CN43-1258/TN 计算机工程与科学 1999年第 21卷第 4期
ISSN 1007-130X COMPUTER ENGINEERING& SCIENCE Vol. 21, No. 4, 1999

多线程技术的现状与前景展望*

Current Status and Prospect of Multithreading

周兴铭 徐 明 肖 刚

Zhou Xingming, Xu Ming *and* Xiao Gang

(并行与分布处理国家重点实验室)

(**National Laboratory for Parallel and Distributed Processing**)

摘 要 这几年,多线程计算机得到了较大的发展,也得到了越来越多的关注。本文在分析了多线程概念以及多线程计算机发展历史后,对多线程计算机的研究和开发现状、技术途径与发展前景进行了较为系统的介绍,并对其中的关键技术问题展开了探讨。

ABSTRACT These several years have seen great progress in the technology of multithreading. In this paper, we analyze the concept of multithreading, introduce the history of multithreaded computers, and discuss their current status as well as their future prospect. Also, we shed some light on the key issues in multithreaded computer design.

关键词 多线程,多线程计算机,实现

KEY WORDS multithreading, multithreaded computer, implementation.

1 引言

近几年来,多线程技术引起了计算机界特别是体系结构设计和研究人员的高度关注。多线程技术与三超(超标量、超流水和超长指令字)技术开发指令级并行性(ILP)的目的不同,它致力于线程级的并行性(TLP)的开发。美国最近筹划的千万亿次计算机(Petaflops)计划亦将多线程体系结构作为其中的一项关键技术。从我们1998年春赴美访问考察获得的情况来看,国外的一些著名大学在多线程计算机的研究上已有相当深入的工作,MIT等院校已经有了数年乃至十余年的积累。1997年秋天,商品化的多线程计算机问世,目前,第一台双处理机MTA系统已安装在美国San Diego超级计算机中心。多线程计算机在保持一致的存储访问模式的同时,从根本上解决了困扰高性能计算机多时的时延问题,因此,其发展将对未来计算机的体系结构和系统软件的设计产生重要而深远的影响。对于这种发展趋势,应给予足够的重视。我们将以一组文章从多线程概念、多

* 收稿日期: 1999年 1月 12日。
作者简介: 周兴铭,男,1938年 12月生,中国科学院院士,教授,博士生导师,研究领域为高性能计算机、分布与并行处理技术。
通讯地址: 410073湖南省长沙市砚瓦池正街 47号并行与分布处理国家重点实验室
Address Nat'l Lab for Parallel& Distributed Processing, 47 Yanwachi St, Changsha, Hunan 410073, P. R. China

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.

线程计算机发展、多线程体系结构、系统软件的设计和实现技术等方面对这一领域的研究和发展现状进行较为全面的介绍，并对关键技术展开重点探讨。

2 多线程的概念、发展历史与分类

多线程并不是一个全新的概念，它是数据流计算机研究的继续。传统上，计算模型被分成数据流和控制流两种模型，而多线程计算模型实际上是这两种模型的综合。换句话说，多线程模型是数据流模型和控制流模型这两极向中间发展的结果。

对线程尚没有一个严格而统一的定义。从程序的角度来看，线程（thread）是一段顺序指令序列。线程分为就绪态、运行态和挂起态，与进程状态空间相同。一旦某个线程产生一个长延时操作，如访存、处理机间通讯或长浮点运算等，该线程即被挂起，随即由调度器从线程池中选择一个就绪线程进入 CPU。这样，时延被隐藏起来了。线程作为 CPU 调度的基本单位，子线程共享父线程的资源。进程可看作是由线程组成的，一个含有多线程的进程中，多个线程共享同一地址空间，所不同的只是私有栈和代码栈。线程的创建和调度成本大大低于进程，所以，线程有时也被称作轻进程（light process）。

多线程的思想最早在 B. Smith 设计的 Denelcor HEP机器中率先采用，更早一点的历史可以追溯到六十年代 CDC6600机器中的多功能部件计分牌。一般认为，多线程的发展遵循两条途径：一是在冯诺依曼控制流结构上扩充对多线程并发的支持；二是在数据流结构上扩充顺序执行的线程。这种结构被称为混合结构（Dataflow /Von Neumann Hybrid）。MIT研制的多线程系统如 P-RISC、Empire和* T等等大多属于这种类型。从体系结构分类上看，多线程计算机属于多指令流多数据流计算机（MIMD）。

多线程内部按控制流驱动方式顺序执行，而线程间则是按数据流方式点火、调度和执行。因此，在多线程内部和多线程间都能开发大量的并行性，最重要的是一些长延时操作对系统性能的影响基本消失。多线程与其他减少时延的方案相比，优势十分明显。

描述多线程模型的两个因素分别是线程特性和调度算法。线程特性包括线程粒度、线程相关性、线程原子性以及线程的产生方法等；而调度算法则涉及线程状态关系、调度机制以及线程执行粒度等。根据线程的调度或执行规模，可以将其划分为粗粒度多线程和细粒度多线程。粒度越小，越接近于数据流。特别是当一个进程就是一个线程时，多线程模型就成为纯粹的控制流模型；当每个线程缩小到只包含一个操作时，多线程模型便回归到了经典的数据流模型。

3 多线程计算机

3.1 多线程计算机的发展

传统高性能计算机在发展过程中总是不能很好地协调高性能和高可用性。为了追求高性能，系统的规模越来越大。但是当系统的规模变大以后，通讯时延成为不能不解决却又得不到很好解决的难题。多线程技术通过隐藏时延，提供完全用户级软件兼容，以此来达到整体性能和效率的提高，为高性能和高可用性这一对矛盾的解决展现了希望。因

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://w

此，这几年形成了多线程计算机研究与开发的高潮，一些公司开始推出商品化的多线程计算机，更多的大学和实验室积极从事各种类型多线程计算机的研究。多线程计算机的发展情况如图 1所示。

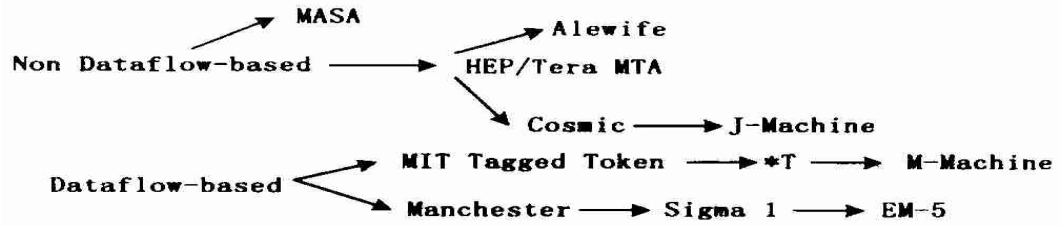

图 1 多线程计算机的发展轨迹

3.2 商品化多线程计算机

B. Smith设计的 Denelcor HEP包括 16台处理机，每台处理机可同时执行 128个线程。由于没有合适的软件 (主要是编译器) 的支持，HEP最后在商业上并没有取得成功。

Tera MTA[1]作为 HEP的后代，采用 GaAs及高密度组装技术，最大配置是 256台处理机，512个存储处理部件，256个 I/O部件，通过 16× 16× 16三维 torus网络互连各个部件之间 64位流水包方式交换信息，主机还可通过 Hippy高速网络与外部工作站联系。Tera MTA的系统以线程为中心而不是以处理机或存储器为中心来设计。每个处理机有 128套硬件设备，每套设备中有 32个私有寄存器，一个程序计数器 (PC)，以保证能同时执行 128个线程。线程之间切换的时间极短，仅需一个时钟周期 (3ns)。相比之下，Tera MTA的一次访存操作需要 140~ 160个时钟周期，这可能也与 Tera MTA没有 Cache有关。由于时延被隐藏，只要有足够多的线程，MTA处理机就会以峰值速度 (0.5 ~ 1Gflops) 运行。事实上，对多线程计算机而言，是程序本身的并行性而不是数据的位置决定处理机的利用率。

Tera MTA的操作系统是一个全对称的分布式 Unix(BSD 4.2)，Tera公司自己开发了一个并发微内核，可支持 15个分立的地址空间，每个空间中的线程数可多可少。Tera MTA操作系统提供了单一的系统映象，并且由于线性的、一致的强共享存储器 (UMA: Uniform Memory Access)设计，用户觉得 Tera MTA用起来象是一种 SMP结构的机器。

对于新型体系结构的机器来说，有两个重要的问题需要解决：一是如何做到软件兼容，即如何实现各种应用软件的平滑移植；二是能够有效支持传统结构机器上不能很好支持的应用。一般认为，迄今为止的高性能机 (甚至包括过去的向量机) 对非规则、稀疏和动态特征的应用问题不能提供很好的支持，而多线程计算机对这些应用有明显的优势。这个结论已有较为充分的验证。软件兼容的一个重要条件就是要有一个高效的编译器将串行代码分解成可以并发的线程，并且最好能够获得更佳的实际运行效果。模拟结果表明，对 SPICE、DYNA3D和 NASTRAN等程序，32个处理机的 Tera MTA较 Cray T90/32提高性能 3倍以上。从对 1998年 4月安装在 San Diego超级计算机中心的双处理机 Tera MTA进行测算的结果亦看出，对 NASA给出的典型 Benchmark程序，Tera MTA的性能比同等配置下的 Cray T90要高[1]。

1998年 3月我们在西雅图 Tera公司访问时，见到有两台 Tera MTA正在组装。据

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.

http://w

称，16处理机的 Tera MTA售价达 400万美元。由于采用 GaAs而非 CMOS,且采用水冷方式,估计对其性能价格比将产生不利影响。尽管 Tera MTA已有不少有意义的结果，但能否在商业上成功还不好预料。也许是存在风险性，著名的大公司(如 IBM、HP)在这方面还没有大动作。 Tera MTA系统规模扩大后效果究竟如何，值得注意。

3.3 研究性的多线程计算机

在指令级并行性的开发已经变得十分困难之后，研究者的注意力开始转移到较粗粒度的并行性的开发，同时回过头来重新审视曾风靡一时的数据流计算模型。所以，这两年多线程计算机获得了广泛的研究兴趣和重要发展并不奇怪。目前，对这项技术的研究主要集中在美国、法国以及日本的某些公司研究所，而研究工作则是多层面、多角度的。早些年，MIT在数据流计算机领域的研究对今天多线程计算机产生了很大的并且也是持久的影响。当前,除了 MIT以外，有影响的研究单位还包括美国的华盛顿大学、普渡大学、阿拉巴马大学、宾夕法尼亚大学等。

从技术集成的角度分析，多线程计算机的设计有单纯的多线程体系结构研究，但更多的是将其他体系结构新技术结合到多线程计算机中,如超标量多线程、前瞻性多线程、超流水多线程、超长指令字多线程等等。美国华盛顿大学正在展开同时多线程 (SMT)[2] 项目的研究，其中心思想是扩充现有乱序执行超标量处理机，让多个独立的线程在单拍内送指令到超标量单元，通过资源竞争、动态共享的方式使所有的硬件现场资源同时活跃,将线程级的并行性与指令级的并行性结合起来,最终目的是提高处理机的利用率。而多现场处理机则是在一个处理机之内设置若干个逻辑处理器，逻辑处理器由指令队列和译码单元以及程序计数器组成。每个逻辑处理器对应一个线程现场，通过对线程的动态调度，使得各个线程交替地在逻辑处理器上得到执行。当然，在技术集成上，可以将多种技术综合在一起，例如， Tera MTA实际上是超长指令字多现场的多线程计算机，其指令是由 3段 (3-wide) 构成的超长指令字[1]。

在通用微处理机结构上,从软件角度进行多线程计算机的研究是一种较为常见的、比较便宜的方式。美国特拉华大学的高光荣教授设计的 Earth系统就是采用这种方式[3]。他们用基于 i860的 SMP构成了多线程系统的模拟试验平台，如图 2所示。

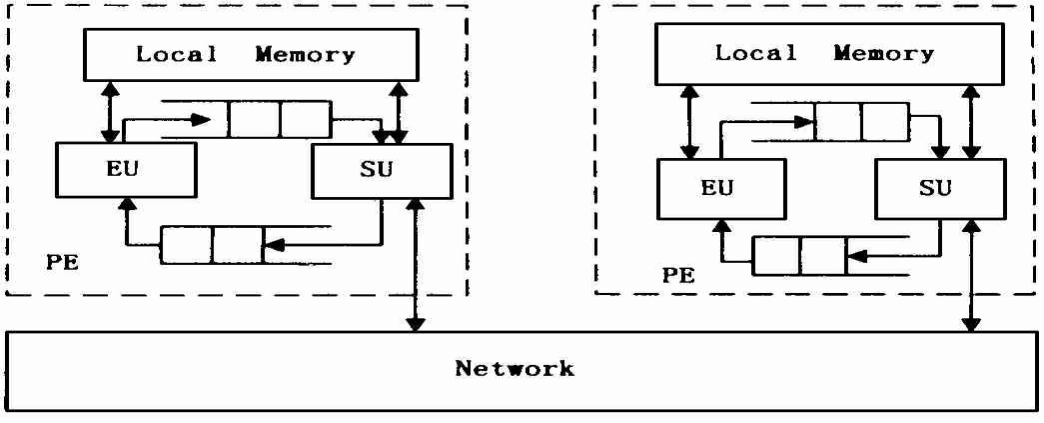

图 2 Earth系统结构

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://w

在每个节点内的两个 i860中, 一个被用作执行单元 (EU), 另一个被用作管理和调度单元 (SU)。 Earth系统的基本思路是研究多线程思想在通用微处理器上的适应性, 用 Earth-C编制的应用程序通过预编译工具对其加注后进行相关性分析并生成线程级代码, 最后通过一个线程编译器 (Threaded C Compiler) 将线程级代码转换到顺序语义的二进制代码执行。 Earth系统还设计了 Runtime子系统, 通过设置一组原语来达到上层与编译器接口, 下层与操作系统连接。 这组原语的功能包括线程的通讯与同步、线程管理以及线程的调度。在 Earth系统中, 线程队列以及线程的现场等都是在存储器中仿真实现的。 这也表明了软件模拟方式另外一个好处, 即灵活性很大。 线程池的规模、线程的大小以及调度策略等都可以灵活地改变从而得到各种参数下的性能改善数据。

4 多线程的软件实现

事实上, 为支持多线程计算模型, 并不一定需要有多线程处理机硬件, 换句话说, 多线程思想也可以在非多线程处理机或 SMP上实现。 这是研究多线程的一个有利条件。 多线程编译系统和多线程操作系统已经成为系统软件发展的主流。 在实现方法上, 大多是结合操作系统的内核、运行库等一起构成一个多线程运行环境。 Mac OS Sun OS LWP POSIX Pthreads以及 Windows NT就是几个典型的多线程操作系统的例子。尽管它们各自提供的系统调用界面有差别, 其实质是一样的, 都是首先利用操作系统原语建立线程文件, 再在此基础上建立一组供应用程序使用的函数调用。从软件的角度来说, Lampson 和 Reball最早在 Mesa语言中提出了线程模型并阐述了建立多线程机制的理论基础[4]。一个多线程模型必须支持线程的创建、删除和同步操作。 从最近几年的发展来看, 线程机制已经成为并发程序设计语言和操作系统的主要特征。 在共享主存多处理机系统的软件设计中更是如此。 例如, C/C++ 提供了用户级的线程库, 可以由用户通过合适的调用来创建和管理线程。 而 Java则是用类来生成和同步线程。 与 Java、ADA-95等支持粗粒度线程不同, MASA上的 MultiLisp作为一种函数式程序设计语言只支持细粒度的、非阻塞的线程。

5 需解决的关键技术问题

多线程技术成功的标志是商品化的多线程计算机的大量出现。 目前的技术离此目标还有一段不小的距离。 今后一段时间需要解决的关键技术问题包括:

(1) 多线程计算机的软件设计

一种体系结构是否有生命力, 关键在于软件在这种结构上的实际执行效果。 在当前的多线程计算机上, 编程需要专门设计的并行程序设计语言, 至少需要对通用程序设计语言进行加注。 为满足程序的调试和分析, 还需要合适的软件开发环境和工具。 如何在操作系统、语言以及硬件上完成功能的划分和协调, 开发定制的微内核, 都是有待解决的难题。

(2) 多线程计算机的规模

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved.

http://w

在理论上，多线程计算机是线性可伸缩的，这是多线程计算机成为未来超高性能计算机主角的基本依据。但目前的多线程计算机系统规模都不大，Tera公司对较大规模MTA的模拟结果尚不足以说明问题。当系统规模扩大之后，系统软件的额外开销增加还没有定量的数据分析方法。网络的性能、编译器以及负载的均衡性将是影响到多线程计算机系统能否线性可伸缩的几个基本因素。

(3) 线程资源的管理和调度

多线程结构比原来的结构更加复杂了，复杂性涵盖了硬件现场规模、现场切换机制、调度算法等软硬件的诸多方面。线程规模大了以后，设计复杂性问题将更加突出，甚至会降低系统实际可得到的性能，但规模太小又不足以体现出多线程体系结构的优点，无法充分地提高资源的利用率。

(4) 可适应的多线程技术

将多线程技术与其他技术有效地结合起来已经做了很多的工作，现在的问题是如何确定多线程技术的定位。在与超标量、超长指令字技术的结合中，一个不足的问题是如何有效开发线程级的并行性。另外，在其他新型体系结构(如单芯片多处理器、PIM等)中，多线程技术能发挥何种作用还值得很好地研究。

除上述问题以外，围绕线程的管理、通讯和同步等也值得作进一步的研究。

6 结束语

由于对多线程计算机的研究还不够深入，可能还有一些深刻的问题待发现，但多线程计算机作为一个新的生长点将会在今后高性能计算机发展中产生重要作用，这一点是毋庸置疑的。

参 考 文 献

1 http // www.tera.com

2 Tullsen D M, et al. Simultaneous Multithreading: Maximizing On-Chip Parallelism. In Proc 22nd Int Symp on Computer Architecture, 1995. 392~403

3 Gao G R. Earth System. In Proc Workshop on Compiler for Supercomputer, Beijing, 1998. 27~30

4 FAQ CompOS Research. 1996

?1994-2015 China Academic Journal Electronic Publishing House. All rights reserved. http://w