

<<多处理器编程的艺术>>

图书基本信息

书名：<<多处理器编程的艺术>>

13位ISBN编号：9787111268055

10位ISBN编号：7111268059

出版时间：2009-8

出版时间：机械工业出版社

作者：Maurice Herlihy, Nir Shavit

页数：356

译者：金海, 胡侃

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

<<多处理器编程的艺术>>

前言

每逢我们在多处理器平台上进行编程时，往往会有这么一种感觉：即使已熟练掌握了系统提供的各种同步原语，但所编制的并行程序的实际性能似乎总有些差强人意，并不十分理想。究其原因，问题的根结在于多处理器编程应是一门科学和艺术完美结合的学科。若要在多处理器系统结构上编制出性能良好的并行程序，要求设计者不仅要精通多处理器系统结构，并行算法以及一些系统构建工具，还应能基于一种设计理念，充分发挥个人的想象空间，合理搭配这些知识和资源，从而和谐地构建完整的系统，使设计者能比底层硬件和操作系统“做得更好”。也就是说，在编写多处理器程序时，要能同时从宏观和微观两种角度分析问题，并能在这两种角度之间灵活地转换。

自20世纪中叶第一台通用电子计算机研制成功以来，程序的编制大多是基于顺序计算模型的，程序的执行过程是操作的有序序列。

由于顺序计算机能够用图灵机精确地描述，因此顺序计算的编程能在一组易于理解且完备定义的抽象之上进行，而不需要了解底层的细节。

近年来，尽管单处理器仍在发展，但由于指令级并行的开发空间正在减少，再加上散热等问题限制了时钟频率的继续提高，所以单处理器发展的速度正在减缓，这最终导致了起源于在单独一个晶片上设计多个内核的多处理器系统结构的出现。

多处理器系统结构允许多个处理器执行同一个程序，共享同一程序的代码和地址空间，并利用并行技术来提高计算效率。

在这种计算模型中，并发程序的执行可以看做是多个并发线程对一组共享对象的操作序列，为了在这种异步并发环境中获得更好的性能，底层系统结构的细节需要呈现给设计者。

作为一名优秀的程序设计员，在编写多处理器程序之前首先应弄清楚多处理器计算机的能力和限制是什么，在异步并发计算模型中什么是可解决的，什么是不可解决的，是什么使得某些问题很难计算，而又使另一些问题容易计算。

这要求设计者具备一定的多核并行计算理论基础知识，掌握多处理器系统结构上并发计算模型的可计算性理论及复杂性理论。

其次，应掌握基本的多核平台上的并行程序设计技术，包括并行算法、同步原语以及各种多核系统结构。

Maurice Herlihy教授和Nir Shavit教授在并发程序设计领域具有很深的造诣，并拥有40年以上一起从事并发程序设计教学的合作经验。

他们对多处理器并行程序设计技术做出了巨大的贡献，并因此而成为2004年ACM / EATCS G6del奖的共同获得者。

这本由他们合著的专著致力于解决如何采用更好的并行算法来克服多核并发程序并行度低的问题。

Amdahl定律早已明确地告诉我们，从程序本身可获得的并行度是有限的，加速比的提高主要取决于程序中必须增加的串行执行部分，而这部分又往往包含着具有相对较高开销的通信和协作。

因此，在多处理器系统结构上，如何提高程序中必须串行部分的并行度，以及降低并行处理器中远程访问的时延是我们目前面临的两大技术挑战。

对这些问题的有效解决，必须依靠软件技术和硬件技术的改进和发展。

本书则侧重于对前一个挑战的研究。

作者先从宏观的抽象角度出发，在一个理想化的共享存储器系统结构中研究各种并行算法的可计算性及正确性。

通过对这些经典算法的推理分析，向读者揭示了现代协作范例和并发数据结构中所隐藏的核心思想，使读者学会如何分析饥饿和死锁等微妙的活性问题，深层次地研究现代硬件同步原语所应具有的能力及其特性。

随后，从微观的实际角度出发，针对当今主流的多处理器系统结构，设计了一系列完美高效的并行算法及并发数据结构，并对各种算法的效率及其机理进行了分析。

所有的设计全部采用Java程序设计语言详细地描述，可以非常容易地将它们扩展到实际应用中。

<<多处理器编程的艺术>>

本书的前6章讲述了多处理器程序设计的原理部分，着重于异步并发环境中的可计算性问题，借助于一个理想化的计算模型来阐述如何描述和证明并行程序的实际执行行为。由于其自身的特点，多处理器程序的正确性要比顺序执行程序的正确性复杂得多，书中为我们展现了一系列不同的辅助论证工具，令人有耳目一新之感。

随后的11章阐述了多处理器程序设计的实践部分。由于在多处理器环境中编写程序时，底层系统结构的细节并不像编写顺序程序那样被完全隐藏在一种编程抽象中，因此，本书在附录B介绍了多处理器硬件的基础知识。

最后的第18章介绍了当今并发问题研究中最先进的事务方法，可以预言这种方法在今后的研究中将会越来越重要。

<<多处理器编程的艺术>>

内容概要

《多处理器编程的艺术》从原理和实践两个方面全面阐述了多处理器编程的指导原则，包含编制高效的多处理器程序所必备的算法技术。

此外，附录提供了采用其他程序设计语言包（如C#、C及C++的PThreads库）进行编程的相关背景知识以及硬件基础知识。

《多处理器编程的艺术》适合作为高等院校计算机及相关专业高年级本科生及研究生的教材，同时也可作为相关技术人员的参考书。

目前，多处理器的编程技术受到广泛关注，多处理器编程要求理解新型计算原理、算法及编程工具；至今很少有人能够精通这门编程艺术。

现今，大多数工程技术人员都是通过艰辛的反复实践、求助有经验的朋友来学习多处理器编程技巧。

这本最新的权威著作致力于改变这种状况，作者全面阐述了多处理器编程的指导原则，介绍了编制高效的多处理器程序所必备的算法技术。

《多处理器编程的艺术》所涵盖的多处理器编程关键问题将使在校学生以及相关技术人员受益匪浅。

<<多处理器编程的艺术>>

作者简介

Maurice Herlihy，哈佛大学的数学学士和麻省理工学院的计算机科学博士，目前为美国布朗大学计算机科学系教授，曾工作于卡内基-梅隆大学和DEC剑桥实验室。他是美国ACM会士，2003年分布式计算Dijkstra奖获得者。

<<多处理器编程的艺术>>

书籍目录

出版者的话译者序前言第1章 引言1.1 共享对象和同步1.2 生活实例1.3 生产者—消费者问题1.4 读者—写者问题1.5 并行的困境1.6 并行程序设计1.7 本章注释1.8 习题第一部分 原理第2章 互斥2.1 时间2.2 临界区2.3 双线程解决方案2.4 过滤锁2.5 公平性2.6 Bakery算法2.7 有界时间戳2.8 存储单元数量的下界2.9 本章注释2.10 习题第3章 并发对象3.1 并发性与正确性3.2 顺序对象3.3 静态一致性3.4 顺序一致性3.5 可线性化性3.6 形式化定义3.7 演进条件3.8 Java存储器模型3.9 评析3.10 本章注释3.11 习题第4章 共享存储器基础4.1 寄存器空间4.2 寄存器构造4.3 原子快照4.4 本章注释4.5 习题第5章 同步原子操作的相对能力5.1 一致数5.2 原子寄存器5.3 一致性协议5.4 FIFO队列5.5 多重赋值对象5.6 读—改—写操作5.7 Common2RMW操作5.8 compareAndSet()操作5.9 本章注释5.10 习题第6章 一致性的通用性6.1 引言6.2 通用性6.3 一种通用的无锁构造6.4 一种通用的无等待构造6.5 本章注释6.6 习题第二部分 实践第7章 自旋锁与争用7.1 实际问题7.2 测试—设置锁7.3 再论基于TAS的自旋锁7.4 指数后退7.5 队列锁7.6 时限队列锁7.7 复合锁7.8 层次锁7.9 由一个锁管理所有的锁7.10 本章注释7.11 习题第8章 管程和阻塞同步8.1 引言8.2 管程锁和条件8.3 读者—写者锁8.4 我们的可重入锁8.5 信号量8.6 本章注释8.7 习题第9章 链表：锁的作用9.1 引言9.2 基于链表的集合9.3 并发推理9.4 粗粒度同步9.5 细粒度同步9.6 乐观同步9.7 惰性同步9.8 非阻塞同步9.9 讨论9.10 本章注释9.11 习题第10章 并行队列和ABA问题10.1 引言10.2 队列10.3 部分有界队列10.4 完全无界队列10.5 无锁的无界队列10.6 内存回收和ABA问题10.7 双重数据结构10.8 本章注释10.9 习题第11章 并发栈和消除11.1 引言11.2 无锁的无界栈11.3 消除11.4 后退消除栈11.5 本章注释11.6 习题第12章 计数、排序和分布式协作12.1 引言12.2 共享计数12.3 软件组合12.4 静态一致池和计数器12.5 计数网12.6 衍射树12.7 并行排序12.8 排序网12.9 样本排序12.10 分布式协作12.11 本章注释12.12 习题第13章 并发哈希和固有并行13.1 引言13.2 封闭地址哈希集13.3 无锁哈希集13.4 开放地址哈希集13.5 本章注释13.6 习题第14章 跳表和平衡查找14.1 引言14.2 顺序跳表14.3 基于锁的并发跳表14.4 无锁并发跳表14.5 并发跳表14.6 本章注释14.7 习题第15章 优先级队列15.1 引言15.2 基于数组的有界优先级队列15.3 基于树的有界优先级队列15.4 基于堆的无界优先级队列15.5 基于跳表的无界优先级队列15.6 本章注释15.7 习题第16章 异步执行、调度和工作分配16.1 引言16.2 并行分析16.3 多处理器的实际调度16.4 工作分配16.5 工作窃取双端队列16.6 本章注释16.7 习题第17章 障碍17.1 引言17.2 障碍实现17.3 语义换向障碍17.4 组合树障碍17.5 静态树障碍17.6 终止检测障碍17.7 本章注释17.8 习题第18章 事务内存18.1 引言18.2 事务和原子性18.3 软事务内存18.4 硬事务内存18.5 本章注释18.6 习题第三部分 附录附录A软件基础附录B硬件基础参考文献

<<多处理器编程的艺术>>

章节摘录

第1章 引言 计算机产业正在经历着一场重大的结构重组和巨变，在没有其他变革之前，这个过程无疑将会继续进行。

主要的芯片制造商，至少是现在，都纷纷放弃尝试研制速度更快的处理器。

虽然摩尔定律仍旧适用：每年集成在同样大小空间中的晶体管数越来越多，然而，由于散热问题难于解决，它们的时钟速度无法继续得到提高。

取而代之的做法是，制造商开始转向“多核”系统结构的研制，由多个处理器（多核）共享硬件高速缓存直接进行通信。

通过将多个处理器同时分配给单一任务以获得更高的并行性，从而提高计算的效率。

多处理器系统结构的普及对计算机软件的发展带来了深刻的影响。

直至今日以前，技术的进步意味着时钟速度的提升，时钟本身的加速导致了软件执行效率的提高。

今天，这种搭便车的现象已不复存在，技术的进步不再指时钟速度的提升而是指并行度的提升，并行问题已经成为现代计算机科学的主要挑战。

本书着重讲述共享存储器通信方式下的多处理器编程技术。

通常称这样的系统为共享存储器的多处理器，现在称之为多核。

在各种规模的多处理器系统中都存在着不同的编程挑战——对于小规模的系统来说，需要协调单个芯片内的多个处理器对同一个共享存储单元的访问，对于大规模系统来说，需要协调一台超级计算机中各个处理器之间的数据路由。

其次，现代计算机系统所固有的异步特征也给多处理器编程带来了挑战：在没有任何警示的情形下，系统的活动可以被各种不同的事件中中止或延迟，例如中断、抢占、cache缺失和系统故障等。

这种延迟现象本身是无法预测的，时延的长短也是不确定的：cache缺失可以造成不到十条指令执行时间的时延，页故障可能造成几百万条指令执行时间的时延，而操作系统抢占则会导致多达上亿条指令执行时间的时延。

<<多处理器编程的艺术>>

编辑推荐

目前，多处理器的编程技术受到广泛关注，多处理器编程要求理解新型计算原理、算法及编程工具；至今很少有人能够精通这门编程艺术。

现今，大多数工程技术人员都是通过艰辛的反复实践、求助有经验的朋友来学习多处理器编程技巧。

这本最新的权威著作致力于改变这种状况，作者全面阐述了多处理器编程的指导原则，介绍了编制高效的多处理器程序所必备的算法技术。

本书所涵盖的多处理器编程关键问题将使在校学生以及相关技术人员受益匪浅。

本书特色 · 循序渐进地讲述共享存储器多线程编程的基础知识。

· 详细解释当今多处理器硬件对并发程序设计的支持方式。

· 全面考察主流的并发数据结构及其关键设计要素。

· 从简单的锁机制到最新的事务内存系统，独立、完整地阐述了同步技术。

· 利用Java并发工具包编写的可完全执行的Java实例。

· 附录提供了采用其他程序设计语言和包（如C#、C及C++的PThreads库）进行编程的相关背景知识以及硬件基础知识。

<<多处理器编程的艺术>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>