

# 多核系统效率与任务属性关系的优化策略

石 玮<sup>1</sup>,李雨前<sup>2</sup>,汪自云<sup>1</sup>

(1.湖北师范学院 计算机科学与技术学院,湖北 黄石 435002;

2.华中科技大学 计算机学院,湖北 武汉 430074)

**摘要:**通过对多核应用技术进行系统效率与任务属性关系的优化分析,得出系统所接受任务的不同属性与对应的系统效率之间的指令级仿真结果,证明了系统面向的任务属性与选择的多核系统性能存在科学的性价比,从而保证了期望的系统计算力与整体架构之间符合原本的科学与经济性。

**关键词:**多核系统效率;任务属性;指令级仿真

中图分类号:TP39

文献标识码:A

## Optimization of relationship between multi-core system efficiency and mission property

SHI Wei<sup>1</sup>, LI Yu Qian<sup>2</sup>, WANG Zi Yun<sup>1</sup>

(1.College of Computer Science and Technology, Hubei Normal University, Huangshi 435002, China;

2. College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

**Abstract:** Through the optimization analysis of the relationship between system efficiency and the mission property on multi-core applied technology, the instruction-level simulation results about the different property and the corresponding efficiency is obtained. It proves that the mission property and the selection of multi-core system performance exist scientific cost-performance, and ensures in line with anticipant scientific and economy between system calculate power and system structure.

**Key words:** multi-core system efficiency; mission property; instruction-level simulation

多核 CPU 支持的紧耦合并行结构对面向移动计算或高性能计算机(HPC)系统的技术支持是时下业界研究的热点。然而,并行体系结构设计与应用表明:系统的并行处理能力与系统的整体效率不一定成正比。对于确定的任务,计算量增加,系统效率可能下降<sup>[1]</sup>。所以,在引入多核结构提升系统并行处理能力过程中,充分考虑系统所处理任务的属性是获得高系统效率的关键因素之一。

人们在追求计算机高速度运行、高可靠性的今天,更加注重系统效率<sup>[2-3]</sup>。尤其是摩尔定律遇到瓶颈时,驱动了多核 CPU 设计,同时基于多核的并行程序设计也随之成为研究热点,多核并行硬件和软件协调进步仍然延续着 ENIAC 以来的相辅相成发展的技术路线,成为新概念上的研究领域。

对计算机性能准确建模非常复杂<sup>[4]</sup>,其中最基本的因素包括存储器层次结构、操作系统、互连网络、处理机技术、高速缓存与存储管理、延迟包容或吸收机制、算法设计与编程语言等。然而,这些技术细节仅仅源自计算机系统本身,而随着非科学计算的处理任务日趋显现(如流媒体处理、模式识别、图像处理、知识发现、多媒体库操作等),传统的并行处理机制与结构所追求的并行能力指标将不再适应新属性任务处理,研究任务属性与系统效率或整体性能,将成为并行处理体系结构设计与应用的重要课题。

### 1 并行系统的并行能力与效率描述

在研究并行处理技术过程中,因为 Von Neumann 机的存储程序结构及系统的整体处理能力,取决于系统的全部处理节点或多核访问内存的机制与效率<sup>[2]</sup>。所

## 技术与方法 Technique and Method

以,当代典型的并行机系统都重点研究访问内存的技术与方法。

### 1.1 几种典型的并行机系统

当前典型的并行机系统有共享存储的对称多处理机 SMP(Symmetric Multi-Processor)、大规模并行处理机 MPP(Massively Parallel Processor)、分布式共享存储器多处理机 DSM(Distributed Shared Memory)、工作站机群 COW(Cluster of Workstations)和跨地域性的、用高速网络将异构性计算节点连接起来满足用户分布式计算要求的网络计算环境 GCE(Grid Computational Environment)。

### 1.2 并行计算机访存模型

均匀存储访问模型(Uniform Memory Access),其重要特征是物理存储器被所有处理器均匀共享,所有处理器访问任何存储字的时间相同;每台处理器都带私有高速缓存,外围设备也可以一定形式共享。也称为紧耦合系统(Tightly Coupled System)。当所有处理器都能等地访问所有 I/O 设备、能同样地运行程序(如操作系统内核、I/O 服务程序等)时,称为对称多处理机(SMP)。

非均匀存储访问模型(Non-uniform Memory Access),所共享的存储器在物理上是分布在所有的处理机中,其所有本地存储器的集合就组成了全局地址空间。处理器访问存储器的时间不一样,访问本地存储器 LM 或群内共享存储器 CSM 较快,而访问外地存储器或全局共享存储器 GSM(Global Share Memory)较慢。每台处理器照例可带私有高速缓存,外设也可以某种形式共享。

全高速缓存存储访问模型(Cache-Only Memory Access),各处理器节点中没有存储层次结构,全部高速缓存组成了全局地址空间。利用分布的高速缓存目录 D 进行远程高速缓存的访问,缓存容量一般都大于二级高速缓存容量。数据开始可任意分配,随着进程的推进,数据最终被迁移到相应存储空间。

高速缓存一致性非均匀存储访问模型(Cache-Coherent Non-uniform Memory Access),绝大多数商用系统都使用基于目录的高速缓存一致性协议,比较 SMP 优化了可扩展性,是一种分布共享存储的多处理机系统。随着进程推进,数据自迁移到所用的空间。

非远程存储访问模型(No-Remote Memory Access),所有存储器是私有的,不支持远程存储器访问。

### 1.3 系统加速比与效率

多处理器加速比和处理机效率可分别表示为:

$$\text{多处理器加速比 } (S_p) = \frac{\text{单个处理器的运行时间}}{\text{多个处理器的运行时间}}$$

$$\text{处理机效率 } (E) = \frac{\text{加速比 } (S_p)}{\text{处理机数 } (P)} \quad (1)$$

因为最佳加速比是线性的,即:  $S_p = O_p$ , 所以最佳效率就是常数,即:  $E = \text{Const}$  ( $0 \leq \text{Const} \leq 1$ )。

由(1)式知,  $P$  增加,则  $E$  下降,但能够通过优化算法使  $E$  增加。当然是增加了算法优化过程的工作量

(Workload),即由于付出 Workload 开销,在系统处理机数量增加时维持系统效率不变。这就是本文提出的基于任务属性分析结果,科学地选择多核系统结构的概念。所以,始终保持一定效率常数的优化系统,应该能够实时对系统内参与运行的处理机数量实现科学调度,这等同于大规模作战系统的指挥,必须科学地调兵遣将。

## 2 指令级并行与多核 CPU

实现指令级并行(ILP)处理的基本要求是被执行指令序列不存在指令与数据相关,系统能在同一绝对时间或相对时间内并行执行多个任务指令或线程,现代多核 CPU 能支持片内多线程平行推进。如果任务的指令序列存在相关性,平行推进过程将出现“参差不齐”或线程暂停而阻塞相关核的运行线程现象。所以,需要事先找出指令代码中合适的指令序列段(S),如果执行 S 的时钟周期能正好等于原来被阻塞的延迟时间(Delay),则能有效地缓冲或吸收线程阻塞,继续维持多核的多线程平行推进。

现代多核 CPU 实际上引入了多线程平行推进过程中自适应进程迁移技术,即当某核的线程被阻塞时,能自动完成相应线程上的进程段迁移,相当于上述执行 S 而吸收线程阻塞。

如上分析,指令级并行度与多核 CPU 支持的系统效率紧密关联。如何利用软硬件技术去最大限度地开发处理机中的指令级并行性,关键在于要知道何时及如何改变指令顺序。在实际运用中,这种改变过程必须由编译器或硬件正确实现。显而易见,多核系统如果不能保证多核多线程平行推进,则更多的核可能更影响多核系统的整体效率,这成为目前对多核 CPU 结构研究的重点。

## 3 不同属性的任务对多核 CPU 处理能力与效率的影响分析

应用系统对计算力或计算机的处理能力的需求是可以标定的。尤其是对于给定的科学计算问题,所需要的处理时间几乎能预估。而面向非科学计算问题的处理,处理系统要开销的时间往往不可预知。比如,在一个大型网络数据库系统内完成相应的知识发现,需要开销的时间可能各不相同,因待发现问题的算法(约束方程)而异。产生“不同”的原因其实与发现问题(处理任务)的属性相关,如果任务(Task)是可细粒度划分的,则由多核 CPU 支持的并行系统处理效率高。由于细粒度划分算法的工作开销(Workload),保证了多核 CPU 的多线程能平行推进,实现高的并行效率。

考虑一种理想状态,被处理任务可划分为均匀的四大模块,且并行系统由四核 CPU 支持,宏观上生成四条平行的流水线,由于每条流水线上的指令序列不存在任何相关性,则四核的指令级线程将平行推进,处理效率最高。如果 Task 客观上只能分成均匀的两大模块

## 技术与方法 Technique and Method

(分成均匀的四块将出现相关), 且同样在四核 CPU 系统上运行, 其系统效率将可以如下计算:

设四核处理均匀四模块的时间为  $T_4$ , 则两核处理均匀两模块的时间为  $2T_4$ 。

如果由四核处理均匀两模块任务序列, 且指令级并行过程中存在的相关性产生的线程阻塞花费的吸收开销为  $T_p$ , 当且仅当  $T_p \leq 2T_4$  时, 该并行系统才是功能上与四核处理四模块等价的, 但性价比则大大降低。

因此, 明晰应用系统的属性更能指导并行系统的选择或构建, 科学地分析系统整体效率或系统结构选择, 在并行系统不断发展进程中应该具有重要意义。

### 4 任务属性与系统效率实际仿真实例

本文采用 VC6.0 编程对多核并行处理进行仿真, 仿真结果如图 1 和图 2 所示。

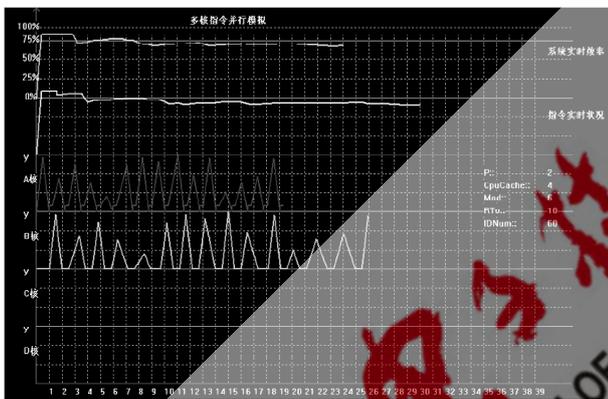


图 1 四核 CPU 并行处理均匀两模块任务仿真图

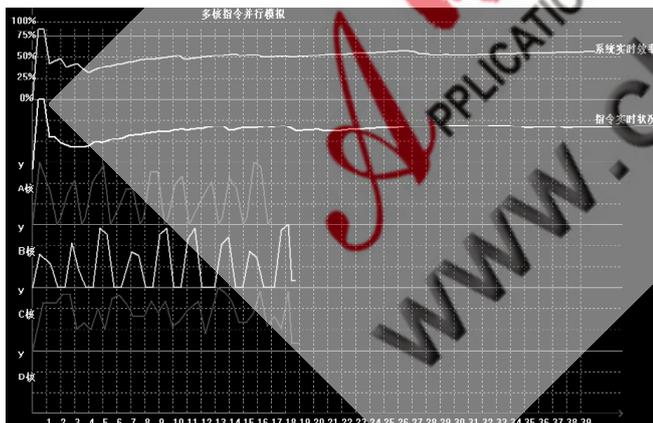


图 2 四核 CPU 处理均匀四模块任务仿真视图

比较图 1、图 2, 可以得到以下结论:

动态效率(图中第二条曲线)始终低于静态效率(理想效率)。动态效率是多核改变原指令执行顺序, 同时受核自身的空间、当前任务量影响。

静态效率(图中第一条曲线)是任务在随机生成时, 静态顺序执行, 运用 Amdahl 定律计算出来的。第二曲线低于第一曲线原因分析如下:

(1) 动态执行中多核之间协同发生同步等待延时,

系统效率下降;

(2) 某时刻处理单元空闲, 产生等待延时, 系统效率下降;

(3) 改变指令原有的执行顺序, 引起效率变化。

第二曲线比第一曲线长是因为产生延时等待。当系统中核的负载均衡, 即线程可平行推进时, 动态效率更接近静态的理想效率, 这与实际非常吻合, 说明了基于 Amdahl 定律计算系统效率是可行的。

当系统中有一个或多个核长时间空闲时, 则整个系统效率明显下降。

当任务分配不均匀, 指令相关产生的等待或吸收, 则动态效率非常不稳定。

以上结论基于最初的指令级抽象、分配、执行策略, 但由于是静态调度, Krste Asanovic 的工作<sup>[5]</sup>指出了其 4 个主要的缺点:

(1) 不可预料的转移;

(2) 可变的内存延迟(无法预料的 cache 不命中);

(3) 代码大小的爆炸;

(4) 编译器的复杂性。

所以, 仿真证明了任务属性与多核 CPU 支持的并行系统效率之间存在的紧密关系, 是指导提高应用系统性价比的重要因素。

多核处理器是处理器发展的必然趋势。无论是移动或嵌入式应用、桌面应用还是服务器应用, 都将采用多核的架构<sup>[6]</sup>。

多核处理器要想发挥出威力, 关键在于并行化软件支持, 多核设计带动并行化计算的推进, 而给软件带来的影响更是革命性的<sup>[7]</sup>。面对多核系统, 需要有并行编程的思想才有可能充分利用资源, 而人类的思维模型习惯于线性思维, 对“面”或者更为复杂的立体编程模式, 效率会下降很多。

仿真结果证明了多核 CPU 支持的并行系统效率提升与确定系统所面向的任务属性至关重要, 它将有效地指导业界的应用系统优化设计。

### 参考文献

- [1] 郭云芳. 计算机仿真技术[M]. 北京: 北京航空航天大学出版社, 1991: 1-2.
- [2] 康继昌. 现代并行计算机原理[M]. 北京: 西北工业大学出版社, 1997.
- [3] 陈树清. 并行计算机的现状与发展趋势[J]. 计算机世界, 1997, (15).
- [4] JOROAN F, GITA A. 并行处理基本原理[M]. 迟利华, 刘杰, 译. 北京: 清华大学出版社, 2004: 35-310.
- [5] ASANOVIC K. VLIW/EPIC: 静态调度的指令级并行. <http://www.core.org.cn/>, 2008.
- [6] 汪东升. 多核引领计算机架构质变[J]. 计算机世界, 2008, (40): 43-45.
- [7] 姜波. 多核引发变革: 服务器软件的并行化革命. <http://net.yesky.com/>, 2008.

(收稿日期: 2009-03-18)