都需要有一个独立的寄存器文件副本、一个独立的 PC 值及一个独立的页表;主存储器则可以通过虚拟存储机制实现共享。此外,硬件还必须对不同线程之间的快速切换提供支持。

按线程粒度大小,多线程可分为两种:细粒度多线程和粗粒度多线程。

细粒度多线程指可以在指令之间进行线程切换,使多个线程交替处理。在某线程停顿时,切换到其他线程去处理其中的指令,从而可以在任一时刻跳过所有停顿的线程,也就要求处理器必须具有在任意时钟周期切换线程的能力。细粒度多线程的优点是可以隐藏由停顿引起的吞吐量损失,缺点是降低了每个线程的处理速度(其他线程指令的插入执行使线程不能连续执行)。

粗粒度多线程指仅遇到代价较高的停顿时(如第二级 Cache 不命中)才发生线程切换,从而在很大程度上避免处理器速度的降低。由于粗粒度多线程的处理器从单独线程发射指令,因此当停顿发生时,流水线必须被清空或暂停,停顿后开始处理新线程,新线程指令必须填满流水线后,才开始有指令结束。这就使得粗粒度多线程的流水线启动开销(建立时间)较大,克服吞吐量损失的能力有限。不过,对于较长的停顿而言,这种启动开销通常可以忽略不计。

3. 超线程技术及其实现

超线程指将一个物理处理器在逻辑上当作两个处理器使用,以使处理器可以并行处理多个线程,提高处理器的效率。超线程技术是利用特殊的硬件指令,把两个逻辑核模拟成两个物理芯片,使单处理器都能实现线程级并行,减少处理器的空闲时间。通常把可以实现超线程技术的处理器称为超线程处理器,其他的称为单线程处理器。

当采用超线程技术使操作系统或应用程序的多个线程运行于一个超线程处理器上时,两个逻辑核共享物理处理器的功能单元,这样可以使得物理处理器的处理能力提高 30%。

单线程处理器虽然可以每秒处理成千上万条指令,但在许多时刻,仅对某个线程的一条指令进行处理,从而导致处理器许多功能单元空闲。而超线程处理器通过并行执行多个线程,使得处理器在任何时刻都可以并行处理多条指令,避免功能单元的空闲。超线程处理器的两个逻辑核均可以单独进行中断响应,当第一个逻辑核跟踪处理一个线程时,第二个逻辑核则开始跟踪处理另一个线程,且为了避免功能单元的冲突,第二个逻辑核的线程仅需要使用被第一个逻辑核的线程闲置的功能单元。例如,第一个逻辑核的线程执行的浮点运算,使用浮点运算单元,那么第二个逻辑核的线程可以执行的整数运算,使用整数运算单元。

超线程技术由 Intel 公司提出,最早出现在其 2002 年推出的 Pentium 4 上,但由于当时支持的应用软件缺乏,其优势无法体现,之后 Intel 公司推出的微处理器也未应用该技术。直到 2009 年 Core i 系列处理器的诞生,超线程技术开始全面应用推广,如 4 核 Core i7 可以支持 8 个线程。

4. 超线程与双核的区别

基于超线程技术的单核处理器 Pentium 4 与基于多核技术的双核处理器 Pentium D,在操作系统中均被识别为具有两个处理器,但有本质区别。支持超线程的 Pentium 4 是两个逻辑处理器可以并行处理两个线程,但它们并没有独立的功能单元、寄存器、总线接口和缓冲存储器等,并行处理两个线程共享资源,若两个线程同时需要同一资源,其中一个线程必须暂停,可以认为:超线程技术通过软途径来优化利用处理器中的资源,以提高运行效率。支持双核的 Pentium D 是两个物理处理器核可以并行处理多个线程,每个物理核有独立的功能单元、指令集等,性能比超线程 Pentium 4 高得多,可以认为:双核技术是通过硬途径来提高性能的。

3.1.4 多线程实现途径及其支持技术

多线程实现有软件和硬件两条途径,即按实现途径,多线程实现可分为软件多线程和硬件多线程两种。

1. 软件多线程及其支持技术

软件多线程指在单处理器上实现多线程,使多个线程共享同一个计算资源,由操作系统通过动态改变线程运行优先级来调度线程。当某线程等待某些资源时,可进行线程切换,让另一个线程运行。软件多线程通过多个线程共享 Cache 进行数据交换,通信开销被隐藏。如图 3-1 所示为软件多线程运行状态时空图,图中由不同填充模式的条形表示线程所处的不同状态。

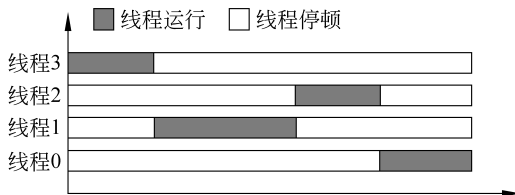


图 3-1 软件多线程运行状态时空图

2. 硬件多线程及其支持技术

硬件多线程指在多核处理器或多个处理器上实现多线程,多个线程使用不同的计算资

源,使多个线程真正并行运行,获得比单处理器高得多的性能。硬件多线程通过显式硬件通信实现,线程之间存在通信开销。运行于不同处理器(核)的各个线程分别具有自己的优先级,某个处理器(核)上线程的运行不受其他处理器(核)上的线程优先级的影响,从而使线程之间的同步和互斥较复杂。如图 3-2 所示为硬件多线程运行状态时空图。

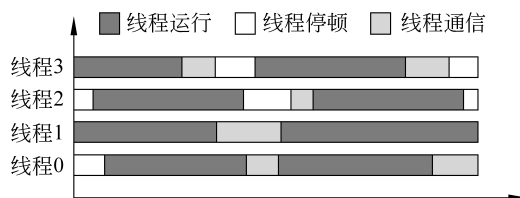


图 3-2 硬件多线程运行状态时空图

对于硬件多线程,处理器任务分配是资源分配的重要内容,它在区分支持多线程的处理器核、单芯多核和多芯多处理器等硬件资源的基础上,既要充分开发程序中的并行性,使各个处理器的负载相对平衡,又要尽量减少线程之间的通信开销。处理器任务分配即是线程和处理器之间的一种映像,它可以由操作系统实现,也可以由应用程序员指定。操作系统的调度算法一般尽可能将线程映像到线程先前所运行的处理器上,这样可以利用可能仍缓存在原处理器 Cache 内的数据,提高 Cache 命中率。但操作系统仅能根据一般规律实现线程映像,不可能根据特殊的硬件分配或特殊的应用特征进行优化。

应用程序的线程映像算法需要相应的优化工具帮助,通过设置线程映像屏蔽向量来指定线程可以运行的处理器,从而实现针对具体应用的优化映像。线程映像屏蔽向量是它所属的进程映像屏蔽向量的一个子集,它的每一位对应一个处理器(核),该位为 1 表示可以在对应的处理器上运行,为 0 则不可以。而多线程并行程序设计面临多线程可伸缩性,它是衡量在性能更加强劲多核处理器上运行时能否有效利用更多线程的指标。例如,一个应用程序是面向 4 核处理器编写的,若该程序运行于 8 核处理器上时,其性能是否能线性增长,可通过该指标衡量。

另外,为了实现负载平衡,不仅需要进行资源分配,还需要进行任务分割,计算任务的分割是资源分配的基础。在任务分割与调度过程中,必须尽量减少分割调度开销、上下文切换开销和线程间的同步开销。

3.1.5 多核同时多线程

1. 同时多线程及其资源利用状态

同时多线程(simultaneous multi threading, SMT)是对多线程的一种改进,它使用多发射和动态调度机制在开发线程级并行的同时开发指令级并行,即将线程级并行转换为指令级并行。由于现代多发射处理器的指令级并行度通常较大,单个线程已无法有效地利用这种并行度。而采用寄存器重命名、保留指令各自的 PC 值、动态调度和为来自多个线程指令的提交提供支持等措施,来自各个独立线程的多条指令可以被同时发射,不必考虑指令间的相关性。相关指令由动态调度来处理,因此线程级并行与指令级并行可同时开发,并且可以在乱序流动处理器的基础上实现。

如图 3-3 所示为不支持多线程超标量处理器的资源利用状态,图中横向表示每个时钟

周期的指令发射能力(宽度即为可以并行执行的指令数亦即流水深度)、纵向表示时钟周期序列、空白一行方格表示对应时钟周期没有指令发射、不同填充方格表示不同线程的指令。可见,在不支持多线程的超标量处理器中,由于缺乏足够的指令并行而限制了多发射槽的利用率,特别地在指令高速缓存不命中时还会出现严重的停顿,将导致整个处理器处于空闲。

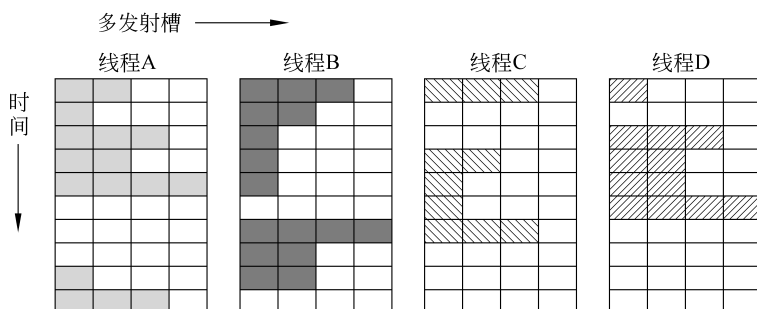


图 3-3 不支持多线程超标量处理器的资源利用状态

如图 3-4 所示为支持粗粒度多线程、支持细粒度多线程、支持同时多线程的超标量处理器的资源利用状态。对于支持粗粒度多线程的超标量处理器,通过线程切换隐藏了长时间停顿带来的开销,提高了资源的利用率,由此减少了完全空闲的时钟周期;但由于指令并行度有限,且仅发生停顿时才进行线程切换,新线程还需要启动时间,所以仍然存在一些完全空闲的时钟周期。对于支持细粒度多线程的超标量处理器,线程交替执行消除了完全空闲的时钟周期;但由于在同一时钟周期内,仅可以发射同一线程的指令,加上指令并行度有限,使得很多时钟周期多发射槽的空闲率仍然较大。对于支持同时多线程的超标量处理器,允许同一时钟周期内发射不同线程的指令,理想情况下,多发射槽的利用率仅受限于多个线程对资源的需求和可用资源的不平衡。当然,实际中还受限于其他因素,如活动线程的个数、缓冲区的大小、从多线程并行取出指令的能力、线程间哪些指令组合可以并行发射等。可见,由于一般情况下动态超标量处理器通常采用深度流水,因此采用粗粒度实现的多线程对性能改进不明显,只有采用细粒度或同时多线程实现才有意义。

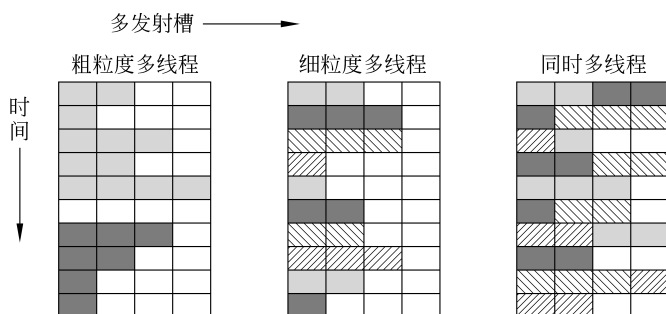


图 3-4 3 种支持多线程超标量处理器的资源利用状态

2. 同时多线程实现的支持技术

当采用细粒度多线程时,为了避免细粒度调度对单个线程性能的影响,可以应用优先线程的方法。优先线程指预取与发射指令具有优先权,只有优先线程停顿或无法发射的情况下,才启动或运行其他线程。但优先线程的数量必须限制,如果有两个优先线程,那么就需要同时预

取两个指令流,这会增加取指单元和指令 Cache 的复杂度。另外,采用优先线程方法时,由于多发射的指令不能来自多个线程,所以若优先线程被停顿,处理器将会损失一些吞吐率。只有足够多的线程混合执行,才能隐藏各种情况下的停顿,使吞吐率最大化。所以,多线程处理器必须在单个线程性能与多个线程性能之间进行综合权衡,从而引出了同时多线程。

同时多线程通过将线程级并行转换为指令级并行,以实现多个线程的指令混合同时执行,但这时仍然需要考虑优先线程,只有当优先线程的缓冲区填满之后,才为其他线程预取指令;另外,为提高单个线程性能,还可以限制多线程的并发数。对于同时多线程,需要许多虚拟寄存器用于为各线程配备更多寄存器组(每个线程均有一个独立的重命名表),以保存线程现场。由于寄存器重命名机制为各寄存器提供了唯一标识,使得多个线程的指令可以在数据路径上混合执行,而不会导致线程之间的源操作数和目的操作数混乱。所以,只要为每个线程设置重命名表、程序计数器,并为多个线程提供指令确认能力,多线程就可以乱序执行。

3. 高性能微处理器体系结构

对于指令级并行,既可以利用流水线技术实现指令并发执行,又可以利用超标量、超长指令字或多核等技术实现指令同时执行。对于线程级并行,既可以利用多线程单核处理器支持并发处理,又可以利用单线程多核处理器支持同时处理。目前,高性能微处理器由开发指令级并行逐渐转向开发线程级并行,其体系结构的变化过程为单核单线程、单核多线程、多核单线程和多核多线程,如图 3-5 所示。

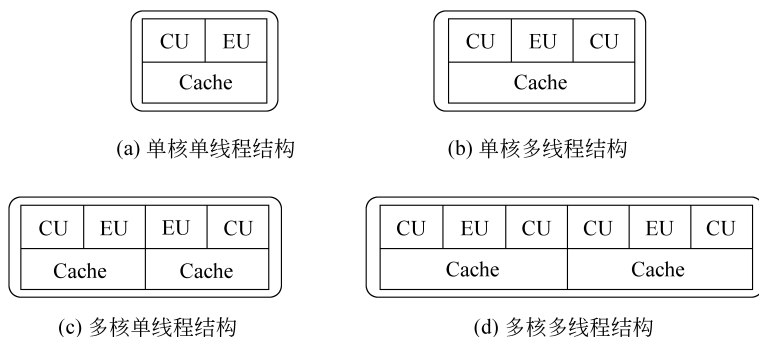


图 3-5 高性能微处理器体系结构变化过程

特别地,在 Pentium 系列微处理器中,Pentium 属于单核单线程,Pentium 4 属于单核多线程,Pentium D 属于多核单线程,Pentium EE 属于多核多线程。另外,目前的微处理器都是 3 核、4 核、6 核和 8 核的,面向服务器和工作站的有 10 核和 12 核的,如 Intel 的 Xeon E5。

3.1.6 典型多核微处理器——T1

1. T1 多核微处理器的组织结构

T1 是 SUN 公司于 2005 年推出的多核微处理器,其建构策略为:重点开发线程级并行性,而不像大多数的处理器是开发指令级并行;主要用于桌面计算或服务器。T1 多处理器同时采用多线程和多核技术的单发射微处理器,以全面提高吞吐率。

T1 微处理器包含 8 个处理器核,每个核为一条 6 个功能段的单发射流水线,最多支持 4 个线程,且单发射流水线中有一个功能段用于线程切换,所以实际类似于 5 个功能段,其体系结构如图 3-6 所示。T1 微处理器采用细粒度多线程,每一个时钟周期均可以切换到新

线程,且线程调度时可以跳过因流水线时延或 Cache 不命中而处于等待状态的线程,仅当 4 个线程均处于等待或停顿时,才会出现空闲状态。它的 load 和分支指令的时延为 3 个时钟周期,但它们都可以通过执行其他线程而被隐藏。特别地,由于浮点运算并不是 T1 微处理器的关注点,因此 8 个核共享一个浮点运算部件。

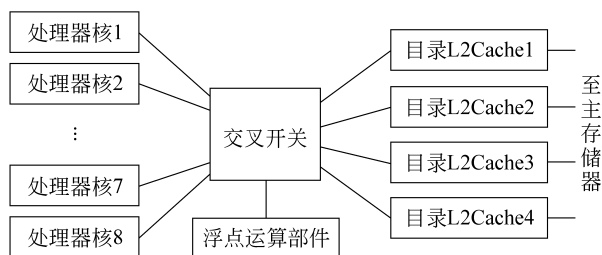


图 3-6 T1 多核微处理器的组织结构

T1 微处理器每个核中均带有第一级高速缓存,高速缓存容量为 16KB 指令高速缓存和 8KB 数据高速缓存,块容量为 64B,在无竞争情况下,L1 不命中开销为 23 个时钟周期。8 个处理器核通过一个交叉开关与 4 个独立的第二级(L2)高速缓存及浮点运算部件相连,每个 L2 高速缓存容量为 750KB 且和主存储器相连,块容量为 64B,在无竞争情况下,L2 不命中的开销为 110 个时钟周期。L2 高速缓存采用目录表法实现高速缓存数据的一致性,且每个 L2 高速缓存块在目录表中都有对应的一项。目录表法类似于目录协议法,目录协议中的目录表用于记录哪些高速缓存中有主存块的副本,而目录表法用于记录哪些 L1 高速缓存中有 L2 高速缓存块的副本。通过把每个 L2 高速缓存与一个具体的主存储器相关联,并强制实现包含性,使得可以把目录表放在 L2 级上,而不是放到主存中,有效地减少访问目录的开销。由于 L1 数据高速缓存采用写直达法,所以实现一致性仅需要写作废消息,且所访问的数据最终都可以从 L2 高速缓存中获得。

2. T1 多核微处理器的性能

利用 3 个面向服务器的基准测试程序:TPC-C、SPECJBB、SPECweb99,对 T1 微处理器进行性能测试,各测试程序对应的每线程 CPI、每核 CPI 及有效 IPC(每个时钟周期完成指令数)等性能指标如表 3-1 所示。特别地,由于 SPECweb99 不能扩展到 8 个处理器核上以利用 32 个线程,所以是在 T1 的 4 核版本上运行的,TPC-C 和 SPECJBB 都是在 8 核 32 个线程的版本上运行的。由于 T1 是一个细粒度多线程处理器,且每核可以支持 4 个线程,所以在并行性足够多时,理想的每线程 CPI 是 4,即每个线程是每 4 个时钟周期完成一条指令,每核 CPI 为 1。这是因为在每 4 个时钟周期中,4 个周期平均分配给 4 个线程,这样每个线程仅能执行一个时钟周期。而有效 $IPC=8/\text{每核 CPI}$ 。

表 3-1 T1 微处理器的测试性能

基准测试程序	每线程 CPI	每核 CPI	8 核有效 CPI	8 核有效 IPC
TPC-C	7.2	1.80	0.225	4.4
SPECJBB	5.6	1.40	0.175	5.7
SPECweb99	6.6	1.65	0.206	4.8

对于 3 个测试基准程序, T1 微处理器的有效吞吐率仅有理想值的 56%~71%。似乎不太有效, 但与超标量处理器相比, 已经提升很多。像 Itanium 2(与 T1 相比, 晶体管更多、功耗更高、硅面积相当)处理器, 若想持续达到每个时钟周期完成 4.5~5.7 条指令, 则需要难以置信的指令吞吐率, 因为 4.5~5.7 已经是比普遍认可的 IPC 高出一倍多。显然, 对于面向整型运算的多线程的服务器, 多核处理器比单核多发射超标量处理器要好得多。

3. 多核微处理器的比较

在近期推出的处理器中, 与采用多发射双核的 Power 5、Opteron、Pentium D 多核微处理器相比, T1 多核微处理器独具特色, 除开发并行性级别(ILP 与 TLP)存在不同之外, 还存在许多差异, 如表 3-2 所示。

表 3-2 多核微处理器的比较

属性或性能	SUN T1	AMD Opteron	Intel Pentium D	IBM Power 5
核数量	8	2	2	2
每核每时钟发射指令数	1	3	3	4
多线程	细粒度	No	SMT	SMT
Cache L1 I/D KB per core L2 per core/shared L3(off-chip)	16/8 3MB shared	64/64 1MB/core	12k uops/16 1MB/core	64/32 L2: 1.9MB Shared L3: 36MB
存储器峰值带宽(GB/s)	34.4	8.6	4.3	17.2
峰值 MIPS/MFLOPS	9600/1200	7200/4800	9600/6400	7600/7600
时钟频率/GHz	1.2	2.4	3.2	1.9

(1) 对浮点运算支持的差异。Power 5 重点关注浮点运算及其性能, Opteron 和 Pentium D 也有许多浮点运算资源, 但 T1 并不关注, 所以其 MFLOPS 仅为 1200, 远小于其他 3 种多核处理器。

(2) 多处理器扩展能力的差异。Power 5 扩展能力最强, Opteron 和 Pentium D 具有一定的扩展支持, T1 不支持扩展。

(3) 存储器峰值带宽的差异。T1 存储器的峰值带宽最高, Power 5 其次, Opteron 和 Pentium D 较低。

对于基于线程级并行的 T1 多核处理器, 若由足够多的线程使其所有核均处于忙碌状态, 且所有线程的性能均可以接受, 那么 T1 具有显著的优势。但线程级并行是否具有绝对性还不能确定; 因为在桌面计算机和服务器中, 单线程性能仍然极其重要。

3.2 机群多处理机

3.2.1 机群多处理机及其性能特点

1. 机群多处理机及其结构模型

由于高性能微处理器的出现和局域网带宽的不断提高, 为了极大地降低多处理机的造

价,人们基于微处理器技术、网络通信技术和并行编程环境,于20世纪90年代中期便设想利用高性能微处理器和局域网来建构多处理机,由此出现了机群(Cluster)多处理机。所谓机群多处理机指利用高速商品化网络将若干高性能工作站或高档PC连接在一起,并在并行程序设计及可视化人机交互集成开发环境支持下,统一调度、协调处理,实现高效并行处理的多处理机。机群多处理机的结构模型见第2章图2-9。

机群多处理机的节点可以同构,也可以异构;其每个节点一般是一台完整的计算机,拥有本地磁盘和操作系统,可以作为一台单独的计算机资源供用户使用,也可以是对称多处理机。机群中的这些计算节点以松散耦合互连,但对用户是以单一形象呈现的,作为一个单一集中的计算资源来使用。从功能来看,机群中的节点可以分为3种类型:①用于完成计算任务的计算节点,它具有很强的计算能力;②用于完成人机信息交换和运行操作系统的管理节点;③用于完成I/O操作的I/O节点,它即是一台I/O设备。

机群多处理机的互连网络可以是普通商品化网络(如以太网、FDDI、ATM等),也可以是专用定制网络。当采用普通商品化网络时,机群仍不同于局域网。局域网是分布式的多计算机,各台计算机各自独立工作,通过局域网共享资源,对用户不是以单一形象呈现的。机群是分布式的多处理机,节点机之间采用消息传递模型进行数据通信,对用户是一个计算资源,屏蔽了节点和网络的异构性。

2. 机群多处理机的性能特点

新型结构的机群多处理机一经问世就受到广泛关注,其原因在于它同其他多处理机相比,具有许多明显的优势。机群多处理机最核心的优势是:随着微处理器技术、网络通信技术和并行编程环境等的发展,其各方面的性能均可以得到改善和提高。例如,由于松散耦合带来数据通信的瓶颈问题,因为高性能通信协议的出现而得到缓解;随着PVM、MPI、HPF、OpenMP等并行编程模型的应用与成熟,使得并行应用程序的开发更加容易方便。机群多处理机还有以下特点。

(1) 开发周期短。由于机群多处理机采用商品化的工作站和局域网来建构,这样既不需要研制计算节点,又不需要研制操作系统和编译软件,开发的重点仅在于数据通信软件和并行编程环境,从而极大地减少了开发研究的时间。

(2) 性价比高。作为巨型或大型计算机的多处理机,生产量不大,价格均比较贵,往往达几百万甚至上千万美元。而机群多处理机采用商品化的节点和互连网,生产量极大,价格很低,机群多处理机的价格比其他多处理机要低1~2个数量级。随着RISC和通信等技术的发展,微处理器和局域网的性能价格比不断提高,使得机群多处理机的性价比还在不断提高。

(3) 可靠性高。机群多处理机的计算节点一般都是兼容的且相互独立工作,所以某计算节点失效不仅不会影响其他节点的工作,而且其计算任务可以迁移到其他节点继续完成,从而可以有效地避免因单节点失效带来整体运行的停顿。

(4) 扩展性强。机群多处理机结构灵活,可以将不同体系结构、不同性能的工作站连在一起组成一个整体,且还是利用商品化网络松散耦合的,所以规模容易扩充、节点便于配置与替换,使得其还具有资源利用率高的优势。

(5) 用户编程方便。机群多处理机的程序并行化只是在原有C、C++或FORTRAN串行程序中插入相应的通信原语,用户仍使用熟悉的编程环境。另外,其可以继承原有的软件

资源,只需对原有的串行程序做有限修改便适用于机群多处理机。

但机群多处理机也存在不足之处。由于机群多处理机的节点是一台完整的计算机,其维护是面向每个节点的,所以维护工作量大、费用较高。对此,目前许多机群多处理机都采用 SMP 为节点,这样便可以减少节点数量。

3.2.2 机群多处理机的分类

由于机群多处理机的分类标准很多,所以按照不同标准,机群处理机的分类方法很多。例如,按照机群节点是否相同,机群可以分为同构的与异构的;按照机群节点是 PC 还是工作站,机群可以分为 PC 机群与工作站机群。但常用分类方法是按照应用目的和构建目的来分类。

1. 按照应用目的分类

按照应用目的,机群多处理机可以分为高可用性机群、负载均衡机群和高性能计算机群等 3 类。

(1) 高可用性机群。高可用性机群的应用目的是在某些节点出现故障时,仍可以继续对外提供服务。该机群采用冗余机制,当某节点由于软硬件故障而失效时,该节点上的任务将在最短的时间内被迁移到另一个具有相同功能与结构的节点上继续执行,这样对于用户来说,机群一直为其提供服务。高可用性机群适用于 Web 服务器、医学监测仪、银行 POS 系统等要求持续提供服务的应用。

(2) 负载均衡机群。负载均衡机群的应用目的是提供与节点数量成正比的负载能力,可以根据各节点的负载状态实时地分配计算任务。为此,该机群专门设置一个监控节点,负责监控每个计算节点的负载和状态,并根据监控结果将任务分派到不同的节点上。负载均衡机群适用于大规模网络应用(如 Web 或 FTP 服务器)、大工作量的串行或批量处理作业(如数据分析)。

(3) 高性能计算机群。高性能计算机群的应用目的是降低高性能计算的 cost,实现仅有超级计算机才能完成的计算任务。该机群通过高速的商用互连网络,将成百乃至上千台 PC 或工作站连接在一起,可以提供接近甚至超过许多并行计算机的计算能力,但其价格却仅是许多并行计算机的几十分之一。高性能计算机群适用于计算量巨大的并行应用,如石油矿藏定位、气象变化模拟、基因序列分析等。

当然,负载均衡机群和高性能计算机群均必须具有一定高可用性特点,以使服务提供稳定。负载均衡机群提供的是静态数据服务,如 HTTP 服务;而高可用性机群既提供静态数据服务,又提供动态数据服务,如数据库等。由于高可用性机群各节点共享同一存储介质,每个服务的用户数据仅一份,存放于专门的存储节点上,在任一时刻仅有一个节点可以读写这份数据,所以它还可以进行动态数据服务。

2. 按照构建目的分类

按照构建目的,机群多处理机可以分为专用机群和企业机群两类。

(1) 专用机群。专用机群为代替传统的大中型机或巨型机而构建,其吞吐率较高,响应时间较短,通信网络频带宽时延小。该机群的特点有:装置比较紧凑,一般都装比较小的机架内,放在机房中使用;节点是同构的,节点中的硬件和软件配置均相同;采用集中控制,由一个(或一组)管理员统一管理;内部通信对外界是屏蔽的,用户一般通过一台终端机来

访问。

(2) 企业机群。企业机群是为充分利用各节点的空闲资源而构建的,各节点一般通过标准的 LAN 或 WAN 互连,通信开销较大、时延较长。该机群的特点有:装置比较松散,节点分散安放,可以不在同一个房间或同一幢楼中;节点是异构的,由不同的个人拥有,节点间的互操作极为重要;采用分散控制,机群管理者仅对各节点进行有限的管理,节点拥有者可以随意地进行关机、重新配置或升级,且对某节点而言,其拥有者任务具有最高优先级,比企业的其他用户均要高;内部通信对外界是暴露的,存在一定的安全隐患,需要在通信软件中采用专门措施来避免。

3.2.3 机群多处理机的软件组织

1. 机群操作系统与单系统映像

由于机群多处理机的组织结构松散、节点独立性强、网络通信连接复杂,造成机群多处理机资源管理不便、操作使用困难。为了实现单一形象呈现,在各节点单机操作系统之上再建立一层操作系统,以管理和呈现所有的计算资源,这就是机群操作系统或机群中间件。可见,软件是机群多处理机系统的重要组成部分。

机群操作系统除需要提供硬件管理、资源共享及网络通信等功能外,还必须实现单系统映像(single system image, SSI),使机群多处理机在使用、控制、管理和维护上如同一个单独的计算机资源,这也是机群多处理机系统的重要特征。单系统映像包含 4 重含义。

(1) “单一系统”。尽管系统包含多台可以单独使用的计算机,但用户看起来是一台计算机。

(2) “单一控制”。逻辑上用户(含系统用户)使用的服务均来自机群中唯一一个位置,如用户将批量处理作业提交到一个作业集,系统管理员则通过一个唯一的控制点配置管理机群的所有软硬件组件。

(3) “对称性”。用户可以从机群的任何一个节点上获得服务,即对于所有节点和所有用户,除具有特定访问权限的服务与功能外,其他所有服务与功能都是对称的,可以通过任何一个节点提供给用户。

(4) “位置透明”。用户不必了解真正提供服务的节点及其具体位置。

2. 单系统映像的功能服务

一般来说,机群操作系统中的单系统映像除提供单一地址空间、单一虚拟网络和单一用户界面外,还至少应提供以下服务。

(1) 单一登录。用户可以通过机群中的任何一个节点登录,且在整个作业处理过程中仅需登录一次,不必因作业被分派到其他节点处理而重新登录。

(2) 单一文件系统。对所有节点都相同的软件,即使执行并行作业时,要求每个节点都可以访问到这些软件,也没有必要在每个节点上重复安装,它们在机群中应只有一个备份。

(3) 单一作业管理。用户可以透明地从任一节点提交作业,但作业可以以批量处理、交互或并行的方式被调度执行。

(4) 单一进程空间。每个进程可以在同一节点或不同的节点声称自进程与任意远程节点的其他进程交换数据信息。

另外,单系统映像至少还应具有两项功能。一是检查点设置,检查点机制使进程状态和

中间结果得以定期保存,当节点失效是故障节点的进程时,可以在另一个正在运行的节点上重新开始,而不会丢失计算结果。二是进程迁移,以使机群各节点的负载达到动态平衡。

3. 机群多处理机软件的结构模型

机群多处理机软件主要包含微内核操作系统、网络通信软件、机群中间件和并行编程环境等,其结构模型如图 3-7 所示。微内核操作系统应具有多用户多任务和稳定性的特征,机群中间件则由机群操作系统(含单系统映像)与机群正常工作所必需的软件组成。对于网络通信软件,其传输协议应具有组播服务及其组管理、快速建立与拆除连接、优先级管理、快速突发控制和选择性突发等功能,具有高吞吐率、低通信时延的特征。

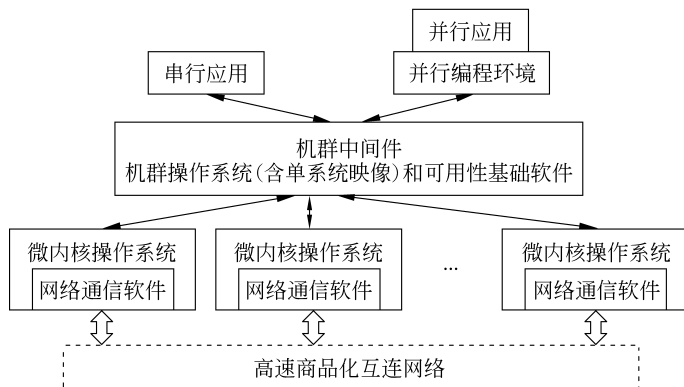


图 3-7 机群多处理机软件的结构模型

并行编程模型与并行编程环境是机群多处理机系统中不可缺少的软件,用户可以通过并行编程环境实现并行应用程序的开发,串行应用则通过机群中间件被调度到任意节点上执行。目前比较流行的并行编程工具包括 MPI、PVM、HPF、OpenMP 等。MPI 是目前最重要的一款基于消息传递的并行编程工具,它具有可移植性好、功能强大、效率高等优点,而且有许多免费、高效、实用的版本,几乎所有的并行计算机厂商都提供对它的支持,使它成为并行编程的事实标准。PVM 也是一款常用的基于消息传递的并行编程环境,它把工作站网络构建成一个虚拟的并行机系统,为并行应用程序提供运行平台。HPF 是一款支持数据并行的并行语言标准。OpenMP 是一款共享存储并行系统上的应用编程接口,规范了一系列的编译制导、运行库例程和环境变量,并为 C、C++ 和 FORTRAN 等高级语言提供了应用编程接口,已经应用于 UNIX、Windows 等多种平台。

3.2.4 机群多处理机的关键技术

对于多处理机,人们期望其节点运算速度高,系统加速比可以随规模扩大而接近线性增长,并行应用程序开发高效、方便。目前,机群多处理机的节点采用高性能工作站或高档 PC,节点运算速度基本可以满足期望。因此,机群多处理机研究的主要目的是提高并行效率和操作使用方便,这便需要高效网络通信技术、负载平衡技术、并程序序设计环境、并程序序调试技术、可扩展性技术、可用性技术等的支持。

1. 高效网络通信技术

机群多处理机一般采用 TCP/IP 的通用局域网以松散耦合的方式实现互连,从而使通信频宽较低、协议处理开销较大、传输时延较长、额外开销较多(如多层协议缓冲复

杂、操作系统的额外开销),因此需要发展高效网络通信技术,以改善提高机群多处理机通信性能。

(1) 发展新型高速局域网络。由于大规模并行计算、实时网络系统和多媒体应用对高速网络的需求,推动了网络技术的发展,从而出现了多种新型的高速网络,如快速以太网和光纤分布式数据接口 FDDI(异步传输 ATM、Myrinet),且快速以太网的通信频宽可达 10Gb/s。

(2) 设计新通信协议。高速网络的运用,使影响通信性能的瓶颈从网络硬件转移到通信软件上,过长的通信协议处理开销使高速网络的高性能得不到充分利用。例如,在物理链路双向频宽为 640Mb/s 的 Myrinet 上,TCP/IP 点-点通信的频宽仅 38Mb/s,说明 TCP/IP 限制了高速网络的链路频宽的利用率。

2. 负载均衡技术

负载均衡性能的好坏会直接影响并行计算的性能,而有效地管理利用所有资源是实现负载均衡的基本途径,所以负载均衡技术实质是全局资源管理利用技术。对于机群多处理机,因节点的异构性、特殊的网络结构、交互用户的介入、后台进程的运行等导致节点性能的动态变化,其负载均衡显得尤其重要。常用的并行编程环境对资源管理利用支持均比较弱,仅提供统一的虚拟机,而节点操作系统是单机的,并不提供全局服务支持,同时也缺少有效的全局共享方法。因此,必须在节点操作系统和并行编程环境之间加入中间件——机群操作系统,以实现所有资源的分配调度和单系统映像。

负载均衡技术的核心是调度算法,即将各个小的计算任务比较均衡地分配到不同的计算节点进行并行处理,以使各节点的利用率达到最大。但负载均衡技术还需要考虑如决策时机、调度模式、负载均衡指标及其收集、负载调度策略等要素。负载均衡评价指标一般有吞吐率、可扩展性和容错性 3 个,容错性指发生故障后任务恢复运行的能力,可扩展性指规模增大或总负载变化时的适应能力,吞吐率指并行运行应用程序的响应时间。

负载调度策略分为静态调度和动态调度两种。

静态调度是在并行优化编译时,根据用户程序中的各种信息(如各个任务的计算量大小、依赖关系和通信关系等)和并行资源的配置状况(如网络拓扑结构、计算节点的计算能力等)对用户程序中的并行任务做出分配方案,在程序运行过程中,按照该分配方案实施任务分配。理论证明,静态调度的最优分配方案是 NP 完全问题,因此在实际中往往采用次优求解算法,以降低算法复杂性。静态调度依赖于任务分配所依据的全信息,但在高度并行的多处理机上,特别是在多用户下,节点的任务负载是动态产生的,不可能作出准确的预测。因此,静态调度对于动态变化的任务负载的负载均衡是不准确的。

动态调度是在应用程序运行过程中,通过分析实时负载信息,对用户程序中的并行任务做出分配,以实现负载均衡的。由于各节点上的计算任务是动态产生的,因此在用户程序运行期间,某节点上负载可能突发性地增加或相对变少,这时重载节点应及时把多余的任务调整到轻载节点上,或由轻载节点及时向重载节点申请任务。动态调度算法简单,可以实时控制平衡负载,但增加了额外开销,因此减小额外开销是动态调度特别关注的问题。

3. 并行程序设计环境

开发并行应用程序要比串行程序困难得多,要涉及多个处理器之间的数据交换与同步,还需要解决数据划分、任务分配、程序调试和性能评测等问题,这些工作需要相应的支持工

具。广义的并行程序设计环境包括硬件平台、操作系统、并行程序设计语言、编程编译软件、调试与性能分析工具等,一般并行程序设计环境至少应该包括并行程序设计语言、并行程序编程编译平台等。机群多处理机的并行性具有两个特点:一是并行进程通常是节点操作系统的进程或线程,所以一般采用调用节点操作系统来创建、消亡及激活进程;二是节点相对独立,通信开销较大,所以批量数据传送一般采用异步通信、进程间协同采用同步通信,适用于中粗粒度任务的并行处理。并行程序设计环境必须与机群多处理机的并行性特点相适应。

并行程序设计语言需要提供数据发送与接收及进程同步的函数,为并行程序的数据传递和进程同步提供支持。例如,PVM 编程平台为用户提供了开发并行程序的 C 语言和 FORTRAN 语言的并行函数库,用户开发并行程序时,仅需要在描述并行特性时加入并行函数,在过去开发的程序中嵌入并行函数,就可以使其在机群多处理机上并行运行。

常用并行编程模型有消息传递和共享存储两种。基于消息传递模型的机群多处理机广泛应用的并行程序编程编译平台,目前主要有 PVM、MPI、Express、Linda 等,其中 PVM 和 MPI 应用最多,且均提供了统一的虚拟机、定义和描述通信与资源管理的可移植的用户编程接口。PVM 是美国 Oak Ridge 国家实验室和多所大学联合开发的并行计算工具软件,支持 C、C++ 和 FORTRAN, MPI 则是一个消息传递标准, PVM 和 MPI 都是免费软件,均可以方便地进行再开发。对于分布共享存储的机群多处理机,应该采用共享变量模型来进行并行编程,相应的并行程序编程编译平台主要有 ThreadMarks DSM、Midway DSM 等。特别地,非共享机群多处理机,也可以采用共享变量的并行编程模型,但需要虚拟共享存储技术的支持。

4. 并行程序调试技术

机群多处理机实现的是线程、进程和作业的并行,高级别的并行处理存在不确定性、死锁、消息错序、全局状态复杂和性能优化等问题。由于并行任务的派生、数据通信、进程同步等的影响,在并行程序不同的运行中,各个进程被执行的时间可能不相同,从而导致并行程序的行为结果(含错误出现)具有不确定性。对于基于消息传递模型的并行程序,即使数据通信可以可靠地实现,死锁和消息错序也是不可能避免的。死锁大多是由于并行程序算法不当造成的,消息错序是由于数据通信的异步性及其时延的随机性造成的。程序运行过程取决于过程状态,而并行程序运行过程非常复杂,其过程的全局状态自然非常复杂,所以只有理解并行程序运行的行为特征、控制其运行过程,才能确定其所有状态及其相互之间转换关系。影响程序并行性能的因素很多,但通信是最关键的,若进程因通信被阻塞,则程序运行效率将会显著降低;合理地划分数据和安排通信进程,可避免阻塞或使阻塞时间减少,为此必须收集每次数据通信的起止时间和通信量,分析造成通信进程等待的原因。显然,高级别并行处理存在的问题,必须通过反复运行调试并行程序才可能得以解决,这便需要以下并行程序调试技术及其相应的方便用户理解和操作调试工具软件的支持。

(1) 记录重放技术。记录重放的目的是解决不确定性问题,保证反复调试的顺利进行,重放方式有历史重放和执行重放。

(2) 事件驱动技术。事件驱动可以对并行程序的运行自动地进行复杂性判断,并通过设置断点来检查其行为特征。事件驱动的关键是对并行程序中的事件进行描述、过滤和识别,但并行程序的事件模型是难以准确建立的。

(3) 分析技术。分析技术的主要目的是寻找并行程序中数据通信方面存在的问题。根

据采用手段,分析技术可分为静态分析和动态分析;根据分析时间,动态分析又可分为在线分析和事后分析。

(4) 可视化技术。可视化技术指采用图形、图像、动画等可视化形式描述并程序的行为、状态、结构等。在机群多处理机中,全局状态和通信行为不仅复杂且可视性差,采用可视化技术直观地将它们呈现出来,有利于并程序调试。

5. 可扩展性技术

可扩展性是多处理机性能评价的主要指标之一,可以反映规模扩大对多处理机性能的影响程度。为使机群多处理机具有可扩展性,必须遵循以下3个原则来构建其体系结构。

(1) 独立性原则。独立性原则要求机群多处理机各节点的软硬件相互独立,即使难以实现该要求,也应使节点间的关联程度尽量小且关联清晰。由此,才可能使独立扩展(增量扩展)得以实现,使异构扩展成为可能。

(2) 平衡性原则。平衡性原则要求最小化所有的性能瓶颈,因为机群多处理机任何一个慢速部件都将会导致整体性能下降,其他部件的速度再快也是无用的。此外,应尽量避免单点失效,一个部件的失效将可能引起整体崩溃。

(3) 时延隐藏原则。时延隐藏指利用计算过程来隐藏通信时延,以保证即使在长时延不可避免时,机群多处理机也可以达到高性能。时延隐藏基本思想之一是使计算和通信在时间上重叠,具体的技术方法有:预取技术、分布式一致性高速缓存、非严格存储一致性模型、多线程处理器。

6. 可用性技术

由2.5.2节可用性定义可知,提高可用性的基本途径为增加MTTF和减少MTTR,如今工作站的MTTF可以达到几百甚至几千小时,再进一步提高MTTF已非常困难且开销很大。机群多处理机的MTTF往往低于节点的MTTF,减少MTTR不仅可以提高机群的可用性,还可以迅速处理故障。适用于机群的可用性技术主要有3个。

(1) 部件冗余且相对独立。改善设备可用性的基本方法是部件冗余,当某主部件发生故障时,由备用部件继续提供服务。当然,主要部件和备用部件之间必须相互独立,使它们不会因为某一原因而同时发生故障。

(2) 故障接管。对于商用机群多处理机,故障接管是最重要的性能需求。某部件发生故障时,故障部件提供的服务允许由正常部件接管继续提供。

(3) 恢复。恢复指为接管故障部件的工作负载所需要的操作,且分为后向和前向两种恢复技术。对于后向恢复,为运行进程在稳定的存储设备中周期地保存其一致状态(即检查点)。发生故障后,通过重组来隔离故障部件,恢复前一个检查点,而后继续正常操作,整个过程称为卷回。独立于应用程序的可移植的后向恢复较容易实现,并已被广泛运用。卷回过程的时间开销较大,在强实时性中是不能容忍的,这时便有人提出了前向恢复技术。前向恢复不是卷回到故障前的某个检查点,而是利用故障诊断信息去重构一个有效状态,并继续运行下去。前向恢复依赖于应用程序且可能还需要额外的硬件设备支持。

3.2.5 典型机群多处理机实例

1. IBM SP2 机群多处理机

IBM SP2(简称SP2)是机群多处理机中的代表性产品,它既可用于科学计算,又可供商

业应用。在 1997 年的“人机大战”中,战胜世界国际象棋冠军卡斯帕罗夫(Garry kasparor)的“深蓝”,就是一台采用 30 个 RS/6000 工作站(带有专门设计的 480 片国际象棋芯片)的 IBM SP2 机群多处理机。SP2 机群采用异步 MIMD、分布式存储物理结构模型,其体系结构如图 3-8 所示,节点硬件和软件都可以根据不同用户应用和环境需要来进行配置,且节点数可以为 2~512 个不等。由于 SP2 机群采用标准的工作站部件,仅在标准技术不能满足性能要求时才使用专用软件和硬件,所以其开发周期较短。

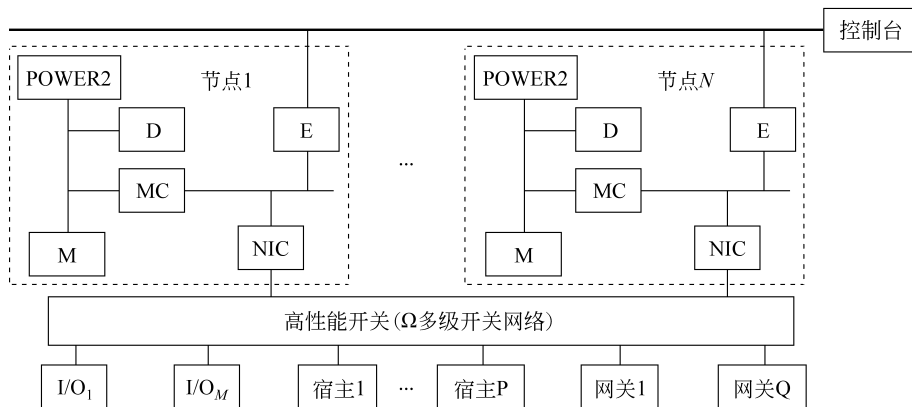


图 3-8 SP2 机群多处理机的体系结构

1) 节点及其微处理器

SP2 机群多处理机的节点可以有 4 种配置,分别是宿主节点、I/O 节点、网关节点和计算节点,它们可以重叠。例如,宿主节点可以作为计算节点,I/O 节点可以作为网关节点。宿主节点(含控制台)用于处理用户注册会话和交互处理,I/O 节点用于实现输入输出如全局文件服务器,网关节点用于连接外部计算机(如外部服务器、附加文件服务器、网络路由器、可视设备等),而计算节点用于计算。其中有一宿主节点为专用的控制台,它配置的是一台 RS/6000 工作站,通过该控制台节点,管理人员可以对机群进行管理。特别地,每个节点、开关和机架上均集成了一个监视板,该监视板用于对硬件部件进行环境检测、控制,管理人员可以用于启动或切断电源和为单点置初始状态。

SP2 机群的计算节点均是一台带私有存储器(M)和本地磁盘(D)的 RS/6000 工作站,RS/6000 采用时钟频率为 66.7MHz 的 POWER2 微处理器,它带有一个 32KB 的指令高速缓存、一个 256KB 的数据高速缓存、两个分支转移控制部件、两个整数运算部件和两个乘加浮点运算部件(乘加运算均可以在一个时钟周期内完成),其峰值速度可达 $4 \times 66.7 = 267\text{MFLOPS}$ 。POWER2 微处理器是 6 发射的超标量处理机,每个时钟周期可以执行 6 条指令,包括 2 条读数写数指令、2 条浮点乘或加指令、1 条变址增量指令和 1 条件转移指令,具有动态分支预测技术和寄存器重命名技术。

为了使节点配置更加灵活,SP2 机群可以配置 3 种不同特性的节点:宽节点、细节点和细 2 节点,区别在于存储器容量、数据通路宽度和 I/O 总线插槽数有所不同。主存容量:宽节点可达 64~2048MB,其他两种节点是 64~512MB;高速缓存容量:细节点和细 2 节点可以有 1MB(或 2MB)的二级高速缓存,宽节点的数据高速缓存为 256KB;存储总线宽度:宽节点是 256 位,细 2 节点是 128 位,细节点则是 64 位;I/O 总线插槽数:宽节点的微通道

MC(micro channel)上有 8 个 I/O 插槽,细节点则仅有 4 个 I/O 插槽。在 SP2 的每个节点中,存储器和高速缓存的容量都比较大,处理器性能亦较高,这使得 SP2 的处理能力能够达到相当高的水平。

2) 互连网络

SP2 机群多处理机的节点之间可以通过两种网络松散耦合互连,这两种网络为标准以太网和定制的 128 路高性能开关(HPS,即 Ω 多级开关网络),且是通过节点本身的 I/O 微通道连接到网络上,而不是通过本身的存储总线。以太网用于对数据通信速度要求不高时的并行程序开发,还可以供机群的监视、引导、加载、测试和其他管理软件使用,并行程序正式运行时则使用 HPS。当然,以太网还具有备份的作用,当 HPS 出现故障时,可以通过以太网来维持正常工作。

SP2 机群的节点通过网络接口电路(NIC)连接到 HPS,通过以太网适配器(E)连接到以太网上,且将网络接口电路称为开关适配器或通信适配器。通信适配器中有一个 8MB 的 DRAM 用来存储各种不同协议所需的大量报文,并由一台 i860 微处理器控制。通信适配器经微通道接口连接到微通道上,经存储开关管理部件 MSMU 连接到 HPS 上。特别地,微通道是 IBM 公司的标准 I/O 总线,用于把 I/O 设备连接到 RS/6000 工作站和 IBM PC 上。对于 SP2 机群,除采用高性能开关外,有的还采用光纤分布式数据接口(fiber distributed data interface,FDDI)环连接各节点。

3) 系统软件

SP2 机群多处理机的系统软件包含标准 AIX 操作系统(IBM 的 UNIX)、高性能服务软件等,其中 AIX 操作系统是核心,每个节点均配置 AIX 操作系统和高性能服务软件,系统软件的层次结构如图 3-9 所示。

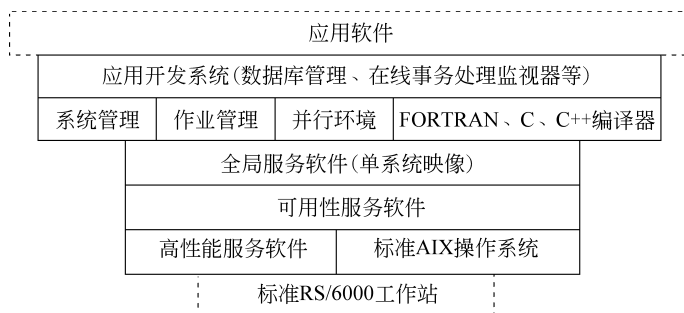


图 3-9 SP2 机群多处理机系统软件的层次结构

SP2 机群除可以运行原在 RS/6000 工作站基于分布式开发的软件外,利用高性能服务软件还提供了许多高性能服务,主要有高性能通信软件、高性能文件系统、并行函数库、并行数据库和并行 I/O 软件等。可用性服务软件用于指示哪些节点在活动、将节点和进程标记归属于某特定集、将失效、停机或重启节点通告节点集成员并可随后调用恢复服务等。全局服务软件即实现单系统映像,并提供多种可选类型。AIX 并行环境(parallel operation environment,PE)为用户提供开发和运行并行政程序的平台,且包含并行操作环境、消息传输库、可视化工具和并行调试器。

SP2 机群在 RS/6000 工作站原有环境下开发的大部分软件均可以重用,包括 1000 多种串行应用程序、数据库管理系统、联机事务处理监控程序、系统与作业管理软件、

FORTRAN/C++ 编译程序、数学与工程程序库等。另外,仅添加一些可扩充并行所必需的软件,或对现有软件进行少量修改,便可以适用于 SP2 机群。

2.3 种不同的机群多处理机

1) Berkeley NOW 机群多处理机

Berkeley NOW 机群多处理机是由美国加州大学伯克利分校开发的、颇具影响的多处理机,它采用了许多先进技术,涉及许多机群共性问题。它具有很多优点:采用商用千兆以太网和主动消息通信协议;通过用户级整合机群软件 GLUNIX 来提供单系统映像、资源管理和可用性服务;开发了一种新的无服务器网络文件系统 xFS,以支持可扩充性和单一文件层次的高可用性。

主动消息通信协议。主动消息是实现低开销通信的一种异步通信机制,其基本思想是在消息头部控制信息中携带一个用户级子例程(称为消息处理程序)的地址,当消息头到达目的节点时,调用消息处理程序通过网络获取剩下的数据,并把它们集成到正在进行的计算中。主动消息高效灵活,以至于各种多处理机均逐渐以它作为基本的通信机制。

GLUNIX 机群软件。GLUNIX 是运行于工作站的标准 UNIX 上的一个自包含软件,其主要思想是认为机群操作系统应该由底层和高层组成,其中底层是运行在核模式下的节点商用操作系统,高层则是提供机群所需功能的用户级操作系统。特别地,该软件层可以提供机群内节点的单系统映像,使得所有的处理器、存储器、网络容量和磁盘带宽均可以分配于串行或并行的应用程序,并且以被保护的用户级操作系统库的形式实现。

无服务器文件系统 xFS。xFS 是一个无服务器的分布式文件系统,它将文件服务的功能分布到机群的所有节点上,以提供低时延高带宽的文件系统服务。xFS 主要采用廉价冗余磁盘阵列、协同文件缓存和分布式管理等技术。

2) Beowulf 机群多处理机

1994 年,美国国家航空航天局(national aeronautics and space administration,NASA)的一个科研项目迫切需要一种工作站,要求它既具有 1GFLOPS 的计算处理能力和 10GB 的存储容量,又不能价格过高。为了实现该要求,工作于 CESDIS 的 Thomas Sterling 与 Don Becker 二人便构建了一个具有 16 个节点的机群多处理机,其硬件使用 Intel 的 DX4 处理器及 10Mb/s 的以太网,软件则主要基于刚诞生的 Linux 系统和 GNU 开发环境等软件,并将其命名为 Beowulf 机群,基于 COTS(commodity off the shelf)思想的技术也迅速被传播。

可以认为,Beowulf 机群定义了构建机群的一种策略,即通过有效使用普通硬件加上 Linux 操作系统、GNU 开发环境及 PVM/MPI 共享库来实现。由此,不仅集中了那些能力相对较弱的计算资源,以高性能价格比提供相当于大型机的性能,还可以保证软件环境的稳定性。实际上,Beowulf 并不只是具体的软件包或是一种新的网络拓扑结构,更重要的是 Beowulf 机群提出了构建多处理机的基本原则,即在实现既定目标的前提下,把注意力集中在获取高性能价格比上。虽然目前为了获取更高性能,有些 Beowulf 机群也使用了一些专用或商用的软件和特殊的网络互连系统,但其基本宗旨是不变的。

3) LAMP 机群多处理机

随着硬件技术的不断进步,SMP 多处理机不仅成本不断下降,内部通信能力也在不断加强,由小规模(2~8 个处理器)的 SMP 来构建机群逐渐成为主流,且称之为 CLUMP