

<<MIC高性能计算编程指南>>

图书基本信息

书名：<<MIC高性能计算编程指南>>

13位ISBN编号：9787517003380

10位ISBN编号：7517003380

出版时间：2012-11

出版时间：水利水电出版社

作者：王恩东 等编著

页数：307

字数：433000

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>

<<MIC高性能计算编程指南>>

前言

高性能计算是信息领域的前沿高新技术，在保障国家安全、推动国防科技进步、促进尖端武器发展方面具有直接推动作用，是衡量一个国家综合实力的重要标志之一。

随着信息化社会的飞速发展，人类对信息处理能力的要求越来越高，不仅石油勘探、气象预报、航天国防、科学研究等需求高性能计算，而且金融、政府、教育、企业、网络游戏等更广泛的领域对高性能计算的需求也在迅猛增长，百亿亿次计算（Exascale）已提上研究日程，人们正期待着未来采用百亿亿次计算机解决更大规模、更加复杂的问题。

21世纪多核、众核时代已经来临，高性能计算产业正在经历一场深刻的变革，并行计算将是未来的发展趋势，也将再次成为研究热点。

目前主流的集群架构系统，采用CPU同构模式，其单节点内拥有十几个甚至几十个CPU计算核心已非罕事，大规模计算应用中一次甚至可以利用数十万个以上CPU核心，然而采用CPU同构架构系统将面临着性能功耗比、性能访存比、并行效率等巨大挑战。

而采用CPU+GPU的异构架构进行通用科学计算，利用GPU众核技术进行加速，在业界已掀起一阵热潮，但其也面临着细粒度并行算法、编程效率、大规模计算性能等重大挑战。

如何在提高大规模计算系统性能、提高软件生产力的同时缩短编程周期，降低功耗将是我们思考和关注的焦点。

Intel公司推出了基于集成众核（Many Integrated Core，MIC）架构的至强融核（Intel·Xeon Phi™）系列产品，用于解决高度并行计算问题。

该产品双精度性能达到每秒一万亿次以上，它基于现有的x86架构，支持OpenMP、pThread、MPI等多种业内熟悉的并行编程模型，采用传统的C/C++/Intel·Cilk™ Plus和Fortran等语言进行软件开发，其特点以编程简单（引语方式）著称，具有丰富的工具链支持。

对于采用传统CPU平台很难实现性能进一步提升的部分应用，使用MIC可以带来性能的大幅提升，并且CPU与MIC可以共用一份代码，在x86架构下实现CPU+MIC异构协同计算的完美结合，为广大高性能计算用户提供了全新的计算解决方案。

浪潮-Intel中国并行计算联合实验室从2011年8月24日成立以来，就一直参与MIC技术研究，并在MIC平台开展实际的高性能计算应用项目工作，为Intel·Xeon Phi™系列产品最终顺利推出，贡献了自己的一份力量。

我们在深入了解MIC软硬件的同时，也积累了大量的开发经验。

我们很荣幸能够参与到这场高性能计算的技术革命当中，并作为先行者，将MIC技术介绍给各位读者。

希望通过本书，能让更多的读者了解MIC技术，并享受到Intel·Xeon Phi™系列产品带来的好处。

本书适合的读者 本书的主要目标是为开发人员提供一些帮助，使他们能够学会使用Intel·Xeon Phi™系列产品，并利用其开发、移植或优化并程序。

本书的主要内容是讲述一些使用MIC技术时的编程语法、程序设计技术和优化手段，并基于我们的应用性能优化经验，介绍一些在实际应用中遇到的问题和解决方案。

我们假设读者拥有一定并程序开发的基础知识，但是对MIC技术知之甚少。

本书并非讲解并行计算原理或算法的书籍，因此假设读者拥有相关原理和算法的知识，但是在面对具体并行算法时，本书仍然会进行描述。

本书假设读者熟悉OpenMP、MPI等并行编程常用的手段，但仍会对常用的语法进行简要的介绍。

本书假设读者熟悉C/C++/Fortran中的一种或几种编程语言，尤以熟悉C/C++为佳。

但是，书中给出的许多思想和建议也同样适合其他的高级语言，而且，如果以后Intel公司的Intel·Xeon Phi™产品支持其他语言，绝大部分优化手段和应用经验仍然可以适用。

总的来说，此书大致适合三类人群：
高校、科研院所学生、老师及科研人员，研究并行，研究多核、众核技术者；
IT从业人员，编写高性能计算软件，利用众核提升程序性能，在高性能计算领域追求性能的开发人员尤为适用；
行业高性能计算领域应用人员，如石油勘探、生物基因、医疗图像、金融、航空航天、气象气候、材料化学等人员，目的是利用MIC提升原CPU程序性能，从而提

<<MIC高性能计算编程指南>>

高生产力。

我们希望通过我们的努力能够使得本书让更多的读者受益，具有更加广泛的读者群。

本书特点： 由于MIC架构的特殊性，本书并不能被明确地归到某一类中。

这是一本介绍MIC编程语言的书，这是一本介绍Intel·Xeon Phi™产品的书，这又是一本介绍并行编程优化的书。

通过阅读本书，我们希望读者能够尽量多地了解有关MIC的方方面面，更希望读者能够在未来的实践中用上MIC技术，用好MIC技术。

本书由三篇组成。

第一篇为“MIC基础篇”，包括第1~8章，介绍MIC架构的基础知识。

其中： 第1章简要回顾了并行计算的发展历程，并对市面上现有的并行计算的硬件技术进行了对比，然后简要介绍了MIC技术的情况，并给出了MIC技术的优势。

第2章介绍了支撑MIC技术的软硬件架构的相关知识。

虽然没有这些背景知识并不影响使用MIC编程，但深入了解MIC架构能够编写出更加适合MIC架构的程序。

第3章介绍了MIC运行、编程环境的安装和配置。

由于MIC技术包含独立的硬件以及相关的驱动、编译器、编程工具等一系列配套软件，因此需要编程人员，尤其是系统配置人员对此有一定掌握。

本章供需要配置系统的读者阅读。

第4章通过一个计算圆周率PI的小例子，直观地展示了MIC程序的特点，并介绍了MIC程序背后的运行流程。

第5章介绍了MIC编程所需的背景知识，包括OpenMP和MPI的基本语法。

如果已有这方面的基础，可以跳过本章。

第6章介绍了MIC的编程模式、语法、环境变量、编译选项等。

通过本章的学习，可以掌握编写自己的MIC程序的方法。

第7章介绍了MIC相关的调试和优化工具以及它们的用法。

学会使用工具，可以为调试和优化带来方便。

第8章介绍了Intel公司的一些可以用在MIC上的数学库的用法，包括VML、FFT、Blas等。

第二篇为“性能优化篇”，包括第9章和第10章的内容。

其中： 第9章首先讲述了MIC优化的基本原则和策略，之后分不同方面详细阐述了MIC优化的方法与使用场合。

本章讲述的内容已基本涵盖MIC优化的主要方法，而且，除个别方法以外，大部分方法也通用于CPU并行计算编程的实践当中。

第10章通过对并行计算中典型示例--矩阵乘法的优化，以理论结合实际的方式，一步步地详细解释了优化步骤是如何应用的。

第三篇为“工程开发篇”，包括第11章和第12章的内容。

这是全书的最后一部分，其中： 第11章通过作者对自身团队工程开发优化经验的总结，提炼出一套并程序开发在工程应用中的方法。

对如何判断一个串行或并行的CPU程序是否适用MIC进行计算，并且如何将程序移植到MIC上进行了讨论。

第12章通过两个实际生产中的案例，讲述了MIC技术是如何对实际项目产生影响的。

本书前期由高效能服务器和存储技术国家重点实验室主任、浪潮-Intel中国并行计算联合实验室主任、浪潮集团高级副总裁王恩东牵头，并明确写作方向，对技术发展趋势把关，浪潮-Intel中国并行计算联合实验室首席工程师张清具体负责制定写作计划、确定整本书的提纲、结构、每章节书写的内容、风格和读者定位。

中期由张清组织和推进团队写作此书，并定期进行审阅，对执笔作者完成的章节内容的准确性、技术表述的深浅度、可读性进行审核，并反馈修改意见，具体由浪潮-Intel中国并行计算联合实验室工程师沈铂、张广勇、卢晓伟、吴庆、王娅娟五位作者共同执笔，其中第1章由沈铂撰写，第2章由吴庆、沈

<<MIC高性能计算编程指南>>

铂撰写，第3章由王娅娟撰写，第4章由沈铂撰写，第5章由沈铂撰写，第6章由沈铂撰写，王娅娟也参与了部分撰写工作，第7章由吴庆撰写，第8章由卢晓伟撰写，第9章由张广勇撰写，沈铂、王娅娟也参与了部分撰写工作，第10章由张广勇撰写，第11章由沈铂撰写，第12章由卢晓伟、张广勇撰写。后期由王恩东、张清、Intel公司的何万青（Warren）博士、Victor Lee（李汇强）博士审阅，张清负责最后内容审定、统稿。

本书中全部完整的源代码已经过作者反复测试通过，但由于MIC技术尚处于起步阶段，不能保证源代码在更新版本中仍然可用。

因此，如果编译器或MIC执行环境有更新，请参阅相应版本的Intel官方手册。

<<MIC高性能计算编程指南>>

内容概要

《MIC高性能计算编程指南》是全球第一本全面介绍MIC软硬件体系架构、应用及编程开发优化的书籍。

书中介绍了使用MIC进行通用计算所需要了解的硬件架构、语法、程序优化技巧等知识，是进行MIC高性能与通用计算程序开发的入门教材和参考书。

本书共分12章。

第1章介绍高性能计算的发展历程；第2章深入介绍MIC的软硬件架构；第3章介绍MIC编程环境的搭建；第4章引入一个简单的MIC实例；第5章简要介绍与MIC编程相关的OpenMP和MPI相关知识；第6章详细讲解了MIC编程的语法；第7章介绍MIC编程用到的工具软件；第8章介绍MIC可以使用的数学库及其用法；第9章详细讲解如何优化MIC程序，从多个方面系统阐述了MIC优化的方式和方法；第10章通过一个典型的矩阵乘法示例，展示MIC优化方法的应用；第11章介绍将MIC技术应用于工程中的流程和方法；第12章引入两个实际工程的例子，讲解如何将MIC技术应用于实际生产过程当中。

《MIC高性能计算编程指南》可作为MIC的入门学习和编程参考书，主要面向从事高性能计算的程序员与工程师、MIC加速计算专业领域的科研人员，以及对MIC通用计算感兴趣的程序员，也可作为开设相关课程的高等院校与科研机构的教材。

本书由王恩东等编著。

<<MIC高性能计算编程指南>>

作者简介

王恩东，浪潮-Intel中国并行计算联合实验室主任，研究员，国务院特殊津贴专家，“863”计划领域专家，高效能服务器和存储技术国家重点实验室主任、浪潮集团高级副总裁，兼任国际信息处理联合会(IFIP)中国委员会主席、中国计算机学会副理事长等职，获国家科技进步奖3项，并获何梁何利奖，发明专利26项。

张清，浪潮-Intel中国并行计算联合实验室首席工程师，浪潮HPC应用技术经理，主要从事高性能计算、并行计算，研究CPU多核、GPU、MIC众核技术，曾在生命科学、石油、气象、金融等HPC领域主持多个异构并行计算项目。

沈铂，浪潮-Intel中国并行计算联合实验室应用研发资深工程师，主要从事高性能算法、软件开发与优化等方面的技术研究与应用工作，具有多年的生命科学、石油物探、气象等领域开发调优经验。

张广勇，内蒙古大学计算机体系结构专业硕士，现任浪潮-Intel中国并行计算联合实验室研发工程师，主要从事GPU / MIC高性能应用软件的开发优化工作，具有丰富的项目开发经验，并在国内外会议期刊发表多篇优秀论文。

卢晓伟，大连理工大学计算机应用技术硕士，浪潮-Intel中国并行计算联合实验室应用研发资深工程师，主要从事多个科学领域的算法移植、优化等工作，具有丰富的高性能异构协同计算开发经验。

<<MIC高性能计算编程指南>>

书籍目录

- 序一
- 序二
- 前言
- 第一篇 MIC基础篇
- 第1章 MIC高性能计算
- 1.1 多核、众核计算的发展
- 1.2 MIC技术简介
- 1.3 为什么要选择MIC
- 1.3.1 SMP
- 1.3.2 集群 (cluster)
- 1.3.3 GPGPU
- 第2章 MIC硬件及软件架构
- 2.1 MIC硬件架构
- 2.1.1 术语解析
- 2.1.2 MIC硬件架构概览
- 2.1.3 MIC Core
- 2.1.4 环形互联总线Ring
- 2.1.5 CLOCK
- 2.1.6 页表 (Page Tables)
- 2.1.7 系统接口
- 2.1.8 性能监控单元和事件管理器
- 2.1.9 电源管理
- 2.2 MIC软件架构
- 2.2.1 概述
- 2.2.2 Bootstrap
- 2.2.3 Linux加载器
- 2.2.4 微操作系统 (OS)
- 2.2.5 对称通信接口 (SCIF)
- 2.2.6 主机驱动
- 2.2.7 sysfs节点
- 2.2.8 MPI应用的MIC软件栈
- 2.2.9 应用编程接口 (API)
- 第3章 MIC安装、环境配置
- 3.1 MIC环境配置
- 3.1.1 前期准备
- 3.1.2 安装Host端的Linux操作系统
- 3.1.3 安装MIC驱动
- 3.1.4 安装在MIC上编译C/C++的编译器
- 3.2 SDK示例运行
- 第4章 第一个MIC实例——计算PI
- 第5章 OpenMP和MPI编程基础
- 5.1 OpenMP基础
- 5.1.1 OpenMP简介
- 5.1.2 OpenMP编程模型
- 5.1.3 OpenMP语法简要介绍

<<MIC高性能计算编程指南>>

5.2 MPI基础

5.2.1 启动和终止MPI库

5.2.2 获取信息

5.2.3 发送和接收消息

第6章 MIC编程

6.1 MIC编程模型

6.2 应用模式

6.2.1 CPU原生模式

6.2.2 CPU为主MIC为辅模式

6.2.3 CPU与MIC对等模式

6.2.4 MIC为主CPU为辅模式

6.2.5 MIC原生模式

6.3 MIC基本语法

6.3.1 offload

6.3.2 变量和函数声明

6.3.3 头文件

6.3.4 环境变量

6.3.5 编译选项

6.3.6 其他问题

6.4 MIC上的MPI

6.4.1 MIC上的MPI限制

6.4.2 MIC上MPI编程模型

6.4.3 MIC上的MPI环境配置

6.4.4 编译及使用

6.4.5 MIC上的MPI示例

6.5 SCIF编程

6.5.1 什么是SCIF

6.5.2 SCIF的基本概念介绍

6.5.3 SCIF基本通信过程

6.5.4 SCIF用到的API函数

第7章 MIC软件调试与性能分析工具

7.1 Intel软件工具链对MIC的支持

7.2 MIC软件调试工具IDB

7.2.1 IDB简介

7.2.2 IDB的操作界面

7.2.3 IDB对MIC架构的支持与要求

7.2.4 使用IDB调试MIC程序

7.3 MIC性能分析工具VTune

第8章 Intel MIC MKL库使用方法

8.1 Intel MKL核心函数库介绍

8.2 在MIC卡上使用Intel MKL

8.2.1 编译器辅助offload方式

8.2.2 自动offload方式

8.3 FFT在MIC上的使用

8.3.1 FFT简介

8.3.2 FFT在MIC上的使用方法一

8.3.3 FFT在MIC上的使用方法二

<<MIC高性能计算编程指南>>

8.4 BLAS在MIC上的使用

8.4.1 BLAS简介

8.4.2 在MIC上调用BLAS库方法

第二篇 性能优化篇

第9章 MIC性能优化

9.1 MIC性能优化策略

9.2 MIC优化方法

9.2.1 并行度优化

9.2.2 内存管理优化

9.2.3 数据传输优化

9.2.4 存储器访问优化

9.2.5 向量化优化

9.2.6 负载均衡优化

9.2.7 MIC线程扩展性优化

第10章 MIC优化示例：矩阵乘法

10.1 矩阵乘法串行算法

10.2 OpenMP多线程矩阵乘法

10.3 MIC多线程矩阵乘法

10.3.1 基本版本

10.3.2 向量化优化

10.3.3 SIMD指令优化

10.3.4 矩阵分块乘法

第三篇 工程开发篇

第11章 基于MIC的HPC应用开发过程

11.1 热点测试

11.1.1 准备工作

11.1.2 热点测试及定位

11.2 程序分析

11.2.1 程序移植模式分析

11.2.2 计算规模分析

11.2.3 特点分析

11.2.4 热点并行性分析

11.2.5 向量化分析

11.2.6 MIC内存分析

11.2.7 程序分析总结

11.3 MIC程序开发过程

11.3.1 基于CPU的OpenMP并行

11.3.2 基于MIC的线程扩展

11.3.3 单节点CPU+MIC协同并行

11.3.4 MIC集群并行

第12章 基于MIC的HPC应用实例

12.1 基于单节点CPU+MIC协同计算电子断层三维重构并行算法

12.1.1 电子断层三维重构技术及SIRT算法介绍

12.1.2 SIRT串行程序分析

12.1.3 基于OpenMP的SIRT并行程序开发

12.1.4 基于MIC平台的SIRT并行程序开发

12.1.5 单节点多卡及CPU+MIC异构协同计算架构设计

<<MIC高性能计算编程指南>>

12.2 基于多节点CPU+MIC协同计算大涡模拟并行算法

12.2.1 格子-Boltzmann大涡模拟算法介绍

12.2.2 大涡模拟串程序分析

12.2.3 基于OpenMP的大涡模拟并行算法

12.2.4 基于MIC的大涡模拟并行算法

12.2.5 基于多节点CPU+MIC协同计算平台的大涡模拟并行算法

参考文献

<<MIC高性能计算编程指南>>

章节摘录

版权页：插图：因为MIC核（core）是x86架构的核，所以Bootstrap在自检时类似于BIOS。Bootstrap会在板卡第一次加电的时候启动，但是也会在发生灾难性失败后由主机端重启时运行。Bootstrap的责任是初始化MIC卡和启动laOS。

Bootstrap由两个不同的代码块构成，FBOOT0和FBOOT1。

FBOOT0属于片上ROM存储器，无法升级。

FBOOT1在Flash上，可以升级。

2.2.2.1 FBOOT0 当卡重置以后，第一个执行的指令就在FBOOT0上。

这个代码段是可信根（root of trust，注：系统会无条件地信任可信根），因为它是不能被更改的。它是为了验证第二部分，即FBOOT1，并将可信根传递给FBOOT1。

如果验证失败，则FBOOT0会切断环和核中的电源，防止任何其他动作。

从这个状态恢复的唯一途径是手工更改卡上的跳线，让卡进入僵尸模式（zombie mode）。

僵尸模式允许主机重编程Flash芯片，将坏掉的FBOOT1代码块恢复。

FBOOT0的执行流如下：（1）设置CAR模式（Cache as RAM，注：一种将cache作为内存的技术）以减少执行时间。

（2）过渡到64—bit保护模式。

（3）验证FBOOT1。

（4）如果验证失败，关闭卡。

（5）如果验证成功，把控制权交给FBOOT1。

2.2.2.2 FBOOT1 FBOOT1的责任是配置卡和启动 μ OS。

配置卡包括初始化所有核、非核单元和内存。

这种处理方式和典型的x86内核一样。

代码必须在64位保护模式下启动才能访问必要的配置寄存器。

当启动第三方 μ OS——当然是基于MPSS的Linux μ OS，可信根不会向下传递。

可信根只会在维护模式下传递，并且只是在需要特权操作的时候。

维护模式会锁定一些寄存器并重写一些硬件。

在验证结束后确定启动的ktOS类型。

FBOOT1回叫（call back）FBOOT0，FBOOT0用嵌入其中的公钥运行验证过程。

只有维护模式的 μ OS会签名私钥，其他模式只能使用未签名的。

如果验证通过，则会启动维护模式的 μ OS。

如果验证失败，则假定使用第三方的pOS并使用Linux启动协议，锁定对敏感寄存器的访问以保护知识产权。

FBOOT1的执行流如下：（1）设置内存频率然后重置卡。

（2）完成核的初始化。

<<MIC高性能计算编程指南>>

编辑推荐

《MIC高性能计算编程指南》由浪潮—Intel联合实验室的几位成员共同完成。书中介绍了英特尔至强融核产品的相关知识、使用至强融核进行高性能计算的编程方法、优化使用至强融核程序的方法，以及在实际应用中的两个利用至强融核技术提高性能的成功案例。

《MIC高性能计算编程指南》结构清晰、通俗易懂，从编程基础到优化到具体工程开发，表述言简意赅，用简单代码实践阐述理论，并配有大量的图、表、程序片段、完整案例帮助读者理解。

<<MIC高性能计算编程指南>>

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>