结构网格 CFD 应用程序在天河超级计算机上的高效并行与优化

王勇献[1,2],张理论[1,2],车永刚[1,2],徐传福[1],刘 巍[1],程兴华[1]
(1.国防科技大学计算机学院,湖南长沙 410073;2.国防科技大学并行与分布处理重点实验室,湖南长沙 410073)

摘 要: 对多区结构网格大规模 CFD 流场模拟的高效并行方法进行了研究,以天河超级计算机平台的 CPU 同构计算环境和 CPU + MIC 异构计算环境为例,重点讨论了 CFD 应用特点与超级计算机运行环境相适应的性能优化与改进策略,发展了一系列多层次并行与性能优化方法.通过在天河 2 高性能计算平台上进行了多个算例的数值模拟,验证了这些优化方法的并行效果;在 CPU + MIC 异构平台上模拟的最大 CFD 问题规模达到 6800 亿个网格单元,共使用 137.6 万 CPU + MIC 处理器核,测试结果表明在 CPU + MIC 异构平台上移植优化后的程序性能提高 2.6 倍左右,且具有良好的可扩展性.

关键词: 计算流体力学; 多区结构网格; 并行计算; 天河计算机; CPU + MIC 异构计算
中图分类号: TP301 **文献标识码:** A **文章编号:** 0372-2112 (2015)01-0036-09
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.01.007

Efficient Parallel Computing and Performance Tuning for Multi-Block Structured Grid CFD Applications on Tianhe Supercomputer

WANG Yong-xian[1,2], ZHANG Li-lun[1,2], CHE Yong-gang[1,2], XU Chuan-fu[1], LIU Wei[1], CHENG Xing-hua[1]
(1. *College of Computer, National University of Defense Technology, Changsha, Hunan* 410073, *China*; 2. *Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha, Hunan* 410073, *China*)

Abstract: How to improve the parallel performance of CFD applications with typical multi-block structured grid based on the CPU sub-platform of Tianhe-1A and CPU + MIC co-processor heterogeneous platform of Tianhe-2 supercomputer system, is focused in this paper. Some strategies of performance optimization matched with both the characteristic of CFD application and the architectures of high-performance computing (HPC) platform are discussed in detail. Some numerical experiments are performed on Tianhe-2 supercomputer system with the maximum of grid cells achieving 6.8×10^{11}, and the total amount of processors and/or co-processors being 1.376×10^6. It shows that the optimized code can get a speedup of 2.6 times faster on CPU and co-processor hybrid platform than that on the CPU platform only, and good scalability is also observed from the test results.

Key words: computational fluid dynamics; multi-block structured grid; parallel computing; Tianhe supercomputer; CPU + MIC heterogeneous computing

1 引言

近年来,随着计算流体力学(CFD)方法的不断突破和计算机技术的快速发展,基于 CFD 的数值模拟方法开始越来越多地被应用到航空航天飞行器的研究和设计当中.为了提高 CFD 数值模拟的计算规模、计算效率,CFD 数值模拟代码通常需要并行计算,以便充分利用高性能计算机的强大并行处理能力.传统的并行 CFD 程序主要采用区域分解方式,根据高性能计算机体系结构的特点,可以实现共享存储或消息传递并行程序,不同区域在不同进程或线程上运行相同的求解器,获得每个时间步的流场结果后需要对区域之间的边界进行信息交换.

在利用 CFD 进行大规模流场的并行数值模拟过程中,需要综合考虑 CFD 应用特征和高性能计算机体系结构的特点,以获得最佳的模拟性能.近年来,以异构众核为特征的高性能计算日渐成为主流.本文试图以天河超级计算机及自有 CFD 应用为例,探索 CFD 应用软件

收稿日期:2013-03-29;修回日期:2014-07-12;责任编辑:孙瑶
基金项目:国家自然科学基金(No.61379056,No.11272352);空气动力学国家重点实验室基金(No.SKLA20130105)

与高性能计算平台的最优适配策略,研究异构众核平台下典型结构网格 CFD 应用的并行优化方法,为同类应用问题的并行移植与性能优化提供借鉴与参考.

2　CFD 应用求解流程与高性能计算平台

2.1　流体控制方程与经典 CFD 求解流程

本文使用经典三维非定常 Navier-Stokes 控制方程对流场进行数值模拟,以来流速度 V_∞、密度 ρ_∞ 以及特征长度 L 为特征量进行无量纲化,在曲线坐标系(ξ,η,ζ)下微分形式的控制方程组可写为:

$$\frac{\partial Q}{\partial t}+\frac{\partial(E-E_v)}{\partial \xi}+\frac{\partial(F-F_v)}{\partial \eta}+\frac{\partial(G-G_v)}{\partial \zeta}=0 \quad (1)$$

其中 Q 为基本物理量(原始量或守恒量),E,F,G 分别为 ξ,η,ζ 三个方向上的对流通量,E_v,F_v,G_v 分别为三个方向上的粘性通量.为方便叙述,方程组(1)中等号左边的四项分别称为时间导数项和三个方向上的空间通量导数项,其中每个方向上的空间通量导数项又进一步分成对流(通量导数)项和粘性(通量导数)项.Q,E,F,G,E_v,F_v,G_v 的具体形式参看文献[1~3],对于三维流场而言,它们都是具有 5 个分量的向量.

为了数值求解连续控制方程(1),本文使用有限差分方法对空间项进行离散,其中对流项采用邓小刚等提出的加权紧致非线性格式差分离散格式,粘性项采用中心差分离散格式[1,2].当流场由多个网格区块组成时,为了保证高阶精度的需求,相邻的多个区块所共享的网格点(称为拓扑结构奇异点)上,其物理量及各阶导数的计算需要经过更复杂的处理[4].时间离散采用经典的隐式方法,最终得到了多时间步的离散方程组,并采用隐式双时间步 LU-SGS 迭代方法或者显式的多步 Runge-Kutta 方法等进行求解.整个 CFD 求解流程如图 1 所示.

2.2　众核异构型高性能计算平台上的 CFD 并行模拟

当前的高性能计算机平台多数采用大型集群或大规模并行机系统(MPP,Massively Parallel Processing)的体系结构,近几年以“主处理器 + 加速器或协处理器”方式进行协同工作的异构型计算机更日益成为主流[5,6].加速器或协处理器的典型代表包括较早出现的 GPU 加速器、以及新出现的至强融核协处理器(Intel Xeon Phi,或简称 MIC)等,其共同特点是集成了更多的轻量级计算核心,具有较高的性能加速比.尽管在传统高性能计算机上进行大规模 CFD 应用数值模拟已经得到了充分的研究[7~9],但对新型众核异构型计算平台上的 CFD 并行与优化技术的研究还比较零散,且目前主要集中在 CPU + GPU 的协同并行方面[10,11];由于 CPU + MIC 异构型高性能平台尚不多见,例如在最新的 2013 年 12 月全球超级计算机 500 强排名中只有 13 台为该混合架构(包括排名第一的天河 2 超级计算机)[5],因此在这类异构平台上开展超大规模 CFD 应用问题研究更为少见.2013 年 Subhash Saini 等在一个 128 结点的小型 CPU + MIC 异构平台上对一个测试程序及两个 CFD 应用进行了小规模初步性能测试,其中,应用级的测试主要限于使用单个计算结点的情形[12].同年,王勇献等基于刚问世的天河 2 计算机平台,将传统结构网格 CFD 应用模拟初步移植到 CPU + MIC 异构平台上,并在 3072 个计算结点上进行了性能测试[13].本文中,我们在前期大规模并行优化方法研究的基础上,发展了一系列应用特征与硬件架构特点相适应的并行与优化方法,使得 CFD 应用的异构并行模拟可取得更好的可扩展性.

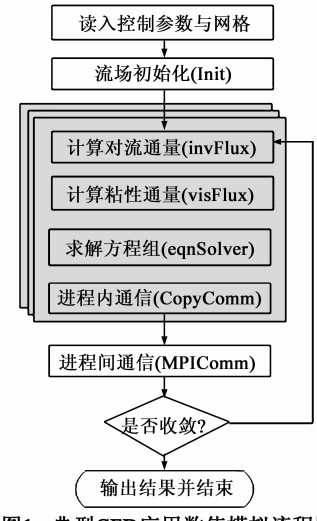


图1　典型CFD应用数值模拟流程图

3　大规模 CFD 模拟中的并行与优化技术

3.1　CFD 计算特征与系统架构相适应的多层次并行策略

为了适应硬件结构特点、充分发挥硬件效率,CFD 并行模拟也要采取多层次、多粒度的混合并行策略,以充分发掘程序的并行度,提高 CFD 模拟的可扩展性.对于同构系统通常采用“MPI 进程 + OpenMP 线程”混合并行策略;对于异构系统,则需要进一步采用“CPU + 加速器或协处理器”的异构编程模型,例如对于 GPU 加速器使用 MPI + OpenMP + CUDA 编程模型,对于 MIC 协处理器可使用 MPI + OpenMP + Offload 方式,等等.在上述特定并行策略与编程模型下,建立 CFD 计算任务向编程模型的映射关系,确定合理的任务划分与负载平衡策略,将直接影响到最终的并行模拟效率.

3.1.1　典型 CFD 应用中的多粒度并行层次

多层次并发是当今高性能计算机提升性能的主要手段,在这类平台上提高典型 CFD 应用的并行模拟效

率,充分挖掘 CFD 计算任务中多粒度、多层次的并发性将必不可缺.按照粒度从粗到细的次序,我们将典型 CFD 并行模拟中的并行性分解为以下几个层次.

(1)流场分区层:采用区域分解方法对网格和流场进行分组,整个流场域(domain)被划分成多个流场分区(zone),各个分区之间可使用数据并行策略进行并发模拟;为了适应现代异构并行计算机的特点,方便将任务分配给多种异构的计算设备,区域分解还可形成更多层次.

(2)计算模块层:单个流场分区内需要完成多项计算任务,当这些计算任务之间不存在依赖性时,可按照任务分解原则形成第二层次的并行性,例如对流项计算与粘性项计算可并发进行,每一项通量计算模块的内部三个方向间彼此无依赖也可并发计算.

(3)单区空间层:每个流场分区的单个计算任务内,往往需要对本流场分区内所有离散网格点(或网格单元)完成相同的计算,这种空间迭代计算蕴含了大量数据并行性;CFD 数值模拟实践中大量并行优化主要是针对这个层次的,而且往往需要使用数据分块技术将迭代空间划分出更多的并发子层,以便针对不同子层次开展不同的并行优化措施.

一旦在逻辑层次将 CFD 并行模拟任务分解成多层次的并发性子任务后,便可建立针对特定高性能计算机平台,分别建立 CFD 并行子任务向软件实现、软件实现向高性能平台硬件之间的映射与实现方案.

3.1.2 同构型计算环境下的 CFD 并行实现方案

在同构型计算环境下,软件并行主要通过传统的 MPI 多进程编程、OpenMP 多线程编程以及单指令多数据(SIMD)的向量化指令三个层次加以实现.

对于最粗粒度的流场分区层(图 2(*a*)所示),经典并行计算使用静态任务划分策略,通过对原始流场网格进行二次剖分获得更多的网格区块.为达到计算负载均衡、同时避免小网格块的出现,可采用 MPI 进程 + OpenMP线程混合并行的方法.该方法不需要过度剖分现有的网格块,而是通过为规模不同的网格块按比例指定一定数量的线程来调整线程级的负载平衡性.

对于中间粒度的计算模块层(图 2(*b*)所示),其并行主要通过 OpenMP 多线程实现.典型 CFD 模拟中单次迭代最耗时的计算过程包括对流通量计算及粘性通量计算、隐式求解器中大型线性方程组的求解,即图 1 及图 2 中的 invFlux, visFlux, eqnSolver 等模块,可重点针对这些模块进行多线程并行改造.这一粒度的并行可分成两个层次:(1)沿 X, Y, Z 三个坐标方向分别计算对流通量(invFlux)和粘性通量(visFlux)形成了六个子模块,它们之间没有相关性,可采用任务划分方式实现并发处理;(2)每个子模块内,都需要对流场区域内所有离散网格点进行计算,通过适当调整计算次序可消除部分数据依赖,从而可采用数据划分的方式加以并行化.第一层次并行粒度较大,第二层次并发度较高,合理配合使用可达到较好效果[8].

对于更细粒度的单区空间层(图 2(*c*)所示),由于主要针对 CFD 中单个流场区域内的某一流程模块进行,映射到硬件层次则体现为单处理器内的并行实现.传统 SIMD(单指令多数据)指令级并行优化可适用于这一层.以天河 2 系统为例,其 CPU 处理器具有 256 位宽的向量指令,MIC 协处理器具有 512 位宽的向量指令,充分利用好向量指令集和向量计算部件,以 SIMD 方式并行可获得最多 4 ~ 8 倍的双精度浮点计算性能收益.

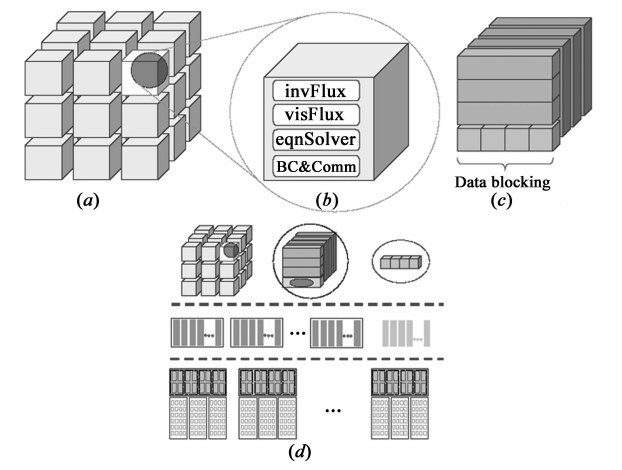


图2 典型CFD问题的多粒度并行性挖掘(*a*)~(*c*)及其并行模拟实现的多层次映射方法(*d*)

3.1.3 新型异构计算环境下的 CFD 并行实现方案

对于异构型平台,除了前一节同构计算环境下的 CFD 并行实现技术外,还需要额外考虑主处理器与加速器二者之间的编程模型、负载分配、任务调度等,以下将以天河 2 高性能计算机的 Offload 编程模型为例,讨论 CPU + MIC 异构协同并行方案.如图 2(*d*)所示,任务调度与负载均衡采用静态方式加以实现,共分成三个层次,分别描述了多区结构网格 CFD 应用的计算任务(应用任务层)、并行数值模拟中的多个进程与线程(运行实体层)以及高性能并行模拟系统的硬件平台结构(硬件平台层)等的特点与组织方式.通过在相邻两层之间建立静态的映射,便可实现任务分配与调度.

除了 3.1.2 节同构计算的通用方法外,异构混合并行实现中重点发展了不同类型处理器之间的协同计算方法.考虑到天河 2 系统中每个处理器结点包含了 2 个 CPU 计算设备和 3 个 MIC 计算设备,为了便于均衡负载,我们设计了如下运行配置方案:每个结点上运行多个 MPI 进程,每个进程使用 OpenMP 多线程方式利用结点内的一组 CPU 核,其中一个线程负责将部分计算任

务加载到 3 个 MIC 设备上.由于多个进程可同时将各自的部分计算任务加载到相应的 MIC 设备上,因此较容易发挥 MIC 设备的利用率.同一进程内的 CPU 与 MIC 协同执行,主要通过 Offload 加载模式将部分计算任务从 CPU 交由 MIC 设备执行,完成后将结果收集回 CPU.

为使 CPU 与 MIC 之间更易于达成负载平衡,我们将 3.1.1 节的流场分区层引入了更多中间层次,具体而言,将分配给每个进程的多个网格区块分成两类规格的 5 个分区组{{G0,G1};{G2,G3,G4}},其中 G0 和 G1 的计算任务交由进程内的两个 CPU 去完成,G2,G3 和 G4 规模相同或相近,其计算任务分别交由 3 个 MIC 设备各自完成.这种对等任务分解所带来的一个好处是,不论 CPU 上的计算任务还是 MIC 上的计算任务,尽管其硬件具有异构性,但在构造其更细粒度的并行(线程级、向量化指令级等)策略方面可以采用基本相似的思路.

3.2 面向同构及异构计算平台的多层次并行优化方法

3.2.1 并行负载平衡方法

流场分区层的并行负载平衡主要通过对现有网格进行二次剖分方式实现,超大规模的并行模拟往往对流场网格剖分提出较高的需求.例如:在高性能计算机系统上组织计算时,每次可用的计算机硬件资源(CPU 结点或 CPU 核数、单结点上的存储容量等)并不相同,为了使模拟过程能适应多种运行时的环境,通常需要先将网格块剖分成更小的子块,并重新组合这些新的子区块,以期分配给各进程上的计算任务量达到负载均衡.网格区块剖分与组合的工具及算法可参见文献[14],本文不再赘述.

对于 OpenMP 多线程并行层次的负载平衡,由于 CFD 应用问题中各离散点上的计算相对均衡,故基于迭代空间划分的多线程并行可简单采用静态任务划分,而在 3.1.2 节描述的计算模块层的 OpenMP 多线程实现方法中,由于 invFlux-X, invFlux-Y, invFlux-Z, visFlux-X, visFlux-Y, visFlux-Z 各子模块的离散点数目、各点上的计算量都可能存在差异,因此需要采用动态任务调度的方式以平衡各线程之间的负载.

对于异构混合计算,还需要考虑 CPU 与 MIC 两种处理器设备间的计算负载平衡问题,我们采用静态任务分配策略,并引入一个可调整的比例因子参数,用于控制 MIC 与 CPU 上总计算量的相对比例,最终针对不同的 CFD 算例测试调整该参数,以达到最佳的异构混合计算负载均衡性.第 4.2.2 节将给出具体测试算例讨论该参数的取值.

3.2.2 多线程并行优化

处理器结点内的众多处理器核主要是通过 OpenMP 多线程模型加以利用的,因此我们对多核多线程并行进行了优化,主要包括:(1)多线程并行的粒度与并发度.为在并行粒度与并发度二者之间取得平衡,通常对 CFD 的单区迭代空间实施多线程并行,并对三个坐标轴方向的循环同时实施数据分块.视运行设备是 CPU 还是 MIC 的不同,利用 OpenMP 编译指导语句对最外一层或数层的块循环进行多线程任务分配与调度.如图 3 所示.(2)减小线程创建与销毁的开销.尽可能将创建线程并行区的编译指导语句提前到嵌套循环的更外层循环中,以减少因反复动态创建与销毁线程带来的额外开销.(3)减小单个线程的内存占用量.为提升计算性能,OpenMP 多线程实现中经常使用大量线程私有变量,从而令应用程序的内存占用量过大,引发访存性能的下降.该问题对于存储容量更小的 MIC 协处理器来说尤为突出.为此我们采用共享数据分块私有化的方法,尽可能让每个线程只分配、访问和释放自身使用的数据,以获取最大的性能收益.(4)线程向处理器核的映射与绑定.确保每个软件线程与硬件处理器核间的静态映射与绑定,可提高 CFD 应用问题的访存性能.CPU 平台上可通过操作系统底层调用、或指定 OpenMP 环境变量加以实现,MIC 平台上可直接使用厂商提供的运行时环境变量选项加以实现.

```
!  $ OMP PARALLEL DO COLLAPSE(LEVEL) ! LEVEL = 1/2/3
do kb = 1, nk, kblksize
do jb = 1, nj, jblksize
do ib = 1, ni, iblksize
    do k = kb, kb + kblksize-1
    do j = jb, jb + jblksize-1
    ! DIR $ SIMD
    do i = ib, ib + iblksize-1
      ! update (i,j,k)
    enddo
    enddo
    enddo
enddo
enddo
enddo
```

图 3　数据分块的应用

3.2.3 多层次通信优化

典型 CFD 并行模拟中的数据通信(包含数据移动与传输)发生在跨机器结点之间、同一结点内的不同进程之间、同一结点内的 CPU 与 MIC 之间等多个层次,优化这些通信的关键是尽可能缩短通信时间、或者尽可能将通信时间隐藏在其它的计算过程之后.

(1)建立合理的“任务 - 硬件”映射,减少通信次数与通信时间.在流场区块层进行任务向 MPI 进程分配、以及运行时 MPI 向处理器结点建立映射时,应当尽可

能将物理上相邻的区块划分到同一进程或同一机器结点上,减少跨结点间的通信次数;将每一对进程间的多次通信打包合并成一次,同时充分利用非阻塞的通信模式,设法使 CFD 中的计算部分与通信部分相重叠,可使通信开销被部分或完全隐藏.

(2)多块相邻的奇异点通信的优化.对于某些格点型高阶精度 CFD 应用问题来说,奇异点通信会带来额外的开销,可采用文献[4,15]中介绍的非阻塞式点通信方式加以优化.

(3)CPU 与 MIC 间数据传输的优化.采用加载模式实现 CPU + MIC 协同并行的主要不利因素是需要使用低速的 PCIe 接口在两种设备之间传输大量数据,这可通过以下方法加以优化:①采用异步传输方式,并尽早启动数据传输.通过 CPU 计算与数据传输的有效重叠,可以部分或全部地隐藏数据传递的时间开销.②MIC 设备上数据的复用技术.若前一次加载到 MIC 上的计算结果数据能在下次加载时重用,则应避免重新传送这些数据;若一个变量仅有一段需要传递,则可利用编译指导语句! dir $ offload 语法仅传递必要的数据节段;若一些传递数据可由另一些传递的数据计算得到时,为减少数据传输量,必要时用重新计算代替直接传递这些数据.

3.2.4 宽向量指令级并行优化

在像天河 2 这样的现代型处理器上,不论作为主处理器的 CPU(支持 256 位宽的向量指令),还是作为协处理器的 MIC(支持 512 位宽的向量指令),用好宽向量指令都是充分发挥其浮点计算性能的关键要素之一,根据浮点精度及处理器类型的不同,可获得 4 ~ 16 倍的理论加速.通过使用 Intel fortran 编译器的编译选项 -vec、并配合用户指定的编译指导语句,编译器可实现大部分简单代码的自动向量化.对于编译器报告出的未能自动向量化的代码段,可有针对性地进行手工优化.

3.2.5 CPU + MIC 协同并行优化

在异构平台上实现 CFD 应用的协同混合并行模拟时,多种处理器设备在计算速度、存储容量、网络带宽及延迟等方面均呈现较大差别,必须仔细加以分析,合理设计计算模式、恰当组织计算流程,以达到其最佳协同性能. CPU + MIC 的协同混合并行主要在高性能计算机的单结点内进行,其 CPU 与 MIC 上的任务分配如 3.1.3 节所述,每个进程的多个网格区块分成两类规格的五个分组,CPU 负责计算第一类规格的两个分组,MIC 负责计算第二类规模的三个分组,两类规格网格的相对大小通过调整比例因子达到负载平衡,如图 4 所示.CPU 端通过 OpenMP 编程模式创建多个线程,由主线程负责与 MIC 设备交互,其余线程负责利用 CPU 的众核完成各自的计算任务.在此基础上,可展开一系列协同计算的并行优化.

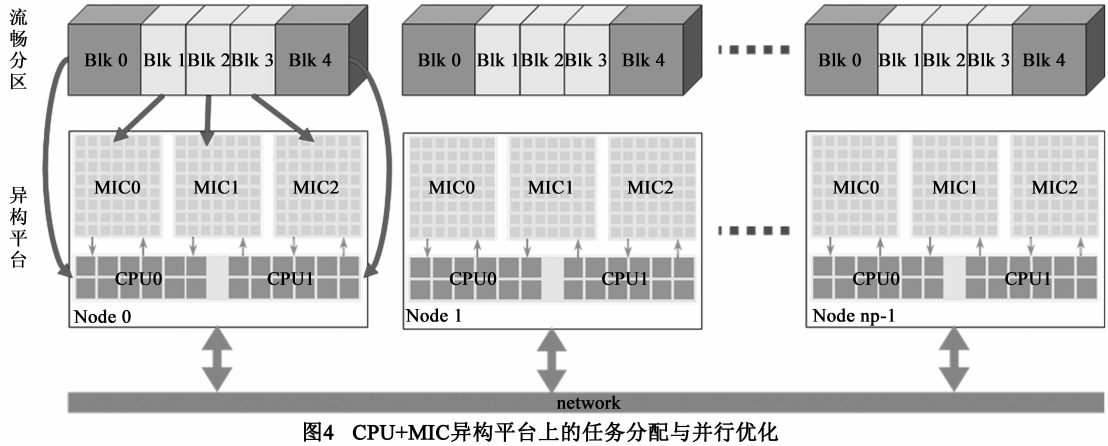


图4 CPU+MIC异构平台上的任务分配与并行优化

(1)降低 offload 加载带来的额外开销.通过程序流程重构,尽可能将可加载到 MIC 的计算核心组合成粒度更大的一个模块,以便通过一次加载完成所有任务,这样可避免因多次向 MIC 传入输出数据而带来的过大时间开销.在使用编译指导语句! dir $ offload 进行数据传递时,通过 alloc_if,free_if 等修饰从句避免每次加载时在 MIC 设备上重新分配及释放空间,而将空间分配及初始化放在迭代之前的"预热"阶段.

(2)不同层次通信之间的重叠.在图 1 的流程中,在每次 eqnSolver 模块求解完毕后,需要进行数据交换与同步,这些通信进一步可区分为两类:跨进程的 MPI 数据通信、进程内 CPU 与 MIC 的 offload 加载通信,如果能对 CFD 问题的网格数据进行合理划分,使得 MPI 通信与 offload 通信两者不产生依赖,则可通过异步通信将两者重叠,进一步隐藏通信开销,改善总体性能.例如,在图 4 所示的左右型一维任务划分方式中,一个计算结点上五个网格块组$\{\mathrm{blk}_i: i=1,\cdots,5\}$中,只有 CPU 负责的两侧区块组$\{\mathrm{blk}_0,\mathrm{blk}_4\}$需要同相邻结点进行 MPI 通信,

而 MIC 负责的三个区块组$\{blk_1, blk_2, blk_3\}$上的通信只限于结点内 CPU 与 MIC 之间,两种通信的并行重叠可有效降低通信总开销,如图 5 所示.

(3)异构处理器间计算任务的并行.在 CPU 与 MIC 设备、以及不同的 MIC 设备之间,计算任务的加载均采用异步方式,以便最大限度地实现多设备计算任务的并行与重叠,只在后续必要的位置进行同步.

图 5 示意了采用上述 CPU 与 MIC 异步通信、多种通信层次重叠、以及多种处理器间异步计算的处理过程.

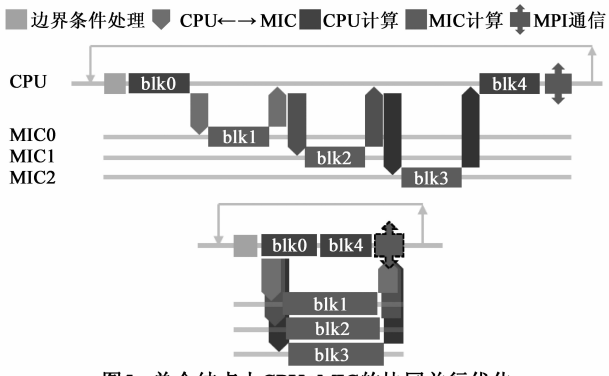


图5　单个结点上CPU+MIC的协同并行优化

除了上述并行优化外,传统针对单处理器的其它性能优化方法,例如面向 cache 的访存优化、数据对齐、异步 I/O 操作、数据双缓冲等,均适用于大规模 CFD 应用的并行模拟,这里不再赘述.

4　数值实验结果与讨论

4.1　测试平台

为了评估上节各种并行与优化方法的有效性,我们进行了一系列的数值实验.早期针对 CPU 同构平台的测试主要基于天河 1A 高性能计算平台,后期针对同构及 CPU + MIC 异构平台的测试主要基于最新的天河 2 高性能计算平台.两种平台的简明配置如表 1 所示,其中天河 2 全系统由 16000 个计算结点提供,每个计算节点呈异构体系结构,由 2 块 Intel Xeon E5-2692(至强 CPU)处理器以及 3 块 Intel Xeon Phi 31S1P(或简称为 MIC)协处理器组成.CFD 并行模拟使用内部开发的 In-House 软件,采用 Fortran 90 语言开发,编译器为 Intel fortran v13,CPU 上的编译选项取-O3 级别的优化,CPU + MIC 协同并行版本中对 MIC 端的优化选项取为-O3-xAVX 以便生成向量化代码.测试算例共取四种配置:(1)DeltaWing 三角翼周围流场,共 44 个网格分区,合计 240 万个网格点;(2)NACA0012 机翼外形流场,单个网格分区,共 1000 万个网格点;(3)DLR-F6 算例,网格量 1700 万;(4)CompCorner 可压缩拐角流场[16],主要用于各种网格规模下的性能测试,为此专门设计了可控制网格分布结构及规模的三维网格生成器,可生成各类规模的网格.本节中所有报告的结果均取 5 次以上独立测试中的最好结果,性能计时仅针对图 1 流程中的时间步迭代.为便于比较程序运行性能,时间步迭代取 50 步,并对测得的墙上时间进行了归一化处理.

表 1　天河测试平台的配置

	天河 1A	天河 2
CPU	Intel Xeon X5670 (6 核 CPU)	Intel Xeon E5-2692(12 核 CPU)
CPU 主频	2.93GHz	2.2GHz
单结点配置	2 CPU	2CPU + 3MIC
单结点存储容量	48 GB	64GB(for CPU) + 24GB(for MIC)
协处理器	(本文未使用)	Intel Xeon Phi 31S1P(57 核 MIC)
协处理器主频		1.1 GHz

4.2　测试结果与讨论

4.2.1　同构平台上的并行与优化效果

首先针对第 3 节负载平衡及通信优化方法,我们用包含 1000 万网格的 NACA0012 机翼外形算例在天河 1A 上进行了验证与测试,使用 64 个对称的 MPI 进程并行模拟,考查迭代求解过程中一步迭代内的通信与计算时间,实施并行通信优化前后的对比如图 6 所示.优化前的每步迭代中,通信时间的比例约为 63%.通过优化,我们消除了冗余的全局通信操作,并通过程序流程调整、最大限度地使用非阻塞式通信,以达到通信与计算重叠、隐藏通信的目标.这些优化大幅度地降低了通信总开销,将每步迭代中的计算通信比提高了近 10 倍.

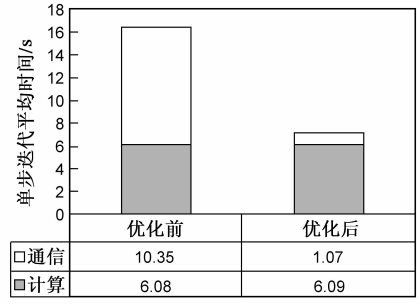


图6　通信优化前后单步迭代求解用时比较

对于 3.2.2 节针对多线程的并行优化,我们在使用数据分块技术的基础上,重点考查了线程向 CPU 核绑定策略对并行 CFD 应用性能的影响作用,我们使用了 DeltaWing 三角翼算例在天河 1A 平台上进行了测试,使用 CPU 核绑定策略及不使用该策略的性能数据如图 7 所示,图中纵轴为单步迭代平均时间(值越小越好),横轴为单个进程所使用的线程数.结果表明,线程向 CPU 核的绑定策略可显著提高并行模拟的性能,线程数目

越多,这种性能增益越明显.进一步考查表明,在 3.2.2 节线程级并行的多种优化方法中,数据分块及减少内存占用量也能略微改善访存、提高性能,但 CPU 核绑定策略对改善性能起到决定作用.

图 8 是使用第 3 节介绍的各类运行时优化技术,对两个不同网格规模的 CFD 算例在天河 1A 上应用不同运行时配置获得的相对加速比曲线图,其中横坐标给出了所使用的 CPU 核数,图 8(a)为 DLR-F6 算例(网格量 1700 万),纵坐标是相对于 2 个 CPU 核时性能的相对加速比(值越大越好),图 8(b)为 CompCorner 可压缩拐角算例(网格量 8 亿),纵坐标是相对于 480 个 CPU 核时性能的相对加速比.从图 8(a)中可知,对于中小规模的 DLR-F6 算例而言,当 CPU 核数小于 256 时,加速比基本呈线性增长关系,之后增加 CPU 核数时,加速幅度明显下降,这主要是由于程序的计算通信比变小造成的.具体来说有两方面原因,一是由于进程数增加,奇异点数量急剧上升造成 MPI 进程间通信开销变大[4,15];二是当网格区块数增加时,单块网格规模减小(最小时单个区块内网格量不足 1 万),采用更多线程并发反而增加了额外开销,加剧了并行性能的退化现象.图 8(b)算例中使用了更大的 8 亿个网格,分别采用 480、600、960、1200 和 2400 个 CPU 核进行并行模拟,由于单个网格块的规模较大(不低于 1000 万的网格量),整个并行程序的计算通信比较高,其相对并行加速比与并行 CPU 核数目之间呈明显的线性关系.

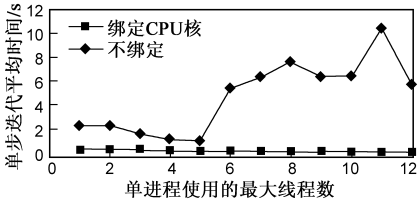
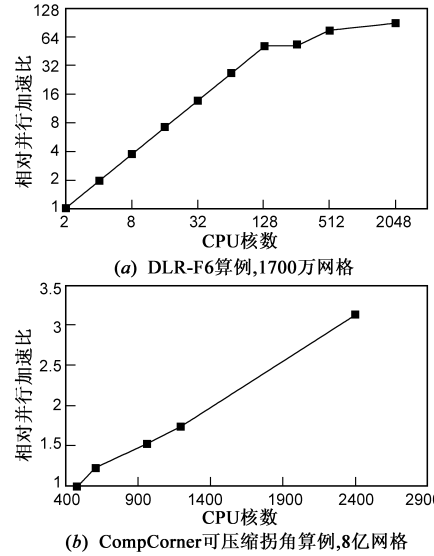


图7 线程与CPU核绑定策略对性能的影响比较

为了考查不同进程与线程组合配置下的并行 CFD 模拟性能,我们对中等规模的 DLR-F6 算例(1700 万网格)的各种运行配置进行了全面的测试,如图 9 所示.图 9(a)示意了各种组合配置下单步迭代时间与所使用的 CPU 核数间的关系(仅列出每进程上线程数不超过 4 的情形).首先,在具有线性加速比的 CPU 核数范围(图中 CPU 核数小于 256)内,只要 CPU 核数一定,不论采用哪种进程与线程组合配置,其性能差别并不大.其次,当限定每个进程最大线程数不超过 3 时,为了达到近似线加速效果,并行模拟最多可使用 256 个核,使用更多 CPU 核反而会造成并行效率的下降,但若允许单进程最大线程数为 4,则线性并行加速可扩展到 1024 个 CPU 核,两种情况下都最多使用了 256 个进程,此时制约并行模拟进一步扩大的主要障碍是进程间的数据通信.图 9(b)显示了不同进程数目情形下 CFD 单步求解时间随线程数目的变化曲线,结果表明,单纯靠增加线程数提高并行性能同样存在上限,对于本算例的问题规模而言,当使用 256 个进程时,线程数取 3 时达到最佳性能,而当进程数增加到 512 时,线程数最大只能取到 2.单进程内的计算通信比急剧减小仍是主要原因,事实上,对于 512 个进程的情形,每个进程上的计算工作仅涉及到 3 万左右网格点,此时若分配更多线程数,其带来的额外开销反而超过了因并行带来的性能提升.



(a) DLR-F6算例,1700万网格

(b) CompCorner可压缩拐角算例,8亿网格

图8 两个算例的并行加速比曲线

4.2.2 CPU + MIC 异构平台上的协同混合并行与优化效果

在 CPU + MIC 混合平台上移植与测试时,我们采用第 3 节介绍的协同并行方案,即每个结点运行 1 个 MPI 进程,结点内使用 OpenMP 多线程细粒度并行,其中主线程负责将部分计算任务加载到 3 个 MIC 设备上.在任务分配上,每个进程分配 5 个网格块,其中 2 个块取相同的规模,分别交由 2 个 CPU 计算,其余 3 个块取另一种规模,分别交给 3 个 MIC 设备进行计算.为了达到最佳的负载平衡效果,我们固定 CPU 负责的网格块规模为 8M × 2 = 16M(规模以网格单元数目表示,下同),每个 MIC 设备负责的网格块规模取 4M、6M、8M 和 10M 四种情形,图 10 报告了四个不同规模的 CompCorner 算例分别在向天河 2 异构平台移植前后的性能数据(以 16 结点 16 进程为例).结果表明,当单个 MIC 设备的网格块规模取 4M ~ 6M 时,协同并行达到最佳的加速效果,异构计算的加速达到 2.6 倍左右.

为进一步研究协同混合并行中 CPU 工作负载与 MIC 负载的最佳比例情况,我们针对 CompCorner 算例,

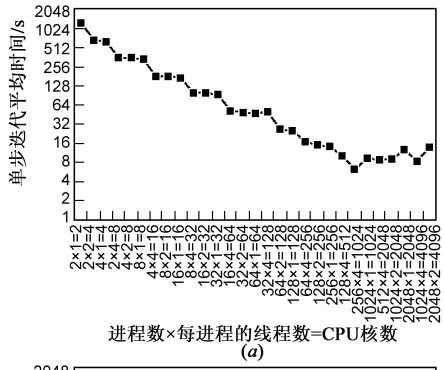


图9　不同进程+线程组合配置下的性能比较

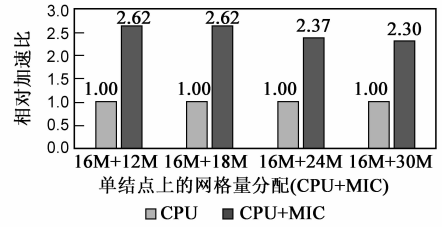


图10　CPU+MIC协同并行优化的性能效果

分别固定单 CPU 的工作负载为 8M,16M,24M,32M 个网格单元,并改变单 MIC 上的工作负载量,然后在天河 2 的 16 结点 16 进程下进行了各种性能比较,图 11(*a*)显示了整体计算性能(用"每秒更新的网格单元总数"刻画)随 MIC 负载比例的变化情况,可以发现,当单 MIC 与单 CPU 负载之比为 0.6 ~ 0.8 时,协同计算的性能最高.图 11(*b*)显示了当单 MIC 与单 CPU 负载比取为 0.6,单结点网格规模取为 3200 万时,使用天河 2 的更多计算结点时的弱可扩展性能.结果表明,由于实现了 MPI 通信与 CPU/MIC 间数据传输的重叠,增加进程数目并不会带来总时间开销的增大.

作为大规模测试,我们进一步采用高阶精度空间离散 CFD 程序[13],取 CompCorner 可压缩拐角算例的两种负载配置,在天河 2 系统上对同构并行以及 CPU + MIC 协同并行实现的可扩展性进行了测试.两种负载配置分别为:(1)粗网格(coarse):每结点处理 4000 万网格单元;(2)细网格(fine):每结点处理 9520 万网格单元.同构测试最大规模使用 8192 个结点,19.66 万个 CPU 核,最大网格量达 7800 亿个网格单元;CPU + MIC 协同测试最大使用 7168 个结点 137.6 万个 CPU + MIC 处理器核,最大网格量达 6800 亿个网格单元.测试结果如图 12 所示,该图显示了问题规模随进程数同比例增加时墙上时间的变化情况.结果表明,在每种 CPU + MIC 协同并行配置中,随着使用的处理器结点数与问题规模的同比增长,总模拟时间基本维持不变,表现了很好的弱可扩展性能.

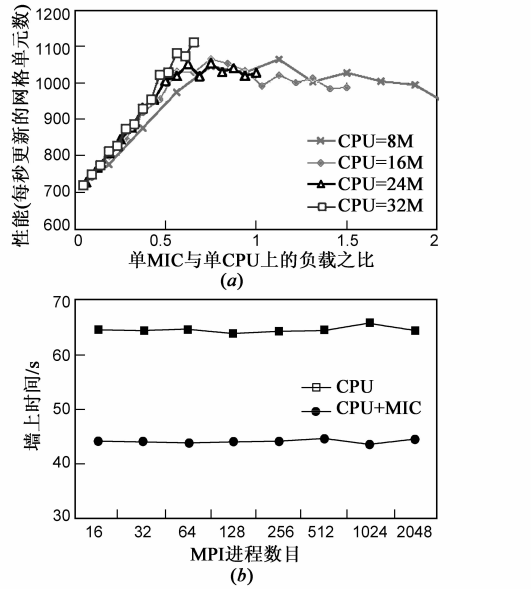


图11　不同的CPU与MIC计算负载比例下的协同并行性能比较

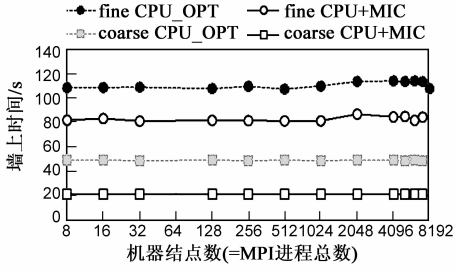


图12　CPU+MIC平台上不同负载比例下的可扩展性比较

5　结束语

本文就多区结构网格上的大规模 CFD 流场模拟的高效并行方法进行了研究,并将体系结构特点与应用问题相结合,讨论了在天河超级计算机上进行 CFD 模拟时的并行与优化技术,特别讨论了 CPU + MIC 异构平台的协同并行方法.大规模数值实验表明,CPU + MIC 异构并行优化后的程序模拟性能提高 2.6 倍;在天河 2 上异构协同模拟的最大规模 CFD 应用达到 6800 亿个网格单元,共使用 137.6 万 CPU + MIC 处理器核,结果显示优化后的程序具有良好的可扩展性.

致谢 感谢国家超算长沙中心及国家超算广州中心提供的天河 2 高性能计算机系统平台与技术支持.

参考文献

[1] Deng Xiaogang, Maekawa H. Compact high-order accurate nonlinear schemes[J]. Journal of Computational Physics, 1997,130(1):77 – 91.

[2] 刘昕.高阶精度加权紧致非线性格式研究与其在复杂流动中的应用[D].博士学位论文,中国空气动力研究与发展中心.2004.
Liu Xin. Study ofhigh-order accurate weighted compact nonlinear schemes and applications to complicated flows[D]. PhD Thesis, China Aerodynamics Research and Development Center.2004.(in Chinese)

[3] Deng Xiaogang, Mao Meiliang, Tu Guohua, et al. Geometric conservation law andapplications to high-order finite difference schemes with stationary grids[J]. Journal of Computational Physics,2011,230(4):1100 – 1115.

[4] 王勇献,张理论,刘巍,等.结点型多区结构网格的奇点重构算法[A].第十五届全国计算流体力学会议[C].山东烟台,2012.609 – 614.
Wang Yongxian, Zhang Lilun, Liu Wei, et al. A fast algorithm for reconstructing singular connection in the multi-block CFD applications[A]. Proceedings of 15th Conference of CFD in China[C]. Yantai, China, 2012.609 – 614.(in Chinese)

[5] Top 500 supercomputer sites[OL]. http://www.top500.org, 2014 – 03 – 12.

[6] Liu Li, Liu Li, Yang Guangwen. A highly efficient GPU-CPU hybrid parallel implementation of sparse LU factorization[J]. Chinese Journal of Electronics.2012,21(1):7 – 12.

[7] 王光学,张玉伦,等.WCNS 高精度并行软件的大规模计算研究[J].计算机工程与科学,2012,34(8):125 – 130.
Wang Guangxue, Zhang Yulun, et al. A study on massively parallel computation[J]. Computer Engineering and Science, 2012, 34(8):125 – 130.(in Chinese)

[8] Wang Yongxian, Zhang Lilun, Liu Wei, et al. Efficient parallel implementation of large scale 3D structured grid CFD applications on the Tianhe-1A supercomputer[J]. Computers & Fluids, 2013,80(10):244 – 250.

[9] Liu Buquan, Yao Yiping, Wang huaimin. On the Technology of high-performance parallel simulation[J]. Chinese Journal of Electronics, 2012,21(1):1 – 6.

[10] Thibault J, Senocak I. CUDA Implementation of a Navier-Stokes solver on multi-GPU desktop platforms for incompressible flows[A] Aerospace Sciences Meetings[C]. USA: American Institute of Aeronautics and Astronautics, 2009.758 – 772.

[11] Jacobsen D A, Senocak I. Multi-level parallelism for incompressible flow computations on GPU clusters[J]. Parallel Computing, 2013,39 (1):1 – 20.

[12] Subhash Saini, Haoqiang Jin, Dennis Jespersen, et al. An early performance evaluation of many integrated core based SGI Rackable computing system[A]. Proceedings of Supercomputing[C]. Denver, Colorado: SC, 2013.17 – 22.

[13] 王勇献,张理论,车永刚,等.高阶精度 CFD 应用在天河 2 系统上的异构并行模拟与性能优化[A].中国高性能计算学术年会[C].广西桂林,2013.1 – 31.
Wang Yong-Xian, Zhang Li-Lun, et al. Heterogeneous computing and optimization on Tianhe-2 supercomputer system for high-order accurate CFD applications[A]. Proceedings of HPC China[C]. Guilin, China, 2013.1 – 31.(in Chinese)

[14] 王勇献,张理论,刘巍,等.CFD 并行计算中的多区结构网格二次剖分方法与实现[J].计算机研究与发展,2013, 50(8):1762 – 1768.
Wang Yong-xian, Zhang Li-lun, Liu Wei, et al. Grid repartitioning method of multi-block structured grid for parallel CFD simulation[J]. Computer Research and Development, 2013, 50 (8):1762 – 1768.(in Chinese)

[15] Wang Yong-Xian, Zhang Li-Lun, Che Yong-Gang, et al. Improved algorithm for reconstructing singular connection in multi-block CFD Application[J]. Transaction of Nanjing University of Aeronautics & Astronautics, 2013, 30(S):51 – 57.

[16] 李邦明,鲍麟,童秉纲.高超声速压缩拐角峰值热流位置预测模型研究[J].力学学报,2012,44(5):869 – 875.
Li Bangming, Bao Lin, et al. Theoretical modeling for the prediction of thelocation of peak heat flux for hypersonic compression ramp flow[J]. Chinese Journal of Theoretical and Applied Mechanics, 2012,44(5):869 – 875.(in Chinese)

作者简介

王勇献 男,1975 年 7 月出生,河南安阳人.国防科技大学计算机学院副研究员、硕士生导师.主要从事高性能计算及其应用、并行算法等方面的研究工作.
E-mail: yxwang@nudt.edu.cn

张理论 男,1975 年 5 月出生,河南南阳人.国防科技大学计算机学院研究员、硕士生导师.主要从事高性能计算与应用方面的研究工作.