



空间信息 并行处理方法与技术

黄 方 王力哲 谭喜成 著



科学出版社

空间信息并行处理方法与技术

黄 方 王力哲 谭喜成 著

国家自然科学基金重点项目 (U1711266)

国家自然科学基金面上项目 (41871312)

国家重点研发计划 (2017YFB0504202)

湖北省自然科学基金面上项目 (2017CFB433)

资助出版

科学出版社

北 京

内 容 简 介

本书着眼于空间信息处理领域的行业应用/算法,采用现阶段主流的多种高性能计算平台和对应的编程模型,对实际的空间信息处理算法的不同并行计算模式进行研究,形成基于集群的 MPI 并行计算方法、基于多核/MIC 的 OpenMP 并行计算方法、基于 GPU 平台的 CUDA/OpenCL 并行计算方法、基于 CPU+GPU 协同的异构并行计算方法,以及基于 Spark 平台的大数据并行计算方法。这些计算方法都有各自适用的范围及优势,作者通过实例为读者提供选择性较多的空间信息并行处理参考方法,从而为大规模的空间数据处理与分析提供技术支持。

本书可作为遥感图像处理、地理信息系统及其相关专业的研究生和本科生的教材,可为相关专业的老师提供参考,也可作为空间数据处理的行业人员提供实际帮助。

图书在版编目(CIP)数据

空间信息并行处理方法与技术/黄方,王力哲,谭喜成著. —北京:科学出版社,2019.1

ISBN 978-7-03-059115-9

I. ①空… II. ①黄… ②王… ③谭… III. ①空间信息技术-并行处理
IV. ①P208

中国版本图书馆 CIP 数据核字(2018)第 240125 号

责任编辑:杨光华/责任校对:董艳辉

责任印制:彭超/封面设计:苏波

科学出版社出版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2019 年 1 月第 一 版 开本:787×1092 1/16

2019 年 1 月第一次印刷 印张:12 3/4

字数:299 400

定价:88.00 元

(如有印装质量问题,我社负责调换)

前 言

随着空间数据获取技术的不断进步,一方面,人们获取的空间数据的途径越来越多,数据容量越来越大,数据格式复杂多样,处理计算也越发复杂;另一方面,人们对空间数据处理的时效性要求也越来越高。面对一些数据密集型和计算密集型的空间数据处理、分析和应用来说,高性能计算(HPC)一直是解决该问题的最为重要的技术手段。高性能地学计算(HPGC)是一门利用如分布式计算、并行计算、网格计算、异构计算及云计算/大数据等各种先进的高性能计算技术高效地处理各类空间数据,对各种复杂地学应用/现象进行分析、模拟、预测的科学技术,它广泛应用于地学领域中各类空间数据处理、图像处理,以及全球尺度的空间数据产品生产等各个方面。

高性能地学计算与高性能计算技术紧密相关,并随着高性能计算技术的不断发展而发展,故存在多种并行实现模式。一直以来,基于集群的并行计算、分布式计算、网格计算是高性能地学计算的传统方式。然而近年来多核、众核计算(如GPU/MIC等),以及基于CPU+GPU/MIC的协同异构计算、云计算/大数据等已经成为高性能计算的主流技术方向,为高性能地学计算提供了更多更新的可替换的计算方案。因此,新的高性能地学计算模式需要更加系统地深入探索和研究,迫切需要与时俱进地随高性能计算技术的发展而发展。

在此背景下,本书着眼于从空间信息处理领域的行业算法/应用,在现阶段流行的不同高性能计算平台和对应的编程模型支持下,对实际的空间数据处理和分析的并行算法进行总结凝练,形成在集群平台、多核/众核平台、CPU+GPU/MIC的异构计算平台及基于云计算/大数据平台下的计算模式、研究方法和并行技术。不同的计算方法和技术有各自适用的对象与优势,都能在提升性能和时效性方面取得显著效果。本书主要根据列举的一些空间信息处理的算法实例来加深读者对本书内容的理解、学习和参考。然而,书中的相关研究仍有很多提升的空间,一些难点问题还可以做进一步的深入研究。

本书由黄方、王力哲、谭喜成共同合作完成。诚挚感谢美国德州农工大学(TAMU)的Jian Tao博士、美国路易斯安娜州立大学(LSU)的Jim教授和Kelin Hu博士,Intel公司的Yue Hu博士等,他们为本书的相关研究提供了建议,为实验开展提供了对应的平台。在实验过程中,还用到了中国科学院超级计算中心、清华大学、电子科技大学资源与环境学院的相关高性能计算平台和设备。本书提供的算法案例,部分来源于一些国家级、省部级项目课题的实际需求,对涉及项目的合作伙伴,如周纪教授、刘鹏副研究员、马艳副研究员、周小成副研究员、王玉柱博士等,在此表示诚挚谢意。部分算法源码来自一些开源项目,对这些开源项目的作者给予的支持表示谢意。

特别感谢研究生冯杰、董蕾、卜栓栓、金都、范光松、朱强等,他们为本书的实际案例进行了细致研究和实验,提供了第一手资料;感谢研究生袁帅、铁博、杨浩、彭思远、陈胤杰、陆俊、李丽、吴俊锋、兰波等,他们对本书的整理、修改、检核、校对等做了大量的工作。感谢本书参考文献中的各位前辈、同仁,他们的研究成果是本书研究的坚实基础。

由于作者水平和撰写时间有限，书中难免会存在一些疏漏之处，希望广大读者和专家批评指正。

作 者
2018 年 11 月

目 录

第 1 章 绪论	1
第 2 章 高性能计算与高性能地学计算	5
2.1 高性能计算演化进程	5
2.1.1 向量机、向量并行机时代	5
2.1.2 大规模并行计算时代	6
2.1.3 异构并行计算时代	7
2.1.4 云计算与大数据时代	8
2.2 常见的高性能计算硬件平台	9
2.2.1 Linux 集群平台	9
2.2.2 GPU 平台	10
2.2.3 集成众核 MIC 平台	11
2.3 常见并行模型与方法技术	14
2.3.1 MPI 并行模型	14
2.3.2 OpenMP 并行模型	15
2.3.3 CUDA 并行模型	16
2.3.4 OpenCL 并行模型	17
2.3.5 云计算编程技术	20
2.4 衡量高性能计算并行算法的指标	22
2.4.1 计时工具	22
2.4.2 加速比	22
2.4.3 执行效率	23
2.4.4 可扩展性	23
2.4.5 阿姆达尔定律	23
2.5 高性能地学计算	24
2.5.1 高性能地学计算研究现状	24
2.5.2 高性能地学计算未来展望	26
2.6 本章小结	26
第 3 章 基于集群平台的 MPI 并行数据处理技术	28
3.1 概述	28
3.1.1 并行计算基础	28
3.1.2 MPI 及 MPICH	29
3.1.3 相关研究现状	31
3.2 基于集群的 MPI 并行计算方法	31
3.2.1 集群平台类型的选择	31

3.2.2	编程开发模型和工具的选择	32
3.2.3	需要开发的并程序的定位	32
3.3	基于 Linux 集群平台的 MODTRAN 并行算法	33
3.3.1	应用背景概述	33
3.3.2	MODTRAN 数据处理并行化研究现状	36
3.3.3	PMODTRAN 并行算法设计	37
3.3.4	PMODTRAN 并行算法实现	38
3.3.5	PMODTRAN 并行算法性能测试	41
3.4	基于 Linux 集群平台的等高线生成并行算法	44
3.4.1	栅格 DEM 生成等高线算法原理	45
3.4.2	优化后的等高线追踪串行算法实现	47
3.4.3	串行算法热点分析	51
3.4.4	栅格 DEM 生成等高线并行算法设计	53
3.4.5	栅格 DEM 生成等高线并行算法实现	55
3.4.6	栅格 DEM 生成等高线并行算法实验与测试	58
3.4.7	栅格 DEM 生成等高线并行算法的进一步优化	61
3.5	基于 Windows 集群的 MPI 并行处理方法	66
3.5.1	Windows 集群搭建与配置	67
3.5.2	基于 Windows 集群的并行算法设计与实现	67
3.6	本章小结	69
第 4 章	基于 Intel 多核/众核平台的 OpenMP 并行数据处理技术	71
4.1	概述	71
4.1.1	多核与 MIC 设备	71
4.1.2	OpenMP 编程模型	72
4.1.3	基于 Intel 多核/众核计算平台的研究现状	74
4.2	基于 Intel 多核/众核平台的 OpenMP 并行数据处理方法	75
4.3	基于 Intel 多核平台的坡度坡向并行算法	77
4.3.1	坡度坡向算法原理及串行实现	77
4.3.2	利用 OpenMP 实现坡度、坡向并行算法	80
4.3.3	在多核平台上并行坡度、坡向算法性能测试	80
4.4	基于 Intel 多核/众核平台的 NLM 图像处理并行算法	81
4.4.1	NLM 图像处理并行算法原理	81
4.4.2	NLM 算法并行化研究现状	83
4.4.3	基于多核平台的 NLM 并行算法并行化设计	84
4.4.4	基于 Intel 多核/众核平台的 NLM 并行算法实现	85
4.4.5	基于 Intel 众核平台 MIC 的 NLM 并行算法实现	87
4.4.6	NLM 并行算法在 Intel 多核/众核平台上的性能测试	88
4.5	本章小结	93

第5章 基于GPU平台的CUDA/OpenCL并行数据处理技术 94

5.1 概述 94

5.1.1 GPU与通用GPU计算 94

5.1.2 CUDA与OpenCL编程模型 95

5.1.3 通用GPU计算在地学领域的应用现状 96

5.2 基于GPU的空间信息并行处理方法 97

5.3 利用CUDA实现压缩感知重构并行算法 98

5.3.1 压缩感知重构算法原理及实现 98

5.3.2 压缩感知重构算法并行化研究现状 99

5.3.3 压缩感知重构算法热点分析及其并行化设计 100

5.3.4 基于CUDA的压缩感知重构并行算法实现 102

5.3.5 基于CUDA的压缩感知重构并行算法性能测试 104

5.4 基于OpenCL的压缩感知重构并行算法 105

5.4.1 基于OpenCL的压缩感知重构并行算法实现 105

5.4.2 基于OpenCL的压缩感知重构并行算法实验 110

5.5 本章小结 112

第6章 基于CPU+GPU/MIC异构平台的协同并行数据处理技术 113

6.1 概述 113

6.2 基于CPU+MIC/GPU异构平台的协同并行处理方法 114

6.3 基于CPU+MIC异构计算平台下的NLM协同并行算法 117

6.3.1 CPU+MIC协同的NLM并行算法 117

6.3.2 基于动态任务分配的CPU+MIC协同NLM并行算法 119

6.3.3 基于CPU+MIC异构计算平台的协同NLM并行算法性能测试 122

6.4 基于CPU+GPU/MIC异构计算平台下的泛Kriging协同并行算法 124

6.4.1 Kriging算法并行化研究现状 124

6.4.2 泛Kriging算法原理及其实现 125

6.4.3 基于OpenCL的泛Kriging并行算法设计 127

6.4.4 基于OpenCL的泛Kriging并行算法实现 130

6.4.5 基于CPU+GPU/MIC异构平台的泛Kriging并行算法性能测试 138

6.5 不同异构计算平台算法性能对比实验 141

6.6 本章小结 143

第7章 基于大数据Spark平台的并行数据处理技术 144

7.1 概述 144

7.2 云计算与大数据及其关键技术 145

7.2.1 云计算与大数据 145

7.2.2 Hadoop 146

7.2.3 Spark 148

7.2.4 Docker容器虚拟化技术 151

7.2.5 大数据集群资源管理框架 153

7.3 基于云计算/大数据平台的并行数据处理方法	155
7.4 DBSCAN 算法及其在 Spark 平台上并行设计与实现	156
7.4.1 DBSCAN 算法	156
7.4.2 DBSCAN 算法并行化现状	158
7.4.3 DBSCAN 并行算法在 Spark 平台上的设计与实现	160
7.4.4 DBSCAN 并行算法在 Spark 平台上的优化	165
7.4.5 DBSCAN 并行算法在不同资源管理器模式下的并行实现	168
7.5 DBSCAN 并行算法性能测试与分析	170
7.5.1 实验平台及配置	170
7.5.2 实验数据	171
7.5.3 实验内容	171
7.5.4 测试结果及分析	172
7.6 基于 Spark 平台的 DBSCAN 并行算法在城市拥堵区域发现应用	177
7.6.1 DBSCAN 并行算法在城市拥堵区域发现的应用流程	177
7.6.2 实验数据与平台	178
7.6.3 实验测试与分析	178
7.7 本章小结	182
参考文献	183

第1章 绪 论

地学计算 (geo-computation, GC) 是一门将计算机应用于解决地理空间问题的学科, 它被视为计算机科学在地理科学领域的应用范例 (薛勇 等, 2008)。由于地学计算没有统一的定义, 一般认为地学计算是以计算机方法为基本科学工具处理地理信息和分析地理现象的地理学分支, 它包括地理信息处理与管理、地理数据挖掘、地理过程建模模拟, 以及支持这些处理与分析的软件工程和计算体系研究, 如地理信息系统 (geographical information system, GIS) (Bonham, 1994; Tomlinson, 1984)、地理决策支持系统 (Matthews et al., 1999) 和空间网格体系 (刘云翔 等, 2007; 李德仁 等, 2003; Foster et al., 2002) 等。有人认为, 如地理学、地质学、海洋学、大气物理学等学科中凡是涉及复杂空间算法计算、建模模拟 (Law et al., 2009) 和计算体系 (Bryant et al., 2011) 等的相关研究, 也属于地学计算的范畴。这种看法有一定的道理, 因为上述研究彼此间都具有很紧密的相关性和通用性, 且同属于地理信息科学。由于专业限制, 本书作者所做工作主要集中在遥感 (remote sensing, RS) 和地理信息系统领域。

概括地说, 地学计算的发展基础和支撑前提为高性能计算 (high performance computing, HPC) 中的计算技术、算法和例证。一般认为, 地学计算包括 4 个前沿方向 (Xue et al., 2004)。

(1) 地理信息系统、遥感和卫星导航系统 (satellite navigation system, SNS) (Grewal et al., 2001): 产生海量的空间数据。

(2) 人工智能 (artificial intelligence, AI) (Russell et al., 2002)、机器学习 (machine learning, ML) (Mitchell et al., 2003) 和深度学习 (deep learning, DL) (Bengio, 2009) 等: 提供分析技术与工具。

(3) 高性能计算: 提供高性能计算基础设施和高效率的处理计算能力。

(4) 先进的 Internet 及无线电通信 (如 4G/5G 网络等) 技术: 提供高速的、高服务质量 (quality of service, QoS) 的网络连接。

高性能地学计算 (high performance geo-computation, HPGC) 是地学计算的新发展, 它更多地强调先进高性能计算技术在解决地学计算中广泛存在的数据密集型或算法密集型算法、应用、分析, 以及模拟问题的快速高效解算, 也就是说它是地学计算与高性能计算技术直接紧密结合的产物。正是因为有以高性能为基础的地学计算, 地理环境中全球性或大区域性以时空演变为特征的地理现象的模拟, 如数值天气预报 (Kalnay, 2003)、全球气候变化 (Yohe, 2003)、地理信息系统、遥感定量解译与图像解译 (Swain et al., 1978) 等, 才有可能真正实现 (薛勇 等, 2008), 原有的一些受计算机条件限制而难以构建的模型才得以完善、改进和运行, 通过高性能计算进一步探索地理环境, 认知新的概念。

作为地学计算提供空间数据的主要来源, 遥感旨在从遥感数据中提取反映地表信息的物理量, 获得地学相关的专题信息和领域知识 (杨海平 等, 2013; 陈述彭 等, 1998; Griffith, 1979); 而 GIS 则不单单作为传统空间数据的存储载体, 更多的是作为空间数据处

理、分析、输出的技术工具。目前遥感技术的最新发展呈现“三全”“三高”“三多”的趋势（顾海燕 等, 2016; Benediktsson et al., 2013），它们分别是：全天候、全天时、全球观测，高空间分辨率、高光谱分辨率、高时间分辨率，多平台、多传感器和多角度。短短几十年，随着信息技术和遥感传感器技术的快速发展，遥感手段获取的对地观测数据急剧增长（李德仁, 2003），现代遥感数据源得到了极大丰富，可以说人们已经进入了遥感大数据时代（李德仁 等, 2014a）。由于空间分辨率的提高，单个遥感图像的文件大小已经从 M 字节增加到 G 字节。如何快速获取地球观测遥感数据并从遥感图像中有效获取有用信息，已成为非常热门的研究课题。此外，对更准确的遥感数据分析和更短更少的处理时间的需求，也与日俱增。在遥感图像处理中存在一些数据密集型或计算密集型操作，将高性能计算用于大规模的遥感数据处理或者计算密集型的遥感图像解译分析，已成为遥感图像领域里面亟待解决的关键问题。与此同时，随着 GIS 的发展，GIS 逐渐演变成一门集计算机科学、地理学、地图学、测绘遥感科学、环境科学、空间科学、信息科学和城市科学等为一体且发展迅速的综合性学科（蔡孟裔, 2000; 陈述彭, 1999; 王之卓, 1979）。GIS 的应用领域也在不断拓宽，其作为一种分析和处理海量地理数据的通用技术，已被广泛应用于城市规划、交通运输、农业生产、自然资源调查、生态环境评估、公共设施管理等多个领域（张新长 等, 2005）。面对 GIS 的多样化应用，对 GIS 的处理能力要求也逐渐提高。然而，如何解决各种结构化和非结构化数据的高效存储问题、如何有效地利用这些数据进行快速分析处理，成为摆在人们面前的一道难题。

一直以来，信息处理的效率与计算机技术，尤其是计算机计算技术息息相关。计算机计算技术的发展，尤其是高性能计算的发展，对信息处理领域，特别是空间信息处理方面产生重要的影响与作用（马艳, 2007）。现阶段国家立项的有关高性能计算的相关科学研究正开展得如火如荼。

高性能地学计算是随着高性能计算技术的发展而不断发展的。早期应用在高性能地学计算的高性能计算技术主要包括集群计算技术（cluster）、分布式计算技术（distributed computing）、网格技术（grid computing）等。高性能计算技术随着计算技术的发展，一些新兴的高性能计算技术涌现出来，为地学计算涉及的实际应用问题提供了切实可行的解决方案。

从 1978 年英特尔（Intel）公司发布 8086 处理器以来，Intel、AMD 等厂商推出的中央处理器（central processing unit, CPU）性能不断提高，并且几乎严格遵循摩尔定律。各处理器制造厂商不断提升制造工艺技术，使用更精细的制程来制造处理器，使得处理器的主频不断提高，但 4 GHz 却是一个不可逾越的门槛。随后，CPU 由“纵向发展”转为“横向发展”，多核 CPU 不断涌现。Intel 公司在 2006 年 7 月 27 日发布的酷睿双核 CPU，标志着多核计算（multi-core computing）逐渐进入公众视野。这样，基于多核 CPU 的并行计算逐渐变得普遍。

1999 年，NVIDIA 公司发布了第一款图形处理器（graphics processing unit, GPU），它是解决图形渲染中的复杂计算的专用处理器（Lindholm et al., 2008）。随着技术的发展，GPU 高度并行的众核架构，以及强大的内存访问带宽，使得其在累积的峰值频率和内存吞吐上已经表现出超过 CPU 的计算能力，再加上后来出现的计算统一设备架构（compute unified device architecture, CUDA）和开放运算语言（open computing language, OpenCL）

等技术的支持, GPU 演变成为一种支持通用计算的众核处理器——GPGPU (general-purpose computing on GPU) (Luebke, 2008), 在数据密集的通用计算方面显示出强大的潜力。目前, GPU 已经成为当前高性能领域最流行的加速器, 并已在高性能计算、桌面计算甚至嵌入式计算等领域取得了广泛的应用, 是一种极具发展潜力的处理器体系结构。

2012 年, 在美国 SC12 全球超级计算大会 (supercomputing conference, SC) 上, Intel 公司正式发布了自己的众核协处理器系统——采用集成众核架构 (Intel many integrated core, MIC) 的至强融核 Xeon Phi, 用于高性能并行计算。Intel 公司推出的 MIC 计算平台, 具有数十个 x86 架构核心的协处理器众核架构, 以不同于 GPU 众核的方式, 使单节点上的高性能程序开发进入了一个新时期。

2013 年, 实测性能达 33.86 PFlops (1 PFlops 等于 1 千万亿次浮点指令/秒), 位列 TOP 500 首位的天河二号超级计算机的异构计算 (heterogeneous computing) 系统引起了人们的广泛注意。虽然 GPU 和 MIC 等协处理器表现出了惊人的计算能力, 但是在它们进行计算的同时, CPU 处于闲置状态。而异构计算则不负众望, 可以将不同计算设备协同运算, 完成那些对计算量有着苛刻要求的应用, 也将计算机的处理能力再次推上一个新的高峰。近年来在 TOP 500 排行榜上看到基于异构体系结构的计算系统的比例呈现上升的趋势。然而, 在异构计算系统获得较高峰值性能的同时, 其本身的编程问题变得日益严峻, 面向异构系统复杂的体系结构的性能优化工作往往也困难重重。

自 2007 年起, 云计算 (cloud computing) 及后续的大数据 (big data) 研究正在展开。然而, 云计算和高性能计算其实是一对出生时被分开的双胞胎, 高性能并行计算技术才是云计算的核心技术。云计算系统的本质是“资源虚拟化+并行计算”; 并行计算的一些成熟技术, 如 Map-Reduce (Dean et al., 2008) 编程框架、Big-Table (Chang et al., 2008) 并行数据库技术及 GFS (Google file system) (Ghemawat et al., 2003) 并行文件系统等都是云计算中具有核心竞争力的关键技术。在技术发展的过程中, 云计算只是高性能计算的新形态。另外, 大数据与高性能计算的影响是相辅相成的。高性能计算是大数据处理的基础计算平台, 而大数据使高性能计算从追求提升计算速度变成着重提高系统的吞吐率。同时, 大数据使高性能计算的应用模式从独占资源、统一付费变为虚拟占用、按量付费的方式。此外, 传统的高性能计算也要从以“计算”为中心转变为以“数据”为中心; 从离线批处理逐渐转变为实时流计算; 数据存储模式也要从集中式的行存储逐渐转变为以分布式的列存储为主 (陈国良 等, 2015)。

显然, 这些新的计算技术将为高性能地学计算提供更多的选项, 从而可以满足不同复杂度和量级的高性能地学计算与分析任务的需求, 进而扩大高性能地学计算在地理研究领域中的应用范围, 同时也需要在新的计算技术下总结探索更多适合地学计算的方法和模式。

一般地说, 在利用这些高性能计算技术, 进行地学计算领域里面的大规模空间数据处理和计算密集型算法研究时, 需要重点关注以下几方面。

(1) 采用何种的高性能计算技术。目前, 有许多高性能计算技术可以应用, 如计算机集群、多核计算、众核计算、异构计算、云计算/大数据计算技术等。每种计算技术都有各自的优缺点, 因此, 采用何种计算技术都应基于给定应用的具体要求、数据/算法的特征, 以及计算量的大小等因素。

(2) 无论采用何种高性能计算技术,都需要克服常见的高性能计算基础技术问题,如海量数据存储、快速高效的 I/O、数据分块、数据索引与可视化等。

(3) 基于应用程序本身的特性,通常需要设计或实现具有特定高性能计算平台的相应的通用并行算法和系统框架。在设计或和实现并行算法和系统框架时,还应考虑并行编程模式,数据或任务处理的并行策略和方法。

由于高性能地学计算需要兼具计算机专业和空间信息科学两种背景,才能很好地设计、获取对应的高性能计算并行算法/模型,这就造成了进行高性能地学计算具有一定的难度。本书着眼于从空间信息处理领域的行业应用/算法,将科学研究中遇到的实际的空间信息处理算法在现阶段流行的不同高性能计算平台、高性能计算编程手段下的不同并行计算模式进行提升精炼和整理。对集群平台上的 MPI (message passing interface) 并行计算方法、基于多核/MIC 平台的 OpenMP 并行处理方法、基于 GPU 平台的众核计算方法、基于 CPU+GPU/MIC 协同的异构计算方法、基于云计算/大数据的处理方法分别予以总结和介绍。同时,展示的算法案例也具有切实可行的通用性,将为高性能地学计算的入门者提供一种技术指导,同时也为其他应用领域进行与高性能计算技术的结合研究提供一定的借鉴。

第2章 高性能计算与高性能地学计算

在高性能计算技术发展日新月异的情况下,如何发展高性能地学计算,如何根据空间信息处理的应用选择合适的高性能计算平台、计算技术与方法,是从事高性能地学计算研究的基础问题,也是后续章节内容的铺垫。因此,本章将阐述关于高性能计算技术的相关基础知识,包括高性能计算技术的演化进程、常见的高性能计算技术、并行模型、并行算法评价指标等,并阐述一些重要的高性能地学计算研究进展。

2.1 高性能计算演化进程

随着高性能计算技术的不断进步和发展,高性能计算手段已经从传统的分布式计算(distributed computing)(Chambers, 1984)、并行集群计算(parallel computing)、网络计算发展到现在的多核计算、GPGPU、异构计算及云计算(陈国良等, 2009)。物理计算平台也从原有的多CPU、集群系统(服务器和PC)发展到GPU、MIC、异构计算平台及云平台(本质仍是集群系统)。

2.1.1 向量机、向量并行机时代

最初的计算机,只有一个计算核心,每次只能运行一个程序,直到实现了批处理功能,才能一次处理多个程序。但这里的一次,只是提交作业时的“一次”,在计算机运算时,仍然是串行处理一系列程序。然而,计算机硬件发展起来后,单独一个程序很难充分利用所有的计算资源,这样计算资源就被白白浪费了。于是进程的概念诞生了(后来又在此基础上发展出线程)。有了进程及随之而来的进程切换的概念,不仅可以充分利用计算资源,同时也出现了广义上“并行”的概念——即同时执行两个或者多个任务,当然,此时的“同时”是从宏观角度观察,在单一时间片里,仍然只有一个任务在执行(陈国良等, 2003)。

然而,计算能力总是难以跟上需求的脚步,随着工艺水平的进步,处理器的功能也在不断增强,超标量、超级流水线、超长指令字、单指令多数据流(single instruction stream and multiple data stream, SIMD)(Gove et al., 1993)、超线程、分支预测等技术不断应用于CPU当中,这些技术带来了“指令级并行”,这是并行领域最底层的概念。它通过CPU硬件的支持,让即使是单核的CPU,也能在二进制指令之内进行并行。但这种并行通常是纯硬件控制的,程序员一般无法控制,只能被动地享受科技发展带来的成果。当然,说不完全控制并不准确,因为程序员可以通过调整代码或是使用一些特殊的汇编指令间接控制CPU的行为。但最终的执行效果,仍由硬件控制。

CPU飞速发展的趋势在最近几年受到了严峻的挑战。CPU提高单个核心性能的主要手段是提高处理器的工作频率和增加指令级并行。这两种手段都遇到了问题:随着制造工艺的不断提高,晶体管的尺寸已经逐渐接近原子的大小,由此产生的漏电问题越发严重,

同时单位尺寸上的能耗与发热量也越来越多,使得处理器的频率很难像以前那样快速提高,4 GHz 的频率成为处理器厂商难以逾越的关口。另外,通用计算中的指令级并行并不多见,因此煞费苦心设计获得的性能提升与增加的晶体管数量并不成正比。

当单颗 CPU 的计算能力难以获得大幅提升时,同时使用多颗 CPU 成为科学家自然而然的想法,于是在一块主板上安装多颗 CPU 成为一种相对廉价的解决方案。但是这种方案受制于成本,仅流行于对成本和功耗并不敏感的服务器领域。可利用多 CPU 同时计算的思想,却广泛应用于高性能领域。

2.1.2 大规模并行计算时代

早在 1966 年,Michael Flynn 就提出了根据指令和数据流的计算机分类体系法:单指令单数据流(single instruction stream and single data stream, SISD)、单指令多数据流(single instruction stream and multiple data stream, SIMD)、多指令单数据流(multiple instruction stream and single data stream, MISD)和多指令多数据流(multiple instruction stream and multiple data stream, MIMD)。其中 MISD 非常罕见,而 SISD 对应着最原始的批处理机模型。SIMD 由单一制零部件控制,按照同一指令系统的要求处理不同的数据,当然 SIMD 通常描述的是硬件,而用在软件上的并行时,“单一部件”并非专门的硬件,而是一条指令。MIMD 是对大多数并行计算机而言的,多个处理器单元都是根据不同的控制流程执行不同的操作,处理不同的数据。在高性能并行计算机中,MIMD 模型最为通用, SIMD 次之。

根据 Flynn 分类法的思想(Flynn, 1966),硬件厂商处在最高端领域,不计成本制造出了超级计算机。超级计算机一般指性能上居于世界领先地位的计算机(如 TOP 500),通常有成千上万个处理器,其内存和 I/O 系统也经过专门设计,构架与个人计算机有很大区别,并非硬件的简单堆砌。即便如此,超级计算机与个人计算机的联系仍然十分紧密,如同军事/航天等高精尖行业的科技成果通常逐渐应用于民间一样(如国际互联网),超级计算机的很多技术也为普通计算机的发展提供了很大的助力。例如,一些超级计算机的 CPU 就可以直接应用于个人电脑,并且 CPU 的向量化技术、处理器封装技术等,也早已普及到普通计算机中(迟学斌 等, 2007)。

但是超级计算机的成本过高,一般研究机构无法承担。随着计算机节点之间网络互联技术的成熟,多节点间的协作也逐渐成熟起来。由于每个节点都是一台完整的计算机,因此可以通过将作业分发到不同节点,实现节点间的并行,充分利用计算资源。通过这种网络连接协作的方式,衍生出了计算机集群和分布式计算两种架构。

计算机集群将一组松散集成的计算机软件或硬件连接起来高度紧密地协作完成计算工作。在某种意义上,它们可以被看成一台计算机。集群系统中的单个计算机通常称为节点,通过局域网连接,但也有其他的可能连接方式。集群计算机通常用来改进单个计算机的计算速度和可靠性。一般情况下集群计算机比单个计算机,如工作站或超级计算机性价比要高得多。根据组成集群系统的计算机之间体系结构是否相同,集群可分为同构与异构两种。集群计算机按功能和结构可以分为,高可用性集群(high-availability clusters)、负载均衡集群(load balancing clusters)、高性能计算集群(high-performance computing clusters)(李俊照 等, 2004)。

分布式计算研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的

部分,然后把这些部分分配给许多计算机进行处理,最后把这些计算结果综合起来得到最终的结果。并行计算与分布式计算相比,并行计算借助并行算法和并行编程语言能够实现进程级并行(如 MPI)和线程级并行(如 OpenMP);而分布式计算只是将任务分成小块到各个计算机分别计算各自执行。从粒度上看,并行计算中处理器间的交互一般很频繁,往往具有细粒度和低开销的特征,并且被认为是可靠的;而在分布式计算中,处理器间的交互不频繁,交互特征是粗粒度,并且被认为是不可靠的。并行计算注重短的执行时间,分布式计算则注重长的正常运行时间。

其他厂商在不断扩展各种技术架构的同时,CPU 厂商也在不断提高 CPU 的主频,以及更改 CPU 的架构,然而思路并没有太多的改变,一颗 CPU 不能满足要求就同时使用两颗 CPU。只不过,这是随着制造工艺的进步,可以在单块芯片上集成更多的处理器核心,这样就出了多核 CPU。2005 年,Intel 和 AMD 正式向消费者市场推出了双核 CPU 产品,2007 年推出了 4 核 CPU,2009 年 Intel 又推出了 8 核 CPU。随着多核 CPU 的普及,现在普通家用电脑都拥有数个 CPU 核心,可以媲美当年一块主板上多颗 CPU 的服务器节点。当然,且不论单核性能,由于多核 CPU 是将多个核心集成在一起,因此即使是所谓的“胶水多核”,其在互联性能上也远超通过主板总线相连的 CPU 架构,更不用说如今的多核 CPU 已经做了相当多的改进,如共享 L3 Cache,其多核之间的协作与之前相比更不能同日而语。

2.1.3 异构并行计算时代

随着 CPU 从单核发展成多核,程序员也开始转变,转而关注多线程与并行编程。虽然相关理论早在 20 世纪七八十年代就已成型,而超级计算机、集群等硬件架构也已在高端领域应用。但由于硬件成本制约,绝大多数程序员只能接触到一个处理器核心,直到多核 CPU 开始普及,程序员才开始捡起尘封已久的 MPI、OpenMP 等手段,充分利用其多核计算资源。

随着人们对计算需求的不断增长,除了 CPU 性能以外,功耗也成了一大问题,于是人们又想起了 CPU 的好帮手——协处理器。从 1978 年 Intel 公司发布 8086 处理器以来,家用计算机开始普及,计算机硬件逐渐变得廉价,也使得曾经是高科技玩物的计算机逐渐应用广泛。之后有一件里程碑事件: Intel 公司在推出 8086 处理器后紧接着又推出了 8087 协处理器(相对于程序员来说是很有意义的事情:正是 8087 协处理器产生了 IEEE 754 浮点标准)。顾名思义,协处理器是协助主处理器的,必须要和中央处理器结合使用。当时 Intel 公司推出 8087 协处理器的目的是 CPU 适合对整数进行运算,对浮点支持较弱,而当时的工艺水平又无法在芯片中集成更多的晶体管,因此使用专用的芯片——8087 协处理器,专门负责浮点运算的运算需求。这件事情的意义在于协处理器从此登上了历史舞台,处理器再也不是 CPU 的专利,CPU 第一次有了帮手。虽然工艺水平有了提高,从 486DX 开始,协处理器已经集成到 CPU 中,但是协处理器的概念并没有随之消失。

从 2007 年开始,GPGPU 的普及吹响了众核协处理器回归的号角。虽然 GPGPU 的本职只是用于显示和处理图像,但 GPGPU 强大的处理浮点运算的能力,使之能够成为天然的协处理器。而作为协处理器的发明者,Intel 公司自然不会忘掉 8087 处理器,因此 Intel 公司在 2012 年推出 MIC 产品,作为新一代众核协处理器,将为高性能计算发挥更大的作用。

计算机世界的发展拥有两个关键词：需求和轮回。需求一直是推动科技发展的源动力。正如上文所述，人们需要更快地计算，因此有了计算机；需要充分利用计算机资源，因此有了多进程和多线程；由于计算能力增强，于是有了更多的计算需求，单核的处理性能不断增强；由于计算需求无止境，而单核处理能力提高受到制约，产生了各种各样多核的计算资源，如双 CPU 节点/集群、多核 CPU 等。轮回，普通说法就是循环，程序员的术语就是迭代，是计算机发展的轨迹，也是向未来发展的手段。例如，最初的 CPU 负责图像显示，后来图像越来越复杂，CPU 无法承担这一繁重的任务，于是有了 GPU，直到现在制造工艺越来越精湛，CPU 和 GPU 的发展也越来越成熟，CPU 和 GPU 再次结合在一起，如 Intel 的部分系列 CPU，AMD 的 APU，NVIDIA 公司在计划中的支持 Arm 架构的 GPU；协处理器也是如此，最初的 8086/8088，有单独的协处理器 8087，后来工艺提高，协处理器被合并入 CPU，但对计算能力要求的提高，从 PhysX 开始，协处理器再次脱离 CPU 而存在，直到现在就有作为协处理器使用的 GPGPU 和功能强大的协处理器 MIC (Satish et al., 2010)。

2.1.4 云计算与大数据时代

随着时代的发展，大数据、云计算成为当下热门的字眼。在互联网带来的“大”问题压力下，人们需要重新定义计算资源的使用方式、服务提供方式，以及社会化大生产的协作过程 (张建勋 等, 2010)。云计算扩大了人们对服务的定义，并带来了一个全新的计算资源管理思路。与此同时，大数据的发展为云计算提供了重要的应用场景。目前，云计算正处于高速发展的时期，“数据及资源”的“大数据”时代已经来临。

大数据 (big data)，指无法在一定时间范围内用常用软件工具进行捕捉、管理和处理的结构化和半结构化的数据集合，是需要新处理模式才能具有更强的决策力、洞察发现力和流程优化能力的海量、高增长率和多样化的信息资产。大数据有五个特点：大量 (volume)、高速 (velocity)、多样 (variety)、价值密度 (value)、真实性 (veracity)，其被称为 5V 特征。这五个特征反映了大数据数据量大、产生的速度快且多样，同时大数据具有低价值密度的特点，也可能会夹杂一些干扰，影响数据的真实性。因此需要一些技术手段能够快速处理海量数据，并且能够从大数据中挖掘有价值的信息。

云计算是传统计算机技术和网络技术发展融合的产物，它旨在通过网络把多个成本相对较低的计算实体整合成一个具有强大计算能力的完美系统。狭义云计算是指 IT 基础设施的交付和使用模式，通过网络以按需、易扩展的方式获得所需的资源 (如硬件、平台、软件)。提供资源的网络被称为“云”。“云”中的资源在使用者看来是可以无限扩展的，并且可以随时获取，按需使用，随时扩展，按使用付费。广义云计算是指服务的交付和使用模式，通过网络以按需、易扩展的方式获得所需的服务。这种服务可以是 IT 和软件、互联网相关的，也可以是任意其他的服务。

云计算的一个核心理念就是通过不断提高“云”的处理能力，进而减少用户终端的处理负担，最终使用户终端简化成一个单纯的输入输出设备，并能按需享受“云”的强大计算处理能力。云计算的核心思想，是将大量用网络连接的计算资源统一管理和调度，构成一个计算资源池向用户按需服务。典型的云计算提供商往往提供通用的网络业务应用，可以通过浏览器等软件或者其他 Web 服务来访问，而软件和数据都存储在服务器上。云计