



并行计算的一体化研究现状与发展趋势

陈国良, 孙广中, 徐云, 龙柏

中国科学技术大学计算机科学与技术系, 国家高性能计算中心(合肥), 安徽省高性能计算重点实验室, 合肥 230027

E-mail: glchen@ustc.edu.cn

2008-12-02 收稿, 2009-03-02 接受

国家自然科学基金资助项目(批准号: 60533020, 60873210)

摘要 在过去的 20 多年里, 中国科学技术大学的研究团队逐渐形成了并行计算“结构-算法-编程-应用”一体化的研究体系, 也就是所谓的并行计算研究的生态环境. 该文就并行计算的一体化研究现状进行简要的综述, 并结合多核系统、云计算、个人高性能计算机等对于并行计算一体化研究的影响, 概要的展望了其发展趋势.

关键词

并行计算

一体化研究

多核

云计算

个人高性能计算机

1 并行计算简介

1.1 并行计算的基本概念

并行计算是计算机科学中重要的研究内容, 已有几十年的发展历程.

用并行计算求解问题的大致过程为: 对于一个给定的应用问题, 首先, 计算科学家将这个应用转化为一个数值或非数值的计算问题; 然后计算科学家对此计算问题设计并行算法, 并通过某种并行编程语言实现它; 最后应用领域的专家在某台具体的并行计算机上运行应用软件求解此问题. 由此, 我们可以很自然的发现并行计算由以下几个部分构成: 并行计算机(并行计算的硬件平台), 并行算法(并行计算的理论基础), 并程序序设计(并行计算的软件支撑), 并行应用(并行计算的发展动力). 这些部分涉及到许多方面的研究者, 包含了计算数学家、计算机科学家、软件工程师和应用领域专家等^[1].

计算科学的发展使得“计算科学”已经与传统的“理论科学”和“实验科学”并列成为推动科技发展和社会文明进步的三大科学. 同时, 并行计算作为一种研究的工具, 也逐渐融入到传统的“理论科学”和“实验科学”中, 如化学家哈姆弗雷·戴维爵士(Sir Humphrey Davy) 曾经明智地指出: “没有什么比应用新工具更有助于知识的发展. 在不同的时期, 人们的

业绩不同, 与其说是天赋智能所致, 倒不如说是他们拥有的工具特性和软资源(非自然资源)不同所致”. 美国总统信息技术咨询委员会(PITAC)在致布什总统的“计算科学: 确保美国竞争力”的报告中, 有这么一段描述: “虽然计算本身也是一门学科, 但其具有促进其它学科发展的作用. 21 世纪科学上最重要的和经济上最有前途的研究前沿, 有可能通过熟练掌握先进的计算技术和运用计算科学得到解决”^[2].

并行计算(parallel computing), 简单的说, 就是在并行计算机上所做的计算, 它和常说的高性能计算(high performance computing)、超级计算(super computing)是同义词, 因为任何高性能计算和超级计算总离不开并行技术.

串行计算是指在单个计算机(具有单个中央处理单元)上执行软件读写操作, 逐个使用一系列指令解决问题. 并行计算是在串行计算的基础上演变而来, 它努力仿真自然世界中, 一个序列中含有众多同时发生的、复杂且相关事件的事务状态. 为利用并行计算, 通常计算问题表现为以下特征: 将计算任务分解成多个部分, 有助于同时解决; 在同一时间, 由不同的执行部件执行多个程序指令; 多计算资源下解决问题的耗时要少于单个计算资源下的耗时. 并行计算可分为时间上的并行和空间上的并行: 前

者典型代表为流水线技术,后者则用多个处理器同时的执行计算^[3]。

1.2 并行计算的一体化研究方法

并行计算早期的发展历程:从20世纪40年代开始的现代计算机发展历程可以分为两个明显的发展时代:串行计算时代和并行计算时代。每一个计算时代都从体系结构发展开始,接着是系统软件(特别是编译器与操作系统)、应用软件,最后随着问题求解环境的发展而达到顶峰。

创建和应用并行计算的主要原因是因为并行计算是解决单处理器速度瓶颈的最好方法之一。而并行计算的硬件平台是并行计算机,它由一组处理单元组成,这组处理单元通过相互之间的通信与协作,以更快的速度共同完成一项大规模的计算任务。因此,从并行计算的角度来看,并行计算机的两个最主要的组成部分是计算节点和节点间的通信与协作机制。并行计算机体系结构的发展也主要体现在计算节点性能的提高以及节点间通信技术的改进两方面。

20世纪60年代初期,由于晶体管以及磁芯存储器的出现,处理单元变得越来越小,存储器也更加小巧和廉价。这些技术发展的结果导致了并行计算机的出现,这一时期的并行计算机多是有一定规模的所谓大型主机(mainframe)。IBM360是这一时期的典型代表。与此相对应的,Fortran语言作为主要的并行机上的编程语言,得到快速的发展。而并行机的应用以数值计算为主,并行算法的研究也以经典的数值并行算法(如快速傅里叶变换等)为主^[4]。

到了20世纪60年代末期,同一个处理器开始设置多个功能相同的功能单元,流水线技术也出现了。与单纯提高时钟频率相比,这些处理器内部的并行特性大大提高了并行计算机系统的性能。伊利诺依大学和Burroughs公司此时开始实施IlliacIV计划,研制一台64个CPU的SIMD主机系统,它涉及到硬件技术、体系结构、I/O设备、操作系统、程序设计语言直至应用程序在内的众多研究课题,已经反映出并行计算一体化研究的需求。

并行计算的现今研究状况:并行计算的研究面很广,主要包括了并行计算的硬件平台(即并行计算机)、并行计算的软件支撑(即并行程序设计)、并行计算的理论基础(即并行算法)以及并行计算的具体应用。由于长期以来并行计算的研究缺乏良好的研究

方法,导致并行计算的研究出现了参差不齐的局面。目前的并行计算研究主要存在以下几个方面的问题:

近年并行算法本身的研究呈现低调的局面;大部分的并行应用的效率依然较低,无法高效的使用并行计算机的资源;并行编程语言对于一般的编程人员还不够简单易行,也缺乏高效的并行编程环境工具;并行计算机本身的构建也存在着能耗过大、管理困难、可扩展性差等方面的问题。

我国的并行计算研究和国际走向大致相同,自20世纪60年代末至今,已经历经三个阶段:第一阶段,60年代末至70年代末,主要从事大型机中的并行处理技术研究;第二阶段,70年代末至80年代初,主要从事向量机和并行多处理器系统研究;第三阶段,80年代末至今,主要从事MPP(massively parallel processor)系统及工作站集群系统研究。

尽管我国在并行计算方面开展的研究和应用较早,目前也拥有很多的并行计算资源,但研究和应用的成效相对美日等发达国家还存在较大的差距,有待进一步的提高和发展。因此,我们需要形成一体化并行计算研究体系和方法来解决目前的困境。

并行计算一体化的研究方法:为了解决上面所提到的并行计算研究中存在的问题,我们亟待建立一个完整的科学研究体系。并行计算研究历程,既有高潮也有低谷,究其原因,它没有形成自身的一套研究方法学。我们在文献^[5]中已提出了一套并行算法的研究方法学,即“理论-设计-实现-应用”的研究体系,也就是所谓的并行算法研究的生态环境。其中,算法理论包括并行计算模型和并行计算复杂性等;算法设计包括并行算法的设计和并行算法的分析等;算法实现包括软件支撑和硬件平台等;并行应用包括科学工程计算应用和社会计算应用等。

并行计算作为并行算法的一个超集,更加需要建立“结构→算法→编程→应用”的一体化研究方法,这样才能稳定的可持续发展并且变得更加实用。我们认为:研究并行计算的人一定要懂并行计算机,能够设计并行算法,要清楚地知道如何在并行计算机上用合适的编程语言来实现,从而解决实际的应用问题。如图1所示,并行计算的结构部分包含了高端的高性能计算机和低端的普及型计算机;并行计算的算法部分包含了并行算法的设计与分析以及算法库和测试库;并行计算的编程部分包含了并行编程模型以及并行编程的环境工具;并行计算的应用部

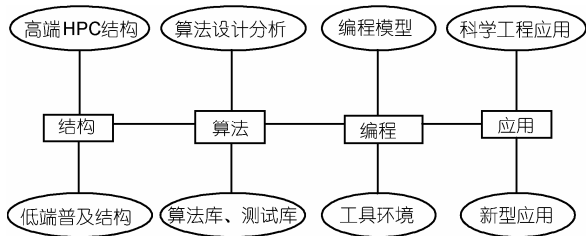


图1 并行计算一体化研究方法

分包含了科学工程应用和各种新型的应用。

自20世纪80年代初开始,中国科学技术大学研究团队开始进行并行计算方面的研究,从单纯的进行并行算法研究,逐步扩展到同时进行并行算法、结构、编程与应用等的研究工作^[6~10]。通过20多年的努力和积累,逐渐建立了如图1所示的并行计算的“结构-算法-编程-应用”一体化研究方法。这样才能确保并行计算研究的生态环境的可持续发展,为并行计算研究的可持续发展提供了保证,也为面对不断快速发展的技术发展的研究者们提供了并行计算研究的有益指导。

2 并行计算近期的若干新发展

并行计算的发展和计算机学科中许多领域一样,除了学科发展的自身规律外,也受到业界的很大影响。近期,随着硬件技术和新型应用的不断发展,并行计算也有了若干新的发展,如多核体系结构、云计算、个人高性能计算机等。

2.1 以多核为主流的体系结构

过去的30多年,中央处理器的发展主要来自于工艺的提高和体系结构的不断发展。工艺的提高使得晶体管面积减小,不断增大集成度;体系结构的发展同样大大推动了性能的提高,如深度流水、指令级并行等。但整体的系统结构上都还是串行的结构,部分使用了并行的技术。最近几年来,随着芯片集成规模极限的逼近,以及能耗和成本的因素,具有多核结构的产品逐渐成为市场的主流^[11]。

所谓多核技术即在同一个处理器中集成两个或多个完整的计算内核,每个计算内核实质上都是一个相对简单的微处理器。多个计算内核可以并行地执行指令,从而实现一个芯片内的线程级并行,并可在特定的时间内执行更多任务实现任务级并行,从而提高计算能力。

一般认为,多核结构具有良好的性能潜力和实

现优势:多核结构将一个复杂的功能芯片划分成多个处理器核来设计,每个核都比较简单,有利于设计的优化;多核结构能够有效地利用了芯片内的资源,可以有效的利用程序的并行性,带来性能的快速提升;处理器核之间的互连缩短,提高了数据传输带宽,可以有效地共享资源,同时降低芯片的功耗。

多核系统在学术界的探讨和研究具有较长的历史,在20世纪90年代初就已有初步的研究工作,但并未成为业界的主流产品。近年来,随着主流芯片生产商将多核系统作为它们的主要产品推出,使得多核的技术和产品逐渐成为主流。全球主要芯片生产商纷纷推出自己的多核通用微处理器。第一个商用的多核通用微处理器是IBM于2001年发布的Power4处理器。每个Power4芯片中集成了两个64位的1GHz的PowerPC核,可以并行执行200条指令。HP公司于2003年推出类似的多核处理器PA-RISC 8800,它在一块芯片上集成了两个主频为1GHz的PA-RISC 8700。Sun公司于2004年推出了自己的多核处理器UltraSparc IV,一块芯片内集成了两个UltraSparc III核心。Intel和AMD也分别于2005年推出了各自的商用双核微处理器Pentium D和Opteron。

目前Intel公司和AMD公司都推出了自己的多核处理器。双核和四核处理器目前已经投入市场。从公司的市场导向来看,他们还计划在2009年到2010年陆续推出八核的处理器。服务器和工作站传统上都是使用双路处理器,这就意味着到2010年底每个主板上处理核心的总数量能够达到16个。另外,AMD和Intel处理器都提供四路甚至八路设计,不久的将来六十四核服务器也有可能出现。

具体到多核结构发展的具体趋势,还是存在着不同的预测,业界和学术界目前依然存在着不同的观点。不过,多核系统是未来计算系统的主流已经成为大家的共识。

2.2 以数据为中心的云计算

云计算(cloud computing),是指基于当前已相对成熟与稳定的互联网的新型计算模式,即把原本存储于个人电脑、移动设备等个人设备上的大量信息集中在一起,在强大的服务器端协同工作。它是一种新兴的共享计算资源的方法,能够将巨大的系统连接在一起以提供各种计算服务。很多因素推动了对这类环境的需求,其中包括互联网的发展与成熟、移动设备的发展、搜索引擎的普及、社会网络和移动商务

等。另外,各种数字设备的大规模发展和普及也使以数字形式存储的信息的规模大幅度增长,从而进一步加强了对一个由统一的强大的服务器进行管理的需求^[12]。

云计算可以看作是分布式计算(distributed computing)、并行计算(parallel computing)和网格计算(grid computing)的最新发展。它的产生和成长来自于业界的需要和推动,已经得到 IBM、Google、微软、雅虎、SUN 等全球主要信息技术公司的支持。

云计算意味着对于服务器端的并行计算要求的增强,因为数以万计用户的应用都是通过互联网在云端来实现的,它在带来用户工作方式和商业模式的根本性改变的同时,也对大规模并行计算的技术提出了新的要求。

云计算还在不断的发展过程中,在信息存储与挖掘、信息安全等方面的研究是相对重要的方向。

2.3 以普及应用为宗旨的个人高性能计算机

个人高性能计算机首先是高性能计算机,应具有高性能计算机的基本特征,即高性能。它比普通的工作站或PC以及网络计算机,在浮点运算速度、能够处理的数据集大小、I/O性能、数据交换性能和同步性能等方面,要好一个数量级以上^[13]。

个人高性能计算机的主要特征体现在“个人”上,既普及型计算设备应具备的特性。从大型计算机向面向个人的 PC 转变过程中发生的 4 个主要变化,也是个人高性能计算机应该有的转变,具体为: 办公室使用环境(office using environment): 高性能计算机是机房使用环境,而 PC 是办公室使用环境,办公室环境对计算机的体积、重量、电源插座、噪音、散热条件都有与机房环境截然不同的要求。 规模产品(volume product): 高性能计算机是一种对制造、维护、服务要求很高的高端产品,在产品的全生命周期都需要人工干预;而 PC 是规模产品,规模产品对计算机的成本、可靠性、量产能力、管理、维护、服务、以及应用面等都有着不同的要求。 用户为中心的使用模式(user-centric using model): 高性能计算机是以机器为中心的使用模式,强调提高多用户下资源的利用率、强调集中管理和作业调度能力;而 PC 是以用户为中心的 GUI 式使用模式,强调独占,强调对资源、数据、应用性能的可控,强调个性化的设置。

高生产率编程(high productivity programming): 高性能计算机采用以性能为中心的编程模式,无论

是 OpenMP、HPF、MPI, 还是 UPC (Universal Parallel C)、IBM X10 都是主要强调性能,没有方便用户的抽象层;而 PC 是以程序员生产率为中心的编程模式,无论是 VB、VC, 还是 Java、C#, MATLAB 都是面向大众的编程语言,易编程是第一位的,性能多靠一些专门的库来实现。

综合个人高性能计算机面向普及和面向高性能计算两方面的需求,应该具有 8 个可归纳为“3 低-2 高-3 易”的主要特征: 低成本、低功耗、低噪音、高效能、高可靠、易编程、易管理、易应用。

2007 年底,中国科学技术大学和中国科学院计算技术研究所基于龙芯 2F 的国产万亿次高性能计算机KD-50- 的研制成功,是高性能计算机向个人化方向发展这一理念的首次尝试^[14]。至 2008 年底,中国科学技术大学又研制了两个KD-50- 的增强型机器,即KD-50-I-E,主要应用于交通信息处理和雷达遥感信息的处理等。

3 并行计算面临的新挑战

多核技术的出现与主流化,对于并行计算体系结构、并行算法、并行程序设计与并行应用的研究都分别产生了重要的影响,带来了新的挑战。

3.1 多核化的并行计算机

多核化趋势正在改变并行计算的面貌。跟传统的单核CPU相比,多核CPU带来了更强的并行处理能力、更高的计算密度和更低的时钟频率,并大大减少了散热和功耗。目前,在几大主要芯片厂商的产品线中,双核、四核甚至八核CPU已经占据了主要地位。在将应用从单核环境向多核系统迁移的过程中,通过选择合适的操作系统,应用开发人员可以大大地减少麻烦。与此同时,多核技术的应用也带来了诸多问题^[15]。

首先,多核芯片如何管理共享资源就是一大问题。在大多数情况下,多核 CPU 的内核拥有独立的 L1 缓存,共享 L2 缓存、内存子系统、中断子系统和外设。因此,系统需要让每个内核独立访问某种资源,并确保资源不会被其他内核上的应用程序争抢。多核的出现还让系统设计变得更加复杂。如运行在不同内核上的应用为了互相访问、相互协作,需要进行一些独特的设计,如高效的进程间通信(interprocess communication)机制、共享内存的数据结构和同步原语(synchronization primitives),也就必然导致了混合

式体系结构的出现。

其次,就是并行计算机系统的高效能问题(high efficiency)。一般来说,人们往往比较关注并行计算机的高性能(high performance)。但是,随着多核技术和并行计算的发展,高效能计算机被提上了历史日程。所谓高效能计算机也就是高“生产力”的计算机。比如,通过简单的将许多 PC 机连接起来构成的机群系统虽然峰值很高,但存在着通讯瓶颈,不能认为是高效能的机器。

再次,它对并行计算系统的算法和编程带来了很大的困难。程序代码迁移(code migration)也是个问题。大多数系统厂商都在针对单核 CPU 架构的代码库上进行了大量投资。因而,这些公司需要有一个清晰的迁移策略,来使其代码可以最大化地利用多核硬件资源,同时为之付出的迁移代价要尽可能的小。

3.2 大规模的并行算法

并行算法是传统并行计算研究中的重要内容,在新的条件下,并行算法的研究也同样受到了很大的影响。

首先,多核系统的发展对于并行计算模型的研究提出了新的挑战。计算模型是为了计算目的将真实体(计算机软/硬件)进行一定的抽象而成的。并行计算模型是算法设计者所看到的参数化了的并行机,是提供给编程者的计算机软/硬件接口,是程序执行时的系统软/硬件支撑环境。传统的并行计算模型一般指的是并行算法设计模型,为并行算法的研究者提供一个独立于具体并行机体系结构的抽象的并行机。一般模型中通常定义了机器参数、计算行为、开销函数,它们被称为计算模型三要素^[16]。随着多核系统等并行结构的飞速发展,为了使计算模型能够反映体系结构的变化,人们不断的向该单一模型中加入旨在反映机器特性的新的参数,调整计算行为,修改开销函数。单一模型变得越来越复杂,在实际的算法设计中难以使用,所以需要通过分层的并行计算模型来解决^[17]。

其次,多核时代对于并行算法的可扩展性的要求有了更多的要求。可扩展性(scalability)评测标准,包括等效率度量标准、等速度度量标准和平均延迟度量标准等。这些指标对于新的多核系统,如何进行建模和计算,成为了很有趣的研究问题。这里主要考虑的因素是如何根据问题和系统来刻画计算中固有的串行部分。例如,对于共享存储的多核系统,存储访

问的瓶颈可以视为是不可并行化的串行分量^[18]。

再次,由于多核系统和云计算的发展,针对海量数据的计算成为并行算法研究中越来越重要的部分。与传统的科学计算不同,针对海量数据的计算不再局限于少量典型的计算算法(如方程组求解,快速傅立叶变换,特征值求解等)的研究,众多与数据相关的一些更为丰富的非数值算法的并行化方法成为了研究和关注的重点^[19]。

3.3 趋向普及的并行编程

随着当前并行计算技术的发展,并行编程日益普及,在可以预见的未来,大量的应用开发工程师将撰写并程序,这对于并行编程的研究也提出了新的要求^[20]。

首先,多核技术的发展促进了并行编程的普及。多核处理器的出现,使得物理上共享存储的并行处理平台的门槛大大降低,目前购买一台四内核系统的价格仅为传统 4CPU 系统价格的 1/5 至 1/10。这个改变将使得并行计算的用户范围大大扩展,从传统的科学与工程应用领域和事务处理领域扩展到桌面与移动计算领域。同时,由于多核这一并行结构已经成为桌面系统通用处理器的主流,使得众多应用程序员也必须学习并行计算,必须从事并行计算的开发。

其次,多核技术的发展促进了并程序设计的进步,并行编程的易用性成为重要的考虑因素。目前流行的并行编程模型可以分为消息传递和共享内存两类。MPI 是消息传递模型的标准,OpenMP 是主流的共享内存模型,得到了几乎所有商业编译器的支持,具有很好的可移植性。此外还有一些应用直接采用操作系统或程序设计语言提供的多线程 API 进行共享内存并行编程,如 UNIX/Linux 的 PThread 库,Windows 的线程和 Java 线程。一般来说,科学与工程计算领域以及网络服务应用领域原有的并程序基本可以顺利移植到多核体系结构,但是在桌面与移动计算领域,大量现存应用都是串行的,其开发者也不熟悉并程序的开发,从而需要对现有应用进行并行化和开发新的并行应用程序。

再次,对于并行编程的工具环境提出了新的挑战。需要从事并行编程的人员不再是少数高端的科研人员,使得对于并行编程语言、环境、工具的要求也更高。目前的并行应用软件却很难完全发挥多核处理器的性能,从编译器、开发语言、函数库到开发

工具, 软件并未作好迎接多核的准备. 由于编译器和带有标准化接口的库文件存在, 程序员们到现在仍然独立于硬件设计之外. 更严峻的是, 在转向新一代硬件时, 程序员需要向多核系统移植现有应用程序. 为了充分发挥新设计的优势, 程序员必须在应用中提取并明确表述并行性(任务或数据并行性). 为了实现最高的峰值性能利用率, 程序员还必须将应用参数化, 以应对不同的 Cache 大小和 Cache 分级体系, 并增加新的代码来获取最佳 NUMA 效果. 另外当前和今后的大型高性能计算系统都必须采用最新的多核处理器, 这样在研制高性能计算系统时就多了一层并行性要考虑, 除了节点内和节点间的并行性外, 还有芯片内的并行性要考虑.

3.4 新型的并行应用

一直以来并行计算的应用均展现了其高端性, 多用于大型的科学计算领域. 随着互联网的飞速发展和多核技术的出现, 并行计算开始向低端的个人高性能机领域发展, 即所谓的桌面应用. 它使得普通用户也可能要面对并发和并行操作, 这些通常是并行计算专业人员和高端用户才需要面对的问题. 以下就相关的并行计算应用加以简述.

首先, 计算密集型的应用依然是并行计算的主要方面. 并行计算在科研方面的主要应用包括以下几个领域: 计算生物学、计算化学、计算流体力学、飞行动力学、计算机辅助设计、数据库管理、油藏建模、中长期天气预报、海洋环流和求解N-body问题等以及面向应用的大型科学与工程问题的并行数值计算, 诸如求解大型稀疏方程组、大型非线性方程和有限元分析等. 在交叉学科中出现了一些非传统计算方式, 例如分子计算和量子计算等. 这些计算本身包含着巨大的并行度, 超出了传统的计算模式, 但目前都还处于研究探索阶段. 相信在可预期的未来, 高性

能计算依然是处理大规模科学计算的重要工具^[21].

其次, 随着并行计算的最新发展, 新型的以数据密集为特征的并行应用也日益成为主流. 在服务器端的最典型应用的就是大规模的商用搜索引擎. 并行计算是大规模搜索引擎系统的核心技术, 如 Google 公司的 GFS/MapReduce/Bigtable^[22~24], 微软公司的 Dryad^[25] 等. 对于每时每刻数以万计的查询请求, 我们很难想像离开了并行技术, 搜索引擎的处理速度会慢到何等的程度. 而对于处理互联网上的海量数据, 搜索引擎也需要并行的技术来处理大规模数据密集的应用, 这里容错与高效的系统管理也成为了需要考虑的重要因素. 并行处理的商业应用在计算能力方面可能不像科学和工程计算要求那么强大, 但它们也更需要大的存储和磁盘容量以及高的 I/O 传输率. 典型的商务应用还有大规模数据库的管理和查询、在线事务处理、数据仓库、数据挖掘和决策支持系统等.

再次, 随着多核系统的普及, 桌面应用和移动应用也成为并行计算新兴的重要应用. 由于个人计算机和移动设备都具有多个处理单位, 使得许多桌面应用和移动设备上的应用也不得不采用并行计算的技术. 这对于许多传统桌面应用领域的研究者, 以及并行计算的研究者, 都是一个新的机遇与挑战.

4 结论

并行计算“结构-算法-编程-应用”一体化的研究方法是华中科技大学并行计算研究团队 20 余年来通过不断的研究工作, 在实践中逐渐总结出来的. 虽然并行计算技术不断的得到发展, 一体化的研究方法依然具有其指导价值. 本文结合一体化的研究方法, 以及并行计算的最新发展, 提出了在“结构-算法-编程-应用”等几个方面, 并行计算研究面临的机遇和挑战, 并就并行计算的发展趋势做出了展望.

致谢 感谢 2008 年 9 月参加“2008 年国际并行算法、结构、编程研讨会”的各位专家, 他们的讨论和评论为本文提供了许多帮助.

参考文献

- 1 Chen G L, Sun G Z, Zhang Y Q, et al. Study on parallel computing. J Comput Sci Tech, 2006. 21(5): 665—673[doi]
- 2 陈国良. 并行计算: 结构·算法·编程. 北京: 高等教育出版社, 2003
- 3 Grama A, Gupta A, Karypis G, et al. Introduction to parallel computing. Boston: Benjamin/Cummings Publish Company, Inc., 2003
- 4 陈国良. 并行算法的设计与分析. 北京: 高等教育出版社, 2002

- 5 陈国良, 孙广中, 徐云, 等. 并行算法研究方法学. 计算机学报, 2008, 31(9): 1493—1502
- 6 Chen G L. A partitioning selection algorithm on multiprocessors. J Comput Sci Tech, 1988, 3(4): 241—250[doi]
- 7 陈国良, 梁维发, 沈鸿. 并行图论算法研究进展. 计算机研究与发展, 1995, 32(9): 1—16
- 8 安虹, 陈国良. 并行程序设计模型和语言. 软件学报, 2002, 13(1): 118—124
- 9 毛睿, 黄刘生, 徐大杰, 等. 淮河中上游群库联合优化调度算法及并行实现. 小型微型计算机系统, 2000, 21(6): 44—48
- 10 Zhang F, Chen G L, Zhang Z Q. OpenMP on Networks of Workstations for Software DSMs. J Comput Sci Tech, 2002, 17(1): 90—100[doi]
- 11 Sutter H, Larus J. Software and the concurrency revolution. Q focus: Multiprocessors, 2005, 3(7): 54—62
- 12 Rajkumar B, Chee S Y, Srikumar V. Market-oriented cloud computing: Vision, hype, and reality for delivering IT services as computing utilities. In: Proceedings of the 10th IEEE International Conference on High Performance Computing and Communications, 2008 Sept 25-27, Dalian. Los Alamitos, CA: IEEE CS Press, 2008. 15—22
- 13 孙凝晖, 陈国良. PHPC: 一种普及型高性能计算机. 中国科学技术大学学报, 2008, 38(7): 745—752
- 14 张俊霞, 张焕杰, 李会民. 基于龙芯 2F 的国产万亿次高性能计算机 KD-50- I 的研制. 中国科学技术大学学报, 2008, 38(1): 105—108
- 15 Asanovic K, Bodik R, James J, et al. The landscape of parallel computing research: A view from Berkeley. Technical Report, Electrical Engineering and Computer Sciences, University of California, Berkeley. 2006
- 16 Zhang Y Q, Chen G L, Sun G Z, et al. Models of parallel computation: a survey and classification. Front Comput Sci China, 2007, 1(2): 156—165[doi]
- 17 陈国良, 苗乾坤, 孙广中, 等. 分层并行计算模型. 中国科学技术大学学报(50 周年校庆专刊), 2008, 38(7): 841—847
- 18 Sun X H. Scalable computing in the multicore era. In: Proceedings of the Inaugural Symposium on Parallel Algorithms, Architectures and Programming, 2008 Sep 16-18, Hefei. Hefei: University of Science and Technology of China Press, 2008. 1—18
- 19 Chu C H T, Yu Y Y, Bradski G, et al. Map-reduce for machine learning on multicore. In: Twentieth Annual Conference on Neural Information Processing Systems, 2006 Dec 4-7, Vancouver. Boston: MIT Press, 2006. 281—288
- 20 Mattson T G, Sanders B A, Massingill B L. Patterns for Parallel Programming, New Jersey: Prentice Hall, 2005
- 21 Dongarra J, Fox G, Kennedy K, et al. Sourcebook of Parallel Computing. San Francisco: Morgan Kaufmann Publishers Inc. 2003
- 22 Dean J, Ghemawat S. MapReduce: Simplified data processing on large clusters. In: Sixth Symposium on Operating System Design and Implementation, 2004, Dec 6-8, San Francisco, CA. Berkeley: USENIX Association, 2004. 10—23
- 23 Ghemawat S, Gobioff H, Shun-Tak L. The Google file system. In: Proceedings of the Nineteenth ACM Symposium on Operating Systems principles, 2003 Oct 19-22, New York. New York: ACM Press, 2003. 29—43
- 24 Chang F, Dean J, Ghemawat S, et al. Bigtable: A distributed storage system for structured data. In: Seventh Symposium on Operating System Design and Implementation, 2006 Nov 6-8, Seattle. Berkeley: USENIX Association, 2006. 205—218
- 25 Isard M, Yu Y, Birrell A, et al. Dryad: Distributed data-parallel Programs from Sequential Building Blocks. Technical Report, Microsoft Research Technical Report, Microsoft Corporation, 2006