《科学计算与企业级应用的并行优化》

书籍信息

版 次：1
页 数：
字 数：
印刷时间：2015年07月01日
开 本：16开
纸 张：胶版纸
包 装：平装
是否套装：否
国际标准书号ISBN：9787111506287
丛书名：高性能计算技术丛书

内容简介

本书系统、深入讲解了科学计算及企业级应用的并行优化方法与*实践。第1章介绍了常见的并行编程基于的多核/众核向量处理器架构。第2章介绍了如何在X86、ARM和GPU上优化常见的线性代数运算。第3章介绍了如何在X86和GPU处理器上优化偏微分方程的求解。第4章介绍了如何在X86处理器和GPU上优化常见的分子动力学算法。第5章详细介绍了如何在X86、ARM和GPU上优化常见的机器学习算法。

目 录

显示全部信息

前 言

T行业急需这本书
和本系列的前两本书一样，在解释为什么笔者认为软件工程师需要这本书之前，笔者先来介绍并行、并发和代码性能优化这3个概念，因为理解这3个概念是阅读本系列3本书的基础。
并行对应的英文单词是parallelism，是指在具有多个处理单元的系统上，通过将计算或数据划分为多个部分，将各个部分分配到不同的处理单元上，各处理单元相互协作，同时运行，以达到加快求解速度或者提高求解问题规模的目的。
并发对应的英文单词是concurrency，是指在一个处理单元上运行多个应用，各应用分时占用处理单元，是一种微观上串行、宏观上并行的模式，有时也称其为时间上串行、空间上并行。
代码性能优化是指通过调整源代码，使得其生成的机器指令能够更高效地执行，通常高效是指执行时间更少、使用的存储器更少或能够计算更大规模的问题。
T行业急需这本书 和本系列的前两本书一样，在解释为什么笔者认为软件工程师需要这本书之前，笔者先来介绍并行、并发和代码性能优化这3个概念，因为理解这3个概念是阅读本系列3本书的基础。 并行对应的英文单词是parallelism，是指在具有多个处理单元的系统上，通过将计算或数据划分为多个部分，将各个部分分配到不同的处理单元上，各处理单元相互协作，同时运行，以达到加快求解速度或者提高求解问题规模的目的。 并发对应的英文单词是concurrency，是指在一个处理单元上运行多个应用，各应用分时占用处理单元，是一种微观上串行、宏观上并行的模式，有时也称其为时间上串行、空间上并行。 代码性能优化是指通过调整源代码，使得其生成的机器指令能够更高效地执行，通常高效是指执行时间更少、使用的存储器更少或能够计算更大规模的问题。 从大的方面来说，并行和并发都是代码性能优化的一种方式，但是今天并行和并发已经是如此重要，以至于需要“ 开宗立派 ”。为了明晰并行、并发和代码性能优化的边界，在本书中，代码性能优化特指除了并行和并发以外的代码优化方法，比如向量化和提高指令流水线效率。在本书中，笔者将向量化独立出来解说。 2003年以前，在摩尔定律的作用下，单核标量处理器的性能持续提升，软件开发人员只需要写好软件，而性能的提升就等待下次硬件更新来解决，在2003年之前的几十年里，这种“ 免费午餐 ”模式一直在持续。2003年后，主要由于功耗的原因，这种“ 免费午餐 ”已经不复存在。为了生存，各硬件生产商不得不采用各种方式提高硬件的计算能力。目前最流行的3种方式如下：
1)让处理器在一个周期处理多条指令，多条指令可相同可不同。如Intel Haswell处理器一个周期可执行4条整数加法指令、两条浮点乘加指令，而访存和运算指令也可同时执行。
2)使用向量指令，主要是SIMD和VLIW技术。SIMD技术将处理器一次能够处理的数据位数从字长扩大到128或256位，也就提升了计算能力。 3)在同一个芯片中集成多个处理单元，根据集成方式的不同，相应地称为多核或多路处理器。多核处理器是如此重要，以至于现在即使是手机上的嵌入式ARM处理器都已经是四核或八核了。 目前绝大部分应用

软件都是串行的，因为串行执行过程符合人类的思维习惯，易于理解、分析和验证。由于串行软件只能在多核CPU中的一个核上运行，和2003年以前的CPU没有多少区别，这意味着花多核CPU的价钱买到了单核的性能。通过多核技术，硬件生产商成功地将提高实际计算能力的任务转嫁给软件开发人员，而软件开发人员没有选择，只有直面挑战。标量单核的计算能力没有办法继续大幅度提升，而应用对硬件计算能力的需求依旧在提升，这是个实实在在的矛盾。在可见的将来，要解决这个矛盾，软件开发人员只有代码性能优化和并行可以选择。代码性能优化并不能利用多核CPU的全部计算能力，它也不要求软件开发人员掌握并行开发技术，另外通常也无需对软件架构做改动，而且串行代码优化有时能够获得非常好的性能(如果原来的代码写得很差的话)，因此相比采用并行技术，应当优先选择串行代码性能优化。一般来说，采用并行技术获得的性能加速不超过核数，这是一个非常大的限制，因为目前CPU硬件生产商最多只能集成十几、几十个核。

从2006年开始，可编程的GPU越来越得到大众的认可。GPU是图形处理单元(Graphics Processing Unit)的简称，最初主要用于图形渲染。自20世纪90年代开始，NVIDIA、AMD(ATI)等GPU生产商对硬件和软件加以改进，GPU的可编程能力不断提高，GPGPU(Generalpurpose computing on graphics processing units)比以前容易许多。另外由于GPU具有比CPU强大的峰值计算能力，近年来引起了许多科研人员和企业的兴趣。 近两三年来，在互联网企业中，GPU和并行计算越来越受到重视。无论是国外的Google、Facebook，还是国内的百度、腾讯、阿里和360，都在使用代码性能优化、并行计算和GPU来完成以前不能完成的任务。 10年前，并行计算还是实验室里教授们的研究对象，而今天多核处理器和GPU的普及已经使得普通人就可以研究它们。对于软件开发人员来说，如果不掌握并行计算和代码性能优化技术，在不久的将来就会被淘汰。 作为本系列的压轴之作，本书专注于领域相关的算法和应用的并行与性能优化。笔者介绍了如何优化线性代数、偏微分方程求解、分子动力学和机器学习领域的一些重要算法。 如果以武侠中的功夫来比喻的话，本系列的第一本书《并行算法设计与性能优化》专注于理论基础和实践的结合，是“ 内功 ”和“ 心法 ”的修炼；本系列的第二本书《并行编程方法与优化实践》专注于程序设计语言的核心内容的应用，是“ 招式 ”的学习；而本书则是“ 内功心法 ”和“ 招式 ”的具体运用。“ 内功心法 ”好意味着基础好、潜力大，以后发展空间广泛；“ 招式 ”好则能够通过“ 奇招 ”“ 怪招 ”解决问题；而只有将“ 内功心法 ”和“ 招式 ”完美融合，做到“ 心中无招而手中有招，无招胜有招，无招即有招 ”，才能成为异构并行计算这个领域真正的集大成者。愿读者和笔者一起向这个目标前进。 本书完全是“ 干货 ”，是一本真正将业界最佳实践和“ 只可意会，不可言传 ”的领域知识简洁明了地贡献出来的著作。为了帮助读者理解，本书使用了大量的示例。开发人员通常比较忙，因此本书力求简洁明了，点到为止。 读者对象 由于多核处理器和GPU已经非常便宜，而代码性能优化、向量化和并行已经深入IT行业的骨髓，所有IT行业的从业者都应当阅读本书。如果非要列一个读者清单，笔者认为下列人员应当阅读本书： 互联网及传统行业的IT从业者，尤其是希望将应用移植到多核向量处理器的软件开发人员；

对向量化和并行化感兴趣的职业工作者；

线性代数、偏微分方程、分子动力学和机器学习相关领域的科技工作者；

大中专院校及研究所的学生、教师； 关注异构并行计算和高性能计算的人们。

如何阅读本书 本系列包括3本书，本书是此系列的第三本。本书重点介绍如何利用目前主流的C语言的各种特定硬件或平台的向量化扩展、并行化库，来设计性能优良的向量

化和并行代码。而本系列的第一本《并行算法设计与代码优化》关注并行优化和并行计算相关的理论、算法设计及高层次的实践经验；本系列第二本《并行编程方法与优化实践》关注C程序设计语言的向量化和并行化扩展及算法到硬件的映射；本书则关注如何将线性代数、偏微分方程求解、分子动力学和机器学习领域的常见算法优良地实现出来。 本书不但包括如何使用SSE/AVX向量化扩展、OpenMP编译制导语句来优化运行在X86多核处理器上的代码性能，还包括使用NEON向量化扩展、OpenMP编译制导语句优化运行在移动处理器(ARM)的代码性能，以及使用CUDA和OpenCL优化运行在图形处理器(GPU)的代码性能。笔者希望通过这种方式能够让阅读本书的软件开发人员了解和掌握如何将常见算法映射到具体硬件上以获得高性能，以及如何依据硬件和算法的特点进行代码性能优化。 本书分为以下几章： 第1章 介绍常见的并行编程基于的多核/众核向量处理器的架构、OpenCL程序如何映射到这些平台上执行及OpenCL程序在这些硬件上运行时具有哪些不同。先介绍Intel Haswell、ARM A15、Intel MIC、AMD GCN GPU和NVIDIA Kepler/Maxwell GPU的架构。然后介绍OpenCL程序如何映射到Intel Haswell处理器、AMD GCN GPU和NVIDIA Kepler/Maxwell
GPU上执行。最后介绍OpenCL程序在这些处理器上运行时的细微区别。 第2章 介绍如何在X86、ARM和GPU上优化常见的线性代数运算，如计算稀疏矩阵向量乘法，求解下三角线性方程组，计算矩阵乘法等。对于电子电路模拟、计算流体力学相关的领域来说，稀疏矩阵向量乘法是非常重要的。本章还介绍如何在主流的X86、ARM和NVIDIA
GPU上优化稀疏矩阵向量乘法运算。 第3章 介绍如何在X86和GPU处理器上优化偏微分方程的求解，主要介绍如何求解热传递问题和三维Stencil问题。 第4章 介绍如何在X86处理器和GPU上优化常见的分子动力学算法，如邻居搜索、范德华力计算、键长伸缩力计算和径向分布函数计算。 第5章 介绍如何在X86、ARM和GPU上优化常见的机器学习算法，如kmeans、KNN、二维卷积和四维卷积的计算性能。最后介绍多GPU并行卷积神经网络的常见方法，以及如何使用数据并行来使用多GPU并行优化Caffe。 对于对并行和代码性能优化不太了解的人员，笔者建议先阅读本系列的第一本书《并行算法设计与性能优化》和第二本书《并行编程方法与优化实践》。对于对并行或代码性能优化非常了解的人员，可按照自己相关的领域选择对应的章节阅读。 勘误和支持 由于笔者的水平有限、工作繁忙、编写时间仓促，而向量化、并行和代码性能优化又是一个正在高速发展的、和硬件及算法密切相关的、影响因素非常多、博大精深、兼具个人特色的领域，许多问题还没有统一的解决方案，虽然笔者已经努力确认很多细节，但书中难免会出现一些不准确的地方，甚至是错误，恳请读者批评指正。你可以将书中的错误或写得不好的地方通过邮件，微博联系“ 异构并行计算风辰 ”或微信联系“ 风辰 ”，以便再版时修正，笔者会尽快回复大家。如果有更多的宝贵意见，也欢迎发送邮件，期待能够得到读者朋友们真挚的反馈。 致谢 首先要感谢我的老婆，她改变了我的人生轨迹，让我意识到人生有如此多的乐趣。如果不是她容忍我晚上10点下班回家后还要接着写书，此系列的出版估计得晚个两三年。 感谢我的母校中国地质大学(武汉)的图书馆，那是我对并行计算产生兴趣的地方。感谢我的母校中国科学院研究生院和中国科学院图书馆，在那里我奠定了从事并行计算事业的基础。 感谢我的朋友陈实富、高洋等，如果没有你们，我还需要更多时间来提升水平。感谢我的老板王鹏、吴韧和汤晓欧，在这些技术大佬和“ 人生赢家 ”的指导下，我才会成长得如此迅速。 感谢我在英伟达(NVIDIA)时的实习生、百度时的实习生和我现在的同事及下属，如果不是你们的努力工作，我没有时间来写这个系列。 感谢机械工业出版社华章公司的高婧雅和杨福川老师，我本无意出版此书，是你们

引导我将这本书付梓成书，是你们帮我修改书稿，让它变得可读、可理解，是你们帮我修正错误，是你们的鼓励和帮助使得我顺利完成全部书稿。 最后感谢我的爸爸、妈妈、姥姥、姥爷、奶奶、爷爷，感谢你们将我培养成人，并时时刻刻为我提供精神力量！ 谨以此书献给我最爱的家人，以及众多热爱代码性能优化、向量化、并行计算的朋友们！ 愿你们快乐地阅读本书！ 风辰
显示全部信息